Spatiotemporal Multimodal Representation Learning for Activity Understanding and Behavior Monitoring

by

Ehsan Asali

(Under the Direction of Guoming Li & Tianming Liu)

Abstract

Advances in artificial intelligence and multimodal sensing technologies have transformed activity understanding and behavior monitoring in complex, real-world environments. This dissertation presents novel frameworks for spatiotemporal multimodal representation learning, focusing on precision livestock monitoring by integrating deep learning with multimodal data fusion to analyze behaviors using audio, RGB, depth, and kinematic data. It introduces DeepMSRF, a multimodal speaker recognition framework that fuses audio and visual streams via advanced feature selection, achieving superior accuracy over unimodal baselines. A low-cost 3D monitoring system for broiler chickens, utilizing affordable RGB-D sensors and custom hardware, enables continuous, non-invasive tracking in commercial farms. A dedicated pipeline for automated 3D gait scoring employs synchronized RGB-D video, pose estimation, and segmentation to provide accurate, high-throughput welfare assessments, outperforming manual methods. Separately, a zero-shot and transformer-based approach for 3D footpad and gait scoring recognizes unseen behaviors and conditions, enhancing adaptability across diverse farm settings. Additionally, mask-based multimodal action recognition pipelines leverage RGB-D data, zero-shot segmentation, and spatiotemporal models to classify fine-grained behaviors like feeding and drinking, offering insights for farm management. Comprehensive experiments validate these approaches, while new datasets, open-source tools, and evaluation protocols advance activity recognition, multimodal fusion, and precision animal welfare monitoring, supporting intelligent, ethical, and sustainable livestock production.

Index words: Spatiotemporal Feature Extraction, Deep Learning, Object Detection, Zero-Shot Image Segmentation, Multimodal Data Fusion, 3D Sensing, Precision Livestock Monitoring, Gait Scoring, Footpad Scoring, Speaker Recognition, Action Recognition, Behavioral Analysis

Spatiotemporal Multimodal Representation Learning for Activity Understanding and Behavior Monitoring

by

Ehsan Asali

B.Sc., Shiraz University of Technology, Shiraz, Iran, 2014
M.Sc., Shiraz University, Shiraz, Iran, 2017

A Dissertation Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the Degree.

Doctor of Philosophy

Athens, Georgia

2025

Spatiotemporal Multimodal Representation Learning for Activity Understanding and Behavior Monitoring

by

Ehsan Asali

Co-Major Professors:	Guoming Li and Tianming Liu
Committee:	Geng Yuan Wei Niu

Electronic Version Approved:

Ron Walcott
Dean of the Graduate School
The University of Georgia
August 2025

Dedication

"Sometimes it is the people no one can imagine anything of who do the things no one can imagine." —Alan Turing

To **Mahtab**, my beloved wife, for her unwavering love, enduring support, and constant encouragement. Her strength and belief in me have carried me through every step of this journey.
To **Zahra**, my dearest mother, for being my greatest supporter forever, always pushing me toward success and prosperity with love and determination.
To **Esmaeil**, my devoted father, for his enduring faith in me and for trusting and supporting my academic path with quiet strength and pride.
To **Omid**, my cherished brother, for being both my closest friend and steadfast companion from childhood to adulthood, always there when I needed him most.

Acknowledgments

First and foremost, I am deeply indebted to my major advisor, Dr. Guoming Li. From the moment I joined the Intelligent Systems for Poultry (ISP) lab, his unwavering trust, insightful guidance, and relentless dedication have steered me back onto the path of rigorous research whenever I faltered. His mentorship has not only honed my analytical and experimental skills, but also fostered in me a true appreciation for the curiosity and perseverance that underlie every scientific discovery. I am profoundly grateful for the countless hours he generously devoted to my development—both as a researcher and as an individual—and for the enduring confidence he has placed in me. His influence will forever shape my professional journey.

I also wish to thank Dr. Tianming Liu for graciously agreeing to serve as my co-major advisor. Although our acquaintance was brief at first, his readiness to support my work and his valuable insights have been truly encouraging.

My sincere thanks go to my committee members, Dr. Geng Yuan and Dr. Wei Niu, whose constructive critiques and collaborative spirit greatly strengthened our joint publications.

I am grateful to my former advisor, Dr. Prashant Doshi, for the years of guidance and collaboration at the THINC Lab. His mentorship during lab meetings and farm visits taught me lessons I will carry forward throughout my career.

I would like to acknowledge the School of Computing and the Department of Poultry Science for generously funding my travel to conferences and for providing outstanding facilities that supported every stage of my doctoral research. I am also deeply thankful to Dr. Hamid Arabnia, who has always been like a father to me and fellow Iranian students at UGA, offering unconditional support and wise guidance whenever it was needed.

Heartfelt thanks to my labmates at ISP lab and also at THINC Lab for their camaraderie, spirited discussions, and willingness to lend a hand whenever challenges arose. I also want to thank Saeid Tafazzol, Farzan Shenavarmasouleh, Farid Ghareh Mohammadi, Erfan Maddah, Jeff Thompson, and his family for being such good friends throughout my PhD journey, for their positive energy, and for all the help they offered along the way.

A special note of gratitude goes to Prasanth Sengadu Suresh. Like a brother, he offered invaluable feedback on my work, shared his expertise whenever I encountered obstacles, and provided steadfast companionship through both the triumphs and trials of this journey.

I would also like to honor the memory of my dear friend, Ardavan Afshar, who passed away in October 2020. He was my closest companion during my time in the United States and served as my academic idol; his absence has been deeply felt since that day. May God bless his soul and grant him eternal peace.

To my beloved wife, Mahtab: your presence has brought immeasurable joy and hope into my life. Your constant encouragement and gentle reminders of what truly matters have given me strength on days when I doubted myself. Thank you for inspiring me to reach both my professional and personal goals.

Finally, to my parents and my brother: your unwavering belief in me has been my greatest source of strength. Your love, support, and pride have carried me through every step of this endeavor, and for that I am eternally grateful.

Publication List:

- Asali, Ehsan, Farzan Shenavarmasouleh, Farid Ghareh Mohammadi, Prasanth Sengadu Suresh, and Hamid R. Arabnia. "Deepmsrf: A novel deep multimodal speaker recognition framework with feature selection." In Advances in Computer Vision and Computational Biology: Proceedings from IPCV'20, HIMS'20, BIOCOMP'20, and BIOENG'20, pp. 39-56. Cham: Springer International Publishing, 2021.

- Asali, Ehsan, Guoming Li, Chongxiao Chen, Oluyinka A. Olukosi, Iyabo Oluseyifunmi, Nicolas Mejia Abaunza, Tongshuai Liu, Mahtab Saeidifar, Venkat Umesh Chandra Bodempudi, Aravind Mandiga, Sai Akshitha Reddy Kota, Ahmad Banakar. "Integration and evaluation of a low-cost intelligent system and its parameters for monitoring three-dimensional features of broiler chickens." Computers and Electronics in Agriculture 237 (2025): 110553.

- Asali, Ehsan, Guoming Li, Tongshuai Liu, Chongxiao Chen, Mahtab Saeidifar, Venkat Umesh Chandra Bodempudi, Oluwadamilola Moyin Oso, Aravind Mandiga, Sai Akshitha Reddy Kota, Tianming Liu, "A Novel Three-dimensional Deep Learning Approach for Auditing Gait Scores of Individual Broiler Chickens", Submitted to Journal of Big Data: Customization and fine-tuning of machine learning models Special Issue, 2025.

- Asali, Ehsan, Guoming Li, Tongshuai Liu, Mahtab Saeidifar, Venkat Umesh Chandra Bodempudi, Oluwadamilola Moyin Oso, Aravind Mandiga, Sai Akshitha Reddy Kota, Geng Yuan, "Zero-Shot Perception and Spatiotemporal Transformers for Automated Gait and Footpad Score Classification", Accepted by The 2025 World Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE'25).

- Asali, Ehsan, Guoming Li, Mahtab Saeidifar, Tongshuai Liu, Venkat Umesh Chandra Bodempudi, Oluwadamilola Moyin Oso, Aravind Mandiga, Sai Akshitha Reddy Kota, Wei Niu, "Comparative Evaluation of Mask-Based Multimodal Action Recognition Pipelines for Broiler Chickens Using Early and Late Fusion Strategies", to be submitted to the Computers and Electronics in Agriculture (COMPAG) Journal.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Advancements in artificial intelligence, deep learning, and multimodal sensing technologies have significantly transformed the landscape of activity understanding and behavior monitoring across diverse domains. As the need for robust and interpretable systems continues to grow, so does the importance of integrating spatiotemporal data from multiple sources to address real-world challenges with higher precision and reliability. This dissertation explores novel approaches for spatiotemporal multimodal representation learning, with a particular emphasis on activity analysis and automated monitoring in precision livestock environments. By developing and evaluating deep learning frameworks that fuse information from different data modalities, such as RGB, depth, and kinematic data, this work aims to bridge existing methodological gaps, enhance decision-making, and contribute to both fundamental research and practical applications in behavior analysis, animal welfare assessment, and poultry management.

1.1 Motivation and Background

The increasing complexity of monitoring dynamic behaviors in both human and animal populations has underscored the limitations of traditional single-modality approaches. In recent years, research in activity understanding and behavior monitoring has pivoted towards leveraging the complementary strengths of multimodal data fusion, spurred by advances in sensor technology and computational modeling. The motivation for this work arises from the pressing need to capture nuanced behavioral patterns and environmental interactions that cannot be fully represented by a single data source. In particular, the agricultural sector, and precision livestock farming specifically, presents unique challenges and opportunities for multimodal analysis—where accurate, real-time monitoring of animal health, welfare, and behavior is essential for sustainable and ethical production. This dissertation addresses these challenges through innovative frameworks that integrate audio, visual, depth, and kinematic data to enable robust speaker recognition, detailed three-dimensional monitoring, automated gait and footpad scoring, and fine-grained action recognition in broiler chickens. This section establishes the foundational concepts and the evolving context that drive the research presented in this dissertation.

1.1.1 Significance of Activity Understanding and Behavior Monitoring

Understanding the activities and behaviors of living organisms—whether humans or animals—has long been recognized as a cornerstone of scientific inquiry, technological advancement, and societal development. In recent decades, the proliferation of sensing technologies and data-driven analytics has elevated activity recognition and behavior monitoring to the forefront of research in artificial intelligence, robotics, healthcare, and agriculture. Accurate interpretation of complex actions and behavioral patterns enables the development of intelligent systems capable of automated surveillance, health monitoring, productivity assessment, and anomaly detection. In human-centered domains, such systems underpin applications ranging from ambient assisted living and smart surveillance to workplace safety and autonomous vehicles. In animal populations, the ability to monitor activity with high granularity opens new possibilities for welfare assessment, early disease detection, and optimized resource management. For instance, in poultry farming, detecting subtle changes in locomotion or interaction patterns with environmental resources (e.g., feeders and drinkers) can provide critical insights into animal health and welfare. Fundamentally, robust activity understanding is essential for creating responsive and adaptive artificial agents and enhancing our scientific understanding of behavior itself. As challenges such as environmental variability, occlusions, and inter-individual differences persist, the field continues to evolve towards more sophisticated, interpretable, and generalizable models that can faithfully capture the richness of real-world behavior.

1.1.2 Role of Multimodal Data Fusion

The integration of information from multiple data modalities—commonly referred to as multimodal data fusion—plays a pivotal role in advancing the capabilities of activity recognition and behavior monitoring systems. Each sensing modality, whether visual, auditory, depth, or kinematic, offers a unique perspective and captures complementary aspects of the underlying phenomena. Visual sensors, for example, provide spatial context and appearance cues, while depth sensors contribute geometric and volumetric information, and audio or kinematic data can reveal otherwise imperceptible temporal dynamics. By strategically combining these heterogeneous data sources, multimodal fusion mitigates the limitations and ambiguities inherent in single-modality systems, resulting in more robust, accurate, and context-aware inference. The process of fusion can be implemented at various stages of the learning pipeline, ranging from raw data and feature-level integration (early fusion) to decision-level and adaptive strategies (late fusion and attention mechanisms). In livestock monitoring, for instance, combining RGB and depth data enables precise tracking of animal movements and interactions, while adaptive fusion strategies can prioritize reliable modalities in challenging conditions. Recent advances in deep learning have further enabled the seamless learning of joint representations that model complex inter-modal relationships. In both human and animal behavior analysis, multimodal approaches have demonstrated superior resilience to noise, occlusion, and environmental variation, ultimately providing richer insights and supporting more effective interventions. As such, multimodal data fusion forms a central pillar of modern activity understanding frameworks, particularly in challenging real-world settings.

1.1.3 Importance of Precision Livestock Monitoring

Precision livestock monitoring has emerged as a critical domain within the broader scope of precision agriculture, driven by the increasing demand for sustainable, efficient, and ethical animal production systems. The integration of advanced sensing, data analytics, and automation technologies enables continuous, non-invasive observation of individual animals and their environment at unprecedented resolution and scale. This paradigm shift addresses longstanding challenges in animal husbandry, such as the early detection of health and welfare issues, optimization of resource use, and minimization of environmental impact. Accurate monitoring of animal behavior, gait, footpad condition, and interactions with resources like feeders and drinkers facilitates timely interventions that can prevent disease outbreaks, improve productivity, and enhance animal well-being. Moreover, granular behavioral data support scientific research aimed at understanding the interplay between genetics, nutrition, environment, and management practices. As societal expectations regarding animal welfare continue to rise, precision monitoring solutions provide the necessary transparency and accountability for producers and regulatory bodies alike. The confluence of computer vision, sensor fusion, and machine learning in this context not only accelerates scientific discovery but also translates directly to tangible benefits in operational efficiency, food safety, and ethical standards within the livestock industry.

1.2 Problem Statement and Objectives

The remarkable progress in deep learning, sensing technologies, and multimodal data analysis has substantially advanced the field of activity understanding and behavior monitoring. However, despite these advancements, significant challenges persist—particularly in translating laboratory-scale solutions into robust, generalizable systems capable of operating in complex and dynamic real-world environments. In domains such as precision livestock monitoring, the stakes are especially high: models must contend with noisy data, varying lighting and environmental conditions, occlusions, and the need for cost-effective deployment at scale. Moreover, the need to distinguish subtle behavioral differences, such as specific actions like feeding or drinking in poultry, adds further complexity to model design and data processing. This section delineates the fundamental problems that motivate this dissertation and articulates the overarching research objectives. By critically examining the current limitations and gaps in multimodal representation learning, as well as the specific hurdles faced in livestock monitoring applications, this section sets the stage for the contributions that follow.

1.2.1 Research Gaps in Current Multimodal Representation Learning

While multimodal representation learning has garnered significant attention for its ability to integrate complementary information from diverse data sources, several key gaps remain that limit its effectiveness in real-world applications. First, many existing models are developed and evaluated on controlled or idealized datasets that do not adequately reflect the variability, noise, and complexity inherent in operational settings such as farms, factories, or open environments. As a result, the learned representations may fail to

generalize when confronted with unseen scenarios, occlusions, or sensor failures. For example, in poultry monitoring, occlusions caused by feeders or other chickens can obscure critical behavioral cues, necessitating robust segmentation and tracking methods. Additionally, the challenge of aligning and synchronizing heterogeneous data streams—each with its own sampling rate, noise profile, and spatial-temporal context—often leads to suboptimal fusion and diminished model performance.

Another persistent gap lies in the interpretability and adaptability of multimodal models. Deep learning approaches, while powerful, frequently act as "black boxes," making it difficult to trace decision pathways or diagnose errors—an issue of critical importance in high-stakes domains such as animal health and welfare monitoring. Moreover, many fusion strategies are static or rely on heuristics rather than dynamically adapting to contextual cues or sensor reliability. This can result in models that are either too rigid or excessively sensitive to outliers and missing data. For instance, distinguishing between similar actions like feeding and drinking requires context-aware fusion to prioritize relevant modalities. The lack of principled, context-aware fusion methods hampers the deployment of truly robust systems that can operate continuously in challenging environments.

Finally, there is a dearth of research on low-cost, scalable, and resource-efficient solutions suitable for deployment in agriculture and similar resource-constrained domains. Much of the literature focuses on high-performance systems that assume abundant computational power and pristine data. In practice, constraints related to hardware, energy, connectivity, and data storage pose major obstacles. For example, processing high-resolution RGB-D data for action recognition in real time requires significant computational resources, which may not be feasible in typical farm settings. Consequently, there is a critical need for models and algorithms that can deliver high accuracy and resilience with modest infrastructure, while also enabling easy adaptation to new tasks and settings. Addressing these gaps is essential to realize the full promise of multimodal representation learning in activity understanding and behavior monitoring.

1.2.2 Specific Challenges in Livestock Monitoring

Livestock monitoring, especially in commercial production environments, poses a unique set of technical and practical challenges that distinguish it from other activity recognition domains. First, the dynamic and often harsh conditions within animal housing facilities—such as fluctuating lighting, variable temperatures, dust, and the presence of obstacles—directly impact the quality and consistency of sensor data. Animals themselves are not static targets; they move unpredictably, interact with their environment and with each other, and may frequently occlude each other or key objects of interest, such as feeders and drinkers. These factors lead to significant variability in visual appearance and behavior, complicating detection, tracking, and identification tasks. For example, accurately recognizing specific actions like feeding or drinking requires distinguishing subtle movement patterns in crowded pen environments. Moreover, the need for non-invasive and continuous observation mandates that monitoring solutions remain unobtrusive, affordable, and scalable across large populations and varying facility layouts.

Another major challenge is the inherent heterogeneity of the data and the subtlety of the target behaviors. For example, early signs of lameness, illness, or stress may manifest as minor changes in gait, posture, or activity patterns that are difficult to capture with single-modality or low-resolution systems. Integrat-

ing information from RGB cameras, depth sensors, thermal imaging, and even acoustic sensors is often necessary to accurately assess animal welfare and health. However, aligning and fusing these diverse data streams in real time is computationally demanding and requires robust calibration and synchronization techniques. The problem is further compounded by the natural diversity within and between animal populations, such as differences in size, color, growth rate, and behavioral norms, all of which must be accommodated by adaptable models.

From an operational perspective, practical constraints in farm settings create additional barriers. Many farms have limited access to high-bandwidth internet, reliable power, or advanced computing resources, making cloud-based or computationally intensive solutions less feasible. Systems must be designed to operate reliably in low-resource environments and to facilitate easy installation, maintenance, and data management by non-specialist personnel. Ensuring data privacy, minimizing storage requirements, and providing actionable outputs—such as behavioral insights for pen design or resource placement—that can be readily interpreted and acted upon by farm managers are also essential considerations. Taken together, these challenges underscore the need for innovative, robust, and context-aware approaches to livestock monitoring that can bridge the gap between research and practice.

1.2.3 Research Objectives and Contributions

In response to these multifaceted challenges, the overarching objective of this dissertation is to advance the state of the art in spatiotemporal multimodal representation learning for activity understanding and behavior monitoring, with a particular focus on precision livestock applications. The research aims to develop deep learning frameworks and multimodal fusion strategies that can robustly integrate diverse data types, operate reliably in real-world farm environments, and provide meaningful insights into animal behavior, welfare, and interactions. By leveraging advances in convolutional and transformer-based architectures, zero-shot learning, and adaptive fusion techniques, this work seeks to address key limitations in the generalizability, scalability, and interpretability of existing solutions.

A central contribution of this dissertation is the design and implementation of low-cost, intelligent monitoring systems that harness the power of multimodal sensing while remaining practical for deployment in resource-constrained agricultural settings. The research introduces and rigorously evaluates novel algorithms for three-dimensional behavior analysis, automated gait and footpad scoring, and fine-grained action recognition, all tailored to the specific requirements and constraints of livestock facilities. In particular, the integration of RGB-D imaging, object detection, pose estimation, segmentation, and mask-based action recognition techniques is explored to enable detailed and non-invasive monitoring of individual animals and their collective dynamics over time.

Beyond technical innovation, this work also contributes new datasets, evaluation protocols, and open-source tools that can support the broader research and practitioner communities. The findings demonstrate the feasibility and advantages of deploying multimodal, data-driven systems for large-scale animal welfare monitoring and behavior analysis, highlighting both the opportunities and persistent challenges in this emerging domain. Collectively, the objectives and contributions of this dissertation advance the

vision of smart, sustainable, and ethical livestock production through the application of cutting-edge artificial intelligence and sensor technologies.

1.3 Overview of Methodological Approaches

This dissertation employs a comprehensive suite of methodological approaches to address the complex challenges of activity understanding and behavior monitoring in livestock environments. Central to these approaches is the integration of advanced deep learning techniques with sophisticated multimodal data fusion strategies, enabling the robust extraction and interpretation of meaningful patterns from heterogeneous data sources. The following subsections detail the core computational paradigms—ranging from convolutional neural networks and transformer-based architectures to zero-shot learning frameworks—as well as the primary strategies for integrating multimodal data, including early, late, and adaptive fusion methods. These methodologies form the technical foundation of the research and underpin the development of practical, scalable solutions for precision livestock monitoring, including fine-grained behavioral analysis.

1.3.1 Deep Learning Techniques

Convolutional Neural Networks (CNNs): Convolutional Neural Networks (CNNs) have revolutionized computer vision by enabling automatic and hierarchical extraction of spatial features from images and video sequences. CNNs leverage convolutional filters, pooling layers, and non-linear activations to learn representations that capture essential patterns such as edges, textures, shapes, and objects. In the context of livestock monitoring, CNNs are widely used for tasks such as object detection, semantic segmentation, and keypoint localization—facilitating the identification and tracking of animals, estimation of body posture, and extraction of behavioral cues from visual data. For example, CNN-based models like YOLOv8 and X3D are employed for detecting chickens and recognizing specific actions like feeding or drinking. Their ability to learn directly from raw inputs without manual feature engineering makes CNNs particularly suitable for handling the visual variability and complexity inherent in real-world farm environments. Moreover, recent architectural innovations, including residual connections and depthwise separable convolutions, have further improved the efficiency and accuracy of CNN-based models.

Transformer-based Models: Transformer-based models, originally introduced for natural language processing, have rapidly gained traction in computer vision and multimodal learning due to their powerful capability to model long-range dependencies and contextual relationships. Unlike CNNs, which rely on local receptive fields, transformers employ self-attention mechanisms to dynamically weigh the importance of each input element, allowing for the flexible integration of information across entire sequences or images. Vision Transformers (ViTs) and related architectures, such as TimeSformer and X3D, have demonstrated remarkable performance in tasks such as image classification, action recognition, and spatiotemporal feature extraction. In this dissertation, transformer-based models are leveraged to capture the

temporal dynamics and cross-modal relationships inherent in activity sequences and multimodal datasets, enabling richer and more interpretable representations for behavior analysis, including fine-grained action recognition in poultry.

Zero-shot Learning: Zero-shot learning (ZSL) represents a paradigm shift in machine learning, where models are designed to recognize and categorize new classes or behaviors without explicit training examples for those classes. This is achieved by leveraging auxiliary information, such as semantic descriptions, attributes, or relationships between known and unknown categories. In the context of livestock monitoring, zero-shot learning is particularly valuable for detecting rare or emerging behaviors, adapting to novel environments, and reducing the dependence on labor-intensive data annotation. Recent advances in deep learning have enabled the practical application of ZSL through the use of embedding spaces, semantic alignment, and pre-trained foundation models. This dissertation explores zero-shot approaches, such as RAFT-3D and SAM2, for automated behavior analysis and segmentation, facilitating more scalable and adaptive monitoring systems.

1.3.2 Multimodal Data Integration Strategies

Early Fusion Approaches: Early fusion refers to the integration of multiple data modalities at the initial stages of the learning pipeline, typically by concatenating or combining raw inputs or low-level features before feeding them into subsequent model layers. This strategy enables the model to learn joint representations that capture complementary information across modalities from the outset. In livestock monitoring applications, early fusion might involve the direct combination of RGB, depth, and kinematic features to analyze behaviors like gait or feeding. While early fusion has the potential to exploit correlations between modalities effectively, it can also be sensitive to differences in data quality, scale, or noise, and may require careful preprocessing and normalization to ensure successful integration.

Late Fusion Approaches: Late fusion, in contrast, involves processing each modality independently through separate model streams, with integration occurring at a higher, typically decision or feature, level. Each stream extracts modality-specific features or predictions, which are then combined—often through averaging, weighted voting, or concatenation—before making the final inference. This approach offers flexibility and robustness, as each modality can be optimized separately, and the system can gracefully handle missing or corrupted data. In livestock monitoring, late fusion is particularly advantageous when sensor reliability varies or when certain behaviors, such as drinking or resting, are best captured by specific modalities. For instance, late fusion strategies are used in action recognition pipelines to combine features from RGB and depth streams for robust behavior classification. However, late fusion may miss subtle inter-modal interactions that are more effectively modeled at earlier stages.

Adaptive Fusion Methods: Adaptive fusion methods represent a more advanced class of integration strategies, wherein the model dynamically determines how and when to combine information from different modalities based on context, data quality, or task requirements. Techniques such as attention

mechanisms, gating networks, and context-aware weighting allow the system to selectively emphasize the most informative modalities or features at each inference step. Adaptive fusion is especially valuable in real-world settings where environmental conditions and sensor reliability can fluctuate unpredictably. For example, in poultry action recognition, adaptive gating mechanisms prioritize planar motion or depth data based on the specific behavior being analyzed. By leveraging adaptive fusion, this dissertation aims to develop systems that are both resilient to challenging scenarios and capable of extracting the most salient behavioral cues from complex, multimodal data streams.

1.4 Application Domains and Case Studies

This section provides an overview of the primary application domains explored in this dissertation, highlighting how multimodal representation learning and state-of-the-art deep learning techniques can be tailored to address real-world challenges in animal monitoring, behavior understanding, and human-centered applications. Each case study exemplifies a unique domain where integrating multiple data modalities and advanced models has enabled significant progress, from speaker recognition to precision poultry farming and fine-grained behavioral analysis. The selected domains are not only representative of diverse multimodal learning tasks but also demonstrate the scalability and adaptability of the proposed methods across both human-centered and animal-centered contexts.

1.4.1 Speaker Recognition (DeepMSRF)

Multimodal Approaches in Speaker Recognition: Speaker recognition systems traditionally relied on single-modal data, such as audio signals, for identifying and verifying speakers. However, recent advancements in multimodal deep learning have introduced the integration of additional cues—such as visual information from facial features or lip movements—to enhance recognition performance, especially under challenging acoustic conditions. The use of multimodal approaches in speaker recognition leverages the complementary nature of different sensory inputs, enabling more robust and accurate recognition by compensating for the limitations of individual modalities. This synergy is particularly valuable in scenarios with background noise, partial occlusions, or overlapping speech.

Challenges and Innovations in DeepMSRF: The DeepMSRF (Deep Multimodal Speaker Recognition Framework) system introduced in this work addresses several persistent challenges in the field, including the alignment of asynchronous audio-visual data, the extraction of discriminative features across modalities, and the design of fusion strategies that are both effective and computationally efficient. Innovations within DeepMSRF include a novel architecture for spatiotemporal feature extraction and selection, allowing the framework to dynamically adapt to varying quality and availability of input modalities. The system demonstrates significant improvements in speaker identification and verification tasks, particularly in noisy or unconstrained environments, underscoring the impact of sophisticated multimodal fusion in real-world scenarios.

1.4.2 Three-dimensional Monitoring Systems for Broiler Chickens

Existing Poultry Monitoring Techniques: Conventional poultry monitoring techniques primarily rely on manual observations and two-dimensional imaging, limiting the granularity and scalability of behavioral and phenotypic assessment. Although these methods provide basic insights into animal health and welfare, they often fail to capture subtle changes in posture, locomotion, and spatial interactions among birds. Recent advances in computer vision have enabled automated tracking and behavioral analysis using RGB cameras; however, these approaches are typically constrained by occlusions, lighting variations, and the inability to perceive depth, which is crucial for a comprehensive understanding of animal movement and space utilization.

Innovations in Low-cost 3D Systems: To address these limitations, this dissertation proposes and validates a low-cost, three-dimensional monitoring system that leverages affordable RGB-D sensors and deep learning-based object detection models like YOLOv8. This system enables real-time, fine-grained tracking of individual broiler chickens in commercial farm environments, capturing rich spatiotemporal data for behavior and welfare assessment. Key innovations include the integration of depth estimation with semantic segmentation and object tracking, facilitating accurate measurement of postures, activity levels, and interactions with environmental resources such as feeders and drinkers. The low-cost nature of the system ensures scalability and accessibility for practical deployment in the poultry industry, advancing the state of precision livestock farming.

1.4.3 Automated Gait and Footpad Scoring

Importance in Animal Welfare: Gait and footpad health are critical indicators of welfare and productivity in broiler chickens, as impairments can lead to pain, reduced mobility, and decreased performance. Traditionally, gait and footpad scoring are performed manually by trained assessors, a process that is labor-intensive, subjective, and often infeasible for large-scale operations. Automated assessment of these welfare indicators is essential for timely intervention and for ensuring the health and well-being of poultry flocks, aligning with growing regulatory and consumer demands for ethical animal production systems.

Deep Learning Approaches in Gait Analysis: The integration of deep learning with pose estimation and computer vision offers a promising avenue for automating gait and footpad scoring. This dissertation presents a pipeline that combines RGB-D video analysis, keypoint detection using YOLOv11, and spatiotemporal feature extraction to quantify gait characteristics and detect abnormalities. Advanced neural network architectures, such as 3D convolutional models and transformer-based frameworks, are employed to analyze complex movement patterns and accurately assign gait scores. This approach enables objective, repeatable, and high-throughput assessment of locomotor health, contributing to more effective management and welfare monitoring in poultry farming.

1.4.4 Zero-Shot Perception and Transformers in Livestock Monitoring

Need for Zero-Shot Techniques: In real-world livestock environments, it is often impractical to collect exhaustive annotated datasets for every possible behavior, condition, or object of interest. Zero-shot learning techniques address this challenge by enabling models to generalize to unseen classes or actions based on prior knowledge and semantic relationships. This capability is particularly valuable in animal monitoring, where behaviors and welfare indicators may vary across breeds, ages, or farm settings, necessitating flexible and adaptable perception systems.

Advances Through Transformer Models: Transformer-based architectures, with their self-attention mechanisms and capacity for integrating diverse modalities, have revolutionized the field of zero-shot perception. By leveraging pretrained vision-language models and large-scale multimodal datasets, transformers can effectively transfer knowledge to novel tasks and environments without requiring extensive retraining or manual annotation. In this dissertation, transformer-based approaches, such as TimeSformer and RAFT-3D, are employed to achieve robust zero-shot segmentation and action recognition in livestock monitoring applications, demonstrating substantial gains in adaptability, scalability, and generalization to new scenarios.

1.4.5 Mask-Based Multimodal Action Recognition for Broiler Chickens

Importance in Behavioral Analysis: Understanding specific behaviors such as feeding, drinking, moving, and resting in broiler chickens provides critical insights into their health, welfare, and resource utilization. Manual observation of these behaviors is labor-intensive and impractical for large-scale operations, necessitating automated systems capable of recognizing fine-grained actions in complex farm environments. Accurate action recognition enables producers to optimize pen design, resource placement, and management practices, ultimately improving animal welfare and production efficiency.

Innovations in Action Recognition Pipelines: This dissertation introduces multiple mask-based multimodal action recognition pipelines that integrate high-resolution RGB-D data, robust object detection (e.g., YOLOv8), zero-shot segmentation (e.g., SAM2), and advanced spatiotemporal models like X3D. These pipelines employ early and late fusion strategies to combine visual and depth information, enabling precise differentiation of subtle behaviors in crowded pen settings. By generating spatial heatmaps and temporal distributions of actions, the system provides actionable insights for farm management, demonstrating significant advancements in scalability, efficiency, and real-time performance for poultry monitoring.

1.5 Dissertation Structure

This dissertation is organized to systematically address the central research questions and objectives through a series of interconnected studies, each presented in a dedicated chapter. The overall structure reflects

a logical progression from foundational theory and methodological innovation to practical implementation and domain-specific applications in precision livestock monitoring. Each chapter builds upon the insights and technologies developed in preceding sections, culminating in a comprehensive understanding of spatiotemporal multimodal representation learning for activity understanding and behavior monitoring. The following paragraphs briefly summarize the focus and contributions of each chapter.

- **Chapter** 1 **- Introduction:** This chapter introduces the research problem, articulates the motivation for this work, outlines the research objectives and scope, and provides a roadmap for the remainder of the dissertation document.

- **Chapter** 2 **- DeepMSRF:** Chapter 2 introduces DeepMSRF, a novel deep multimodal speaker recognition framework with integrated feature selection. This chapter explores the challenges and opportunities of fusing audio and visual modalities for the task of speaker identification, particularly under conditions of limited or noisy data. The architecture leverages parallel convolutional neural network streams for audio and image features, followed by an advanced feature selection pipeline that improves accuracy and reduces redundancy. The chapter details the experimental design, comprehensive ablation studies, and comparative evaluation against unimodal and baseline methods. The results demonstrate the superior performance and interpretability of the proposed multimodal approach, setting the stage for subsequent investigations into multimodal fusion in other domains.

- **Chapter** 3 **- Low-cost** 3**D Monitoring System:** Chapter 3 presents the design, integration, and field evaluation of a low-cost, intelligent system for three-dimensional monitoring of broiler chickens. This chapter addresses the practical and technical challenges of deploying robust sensing solutions in real-world farm environments. It covers the selection and configuration of affordable hardware components, the development of efficient algorithms for 3D data acquisition and processing, and the implementation of storage and alert mechanisms suitable for large-scale operation. Experimental results are reported from extensive farm deployments, demonstrating the system's effectiveness in capturing detailed spatiotemporal information under diverse lighting and environmental conditions. The innovations described in this chapter lay the groundwork for more advanced behavior analysis in subsequent chapters.

- **Chapter** 4 **- Automated** 3**D Gait Scoring:** Chapter 4 focuses on the development of a deep learning pipeline for the automated 3D gait scoring of individual broiler chickens. Building upon the hardware and sensing infrastructure described in the previous chapter, this work introduces novel computer vision and machine learning algorithms for extracting pose, movement, and kinematic features from synchronized RGB-D video data. The pipeline integrates state-of-the-art object detection, pose estimation, segmentation, and feature fusion modules to enable accurate and non-invasive assessment of gait and welfare indicators. Rigorous experimental validation demonstrates significant improvements in accuracy, scalability, and robustness compared to manual and traditional approaches. The chapter also discusses the broader implications of automated gait analysis for animal welfare and production efficiency.

- **Chapter 5 - Zero-Shot and Transformer Approaches:** Chapter 5 extends the methodological contributions of this dissertation by investigating the application of zero-shot learning and transformer-based architectures to automated behavior and footpad score classification in livestock. The chapter describes the development of advanced models that leverage spatiotemporal transformers, adaptive fusion mechanisms, and zero-shot segmentation techniques to recognize behaviors and anomalies that were not explicitly represented in the training data. Detailed analyses highlight the advantages of these approaches in terms of generalizability, data efficiency, and adaptability to novel or rare events. Experimental results from real-world deployments illustrate the practical value of integrating zero-shot and transformer-based methods into large-scale livestock monitoring systems.

- **Chapter 6 - Mask-Based Multimodal Action Recognition:** Chapter 6 presents a comparative evaluation of multiple mask-based multimodal action recognition pipelines for broiler chickens, focusing on the recognition of key behaviors such as drinking, feeding, moving, and resting. This chapter leverages high-resolution RGB-D data, robust object detection, and advanced spatiotemporal models like X3D, combined with early and late fusion strategies, to achieve accurate and efficient behavior classification. The pipelines incorporate zero-shot segmentation and post-processing techniques to handle occlusions and complex pen environments, providing spatial and temporal insights into animal behavior. The chapter evaluates the performance of different modalities and fusion approaches, demonstrating their practical utility for optimizing farm management and resource allocation.

- **Chapter 7 - Concluding Remarks:** Chapter 7 provides a synthesis of the main findings and contributions of the dissertation, offering a critical reflection on the progress made and the challenges that remain. It highlights the broader impact of the research on the fields of computer vision, multimodal learning, and precision livestock monitoring. The chapter also outlines future research directions and opportunities for advancing automated behavior monitoring, with particular attention to scalability, real-time processing, and the integration of emerging sensing and artificial intelligence technologies. Through this synthesis, the dissertation concludes by emphasizing the transformative potential of spatiotemporal multimodal representation learning for both scientific discovery and practical application.

- **Bibliography:** A comprehensive collection of referenced works, ensuring accessibility and proper attribution.

Chapter 2

DeepMSRF: A novel Deep Multimodal Speaker Recognition framework with Feature selection

For recognizing speakers in video streams, significant research studies have been made to obtain a rich machine learning model by extracting high-level speaker's features such as facial expression, emotion, and gender. However, generating such a model is not feasible by using only single modality feature extractors that exploit either audio signals or image frames, extracted from video streams. In this paper, we address this problem from a different perspective and propose an unprecedented multimodality data fusion framework called DeepMSRF, Deep Multimodal Speaker Recognition with Feature Selection. We execute DeepMSRF by feeding features of the two modalities, namely speakers' audios and face images. DeepMSRF uses a two-stream VGGNET to train on both modalities to reach a comprehensive model capable of accurately recognizing the speaker's identity. We apply DeepMSRF on a subset of the VoxCeleb2 dataset with its metadata merged with the VGGFace2 dataset. The goal of DeepMSRF is to identify the gender of the speaker first and further recognize his or her name for any given video stream. The experimental results illustrate that DeepMSRF outperforms single modality speaker recognition methods by at least 3 percent accuracy.

2.1 Introduction

Artificial Intelligence (AI) has impacted almost all research fields in the last decades. There exists a countless number of applications of AI algorithms in various areas such as medicine (Afshar et al., 2020; Sotoodeh & Ho, 2019), robotics (Buffinton et al., 2020; Haeri et al., 2019; Seraj & Gombolay, 2020), multi-agent systems (Dadvar et al., 2020; Etemad et al., 2020; Karimi & Ahmazadeh, 2014), and security and privacy (Tahmasebian et al., 2020). Deep learning is, with no doubt, the leading AI methodology that has revolu-

tionized almost all computer science sub-categories, such as IoT (Voghoei et al., 2018), Computer Vision, Robotics, and Data Science (Mohammadi et al., 2020). The field of Computer Vision has been looking to identify human beings, animal, and other objects in a single photo or video stream for at least two decades. Computer vision provides variety of techniques such as image/video recognition (Amirian et al., 2018; Mohammadi et al., 2019), image/video analysis, image/video segmentation (Z. Wang et al., 2018), image/video captioning, expert's state or action recognition (Soans et al., 2020), and object detection within image/video (S. Ren et al., 2017; Shenavarmasouleh & Arabnia, 2020). Object detection plays a pivotal role to help researchers find the most matching object with respect to the ground truth. The greatest challenge of the object recognition task is the effective usage of noisy and imperfect datasets, especially video streams. In this paper, we aim to address this issue and propose a new framework to detect speakers.

2.1.1 Problem Statement

A copious amount of research has been done to leverage a single modality, which is either using audio or image frames. However, very little attention has been given to multimodality based frameworks. The main problem is speaker recognition where the number of speakers are around 40. In fact, when the number of classes (speakers) proliferate to a big number and the dimension of extracted features becomes too high, traditional machine learning approaches may not be able to yield high performance due to the problem of the curse of dimensionality (Mohammadi & Amini, 2019a, 2019b). To explore the possibility of using multimodality, we feed the video streams to the proposed network and extract two modalities, including audio and image frames. We aim to use feature selection techniques in two different phases in the proposed method.

Now this approach may prompt some questions: Why do we need multimodality if just a single modality would give us a high enough accuracy? Is it always better to add more modalities, or would an additional modality actually bring down the performance? If so, by how much? Bolstered by our experimental results, these are some questions we are going to delve into and answer in this paper. Let's start by looking at the potential impediments we could run into while using a single modality. Let's say, for instance, we just use audio-based recognition systems; in this case, we often face a bottleneck called SNR(Signal-to-Noise Ratio) degradation, as mentioned in (Chetty & Wagner, 2008). In short, when SNR is low in the input dataset, we observe that our model's efficiency plummets. On the other hand, image-based data is not unfettered by such predicaments as well. Images face problems like pose and illumination variation, occlusion, and poor image quality (P. Li et al., 2018; Mudumuri & Biswas, 2015; Sellahewa & Jassim, 2010; Shah et al., 2015). Thus, we hypothesize that combining the two modalities and assigning appropriate weights to each of the input streams would bring down the error rate.

2.1.2 Feature Selection

Feature selection is arguably one of the important steps in pre-processing before applying any machine learning algorithms. Feature selection or dimension reduction works based on two categories, including (i) filter-based and (ii) wrapper-based feature selection. Filter-based feature selection algorithms evaluate

each feature independent of other features and only relies on the relation of that feature with the target value or class label. This type of feature selection is cheap, as it does not apply any machine learning algorithms to examine the features. On the contrary, wrapper-based feature selection algorithms choose subsets of features and evaluate them using machine learning algorithms. That is the main reason why wrapper-based feature selection algorithms are more expensive. Mohammadi and Abadeh (Mohammadi & Abadeh, 2014a; Mohammadi & Abadeh, 2014b) applied wrapper-based algorithms for binary feature selections using artificial bee colony. In this study, we apply wrapper based feature selection, as it yields a high performance on supervised datasets.

2.1.3 Contribution

Most of the speaker recognition systems currently deployed are based on modeling a speaker based on single modality information, i.e., either audio or visual features. The main contributions of this chapter are as following:

- Integration of audio and image input streams extracted from a video stream, forming a multimodality deep architecture to perform speaker recognition.
- Effectively identifying the key features and the extent of contribution of each input stream.
- Creating a unique architecture that allows segregation and seamless end-to-end processing by overcoming dimensionality bottlenecks.

The rest of the chapter is arranged as follows: First, we touch base with the related work that has been done in this field, and then we explain the overview working methodology of CNNs, followed by how we handle the data effectively.

We also compare and contrast other classifiers that could be used instead of the built-in neural network of VGGNET. Then, we explain the experiments performed, compare the results with some baseline performance and conclude with discussion, future work and varied applications of the model developed in this paper.

2.2 Related Work

As explained before, most of the work done so far on speaker recognition is based on unimodal strategies. However, with the advancement of machine learning and deep learning in the past few years, it has been proven that multimodal architectures can easily surpass unimodal designs. (Chetty & Wagner, 2008; Chibelushi et al., 1994; Koda et al., 2009) were some of the very first attempts to tackle the task of person identity verification or speaker recognition while leveraging multiple streams to combine data collected from different sources such as clips recorded via regular or thermal cameras, and the varieties of corresponding features extracted from them via different external speech recognition systems, optical

flow modules and much more. After the features were extracted and fused, some basic machine learning models such as Hidden Markov Model (HMM), Latent Dirichlet Allocation (LDA), and Principal Component Analysis (PCA) were trained over them to act as the final classifier.

As impressive as these look, they can never beat the accuracy that one can achieve with deep learning models. Almost all of the architectures that are at the cutting edge in modern tasks, make use of two or more streams. Video action recognition is one example. (Feichtenhofer et al., 2016) and (Rezazadegan et al., 2017) both employ two parallel streams, one for capturing spatial data and the other for extracting temporal information from videos to classify the actions. Similarly, (Peng & Schmid, 2016) uses two separate streams for RGB frame and a sequence of different flows, whereas (X. Yang et al., 2016) brings four modalities into play and makes use of 2D and 3D CNNs and their flows simultaneously. Another excellent work (Soans et al., 2020) that deals with multimodal inputs, suggests a unique framework for recognizing robots' state-action pairs which uses two streams of RGB and Depth data in parallel. More creatively, (Feichtenhofer et al., 2019) utilizes one slow and one fast stream, proving that the former is good to understand spatial semantics, and the latter, which is a lighter pathway, can be beneficial in finding out the temporal motion features. Also, (Xiao et al., 2020) builds on top of this and adds one more stream to engage audio as well.

Tracking objects in videos, finding tampered images, and testing audio-visual correspondence (AVC) are some other tasks that parallel streams have been used for them to achieve the state-of-the-art performance. (Feichtenhofer et al., 2017) leverages two streams to jointly learn how to detect and track objects in the videos at the same time. (**He2018**) uses two parallel Siamese networks to do real-time object tracking, and (Zhou et al., 2017) employs face classification and patch triplet streams to investigate the possible alterations to the face images. Also, (Arandjelovic & Zisserman, 2017) and (Cramer et al., 2019) both use parallel streams to enable their models to identify whether the input audio corresponds to a given video or not.

It can be perceived that an extensive amount of research has been done in the field of multimodal deep architectures. Nevertheless, to the best of our knowledge, (Dhakal et al., 2019) is the most related work that has been done for the task of speaker recognition, and it only uses multimodality in the process of feature extraction. Additionally, it only uses audio data and tests on two datasets with 22 and 26 speakers, respectively. On the contrary, in this paper, we design our architecture to make use of video frames along with the audio in separate streams. On top of that, we create our custom dataset with 40 unique speakers and extend the scale of previous works.

2.3 Proposed Method

In this section, we propose a late-fusion framework using a dual-modality speaker recognition architecture using audio and frames extracted from videos. Firstly, we discuss challenges in speaker classification and recognition, then we talk about the bottlenecks of the architecture, and finally, we present our model's architectural details.

2.3.1 Challenges

As the number of images and videos proliferates, the process of image/video classification becomes more challenging; so, the task of real-world computer vision and data analysis becomes crucial when the number of classes exceeds 10. The more classes, the more time and computational power are required to do the task of classification. To learn a model to classify the speakers, we are required to have a proper dataset and a structured framework to do that. In this paper, the greatest challenge was to recognize 40 unique speakers. More so than that, since no standard dataset is available for our hypothesis, we had to create our own by subsampling from a combination of two other datasets. During the last 10 years, researchers have proposed different frameworks for deep learning using a complex combination of neural networks, such as ResNet (He et al., 2016) and GoogleNet (Szegedy et al., 2015) for image classification. These only focus on singular modality, either image or voice of the video, to do speaker recognition. In this paper, we address this problem and propose a new architecture leveraging VGGNET (VGG-16) (X. Zhang et al., 2015) for the speaker recognition task using multimodality to overcome all these limitations. The simple VGGNET, like other frameworks, suffers from having insufficient performance on speaker recognition. We provide three main steps consisting of combining two networks of VGGNET, followed by feature selection, and performing late-fusion on top of them.

Another common conundrum is on how to interpret the audio signals into a format that is suitable for VGGNET to work with. In general, in order to deal with audio streams, we have three options to choose from. One is to map the input audio to waveform images and feed the resultant diagrams to VGGNET as input. Another choice is to apply feature extraction to obtain a meaningful representation of the audio streams, which is now a feature vector. The last but not the least, we can perform one more step on top of feature extraction by visualizing them as a two-dimensional image. Later in this paper we will see that the third choice has the best performance and is utilized in our final model.

2.3.2 Video Speaker Recognition

We present a base speaker recognition architecture that leverages two VGG16 networks in parallel: One for speakers' images and another for speakers' audio frames. We discuss generating speaker audio frames in the next subsection. Figure 2.1 illustrates the base speaker recognition architecture. VGGNET produces a 1-D vector of $4,096$ features for each input frame that could be used as an input to all common classification methodologies. Fusing these feature vectors leads to high dimensionality problem called the curse of dimensionality (COD) (Mohammadi & Amini, 2019a, 2019b). To reduce the problem's complexity, we apply feature selection as discussed earlier in the Introduction of this chapter.

Data Preparation: We prepare a standard dataset of speaker images, along with speaker audios trimmed to four seconds, about which we discuss further in the Dataset subsection under the Experiments section. The dataset of the speakers' face images can be created quite easily; Nevertheless, as previously mentioned, we should convert speaker voices into proper formats as well. To tackle this issue, we have tried various

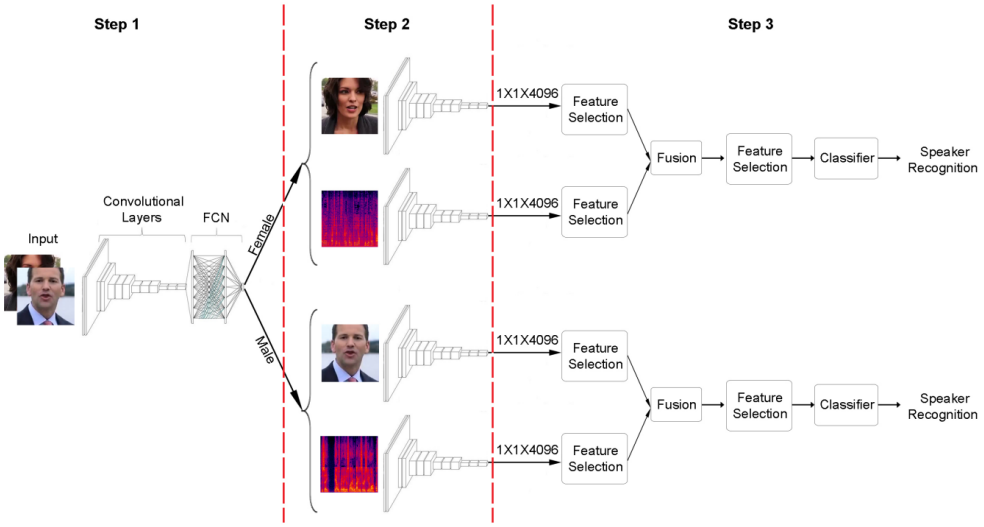

Figure 2.1: DeepMSRF architecture. Step 1: A unimodal VGGNET takes the speaker's image as input and detects the speaker's gender. Step 2: Based on the gender, the image and voice of the speaker are passed to the corresponding parallel multimodal VGGNETs to extract each modality's dense features. Step 3: Feature selection on each modality will be applied first; then, the resultant feature vectors are concatenated, and feature selection is performed again after concatenation. Eventually, a classifier is trained to recognize the speaker's identity.

available methods to generate meaningful features out of input audio signals. There are a couple of choices that we have briefly mentioned previously and will investigate more in this section.

The first approach is to directly convert the audio files into wave form diagrams. To create such images, the main hurdle that we face is the frequency variation of the speakers' voices. To solve this issue, we plot them with the same y-axis range to have an identical axis for all the plots. Obviously, the y-axis length must be such that all waveform charts fall within its range. The generated images can be directly fed into VGGNET; Nonetheless, later we will see that this approach is not really useful because of a lack of sufficiently descriptive features. Another approach is to extract meaningful and descriptive features of the audio streams instead of just drawing their waveform diagrams. Here, we have multiple options to examine; Mel-frequency cepstral coefficients (MFCCs), Differential Mel-frequency cepstral coefficients DMFCCs, and Filter Banks (F-Banks) are the algorithms reported to be effective for audio streams. MFCCs and DMFCCs can each extract a vector of $5,200$ features from the input audio file, while F-Banks can extract $10,400$ features. Now we have the option of either using these feature vectors directly and concatenating them with the extracted feature vectors of the face images coming from VGGNET Fully-Connected Layer 7 (so-called FC7 layer) or mapping them first to images and then, feeding them into the VGGNET. In the latter, we first need to fetch the flattened FC7 layer feature vector; afterwards, we have to perform the concatenation of the resultant vector with the previously learned face features. Spectrograms are another

meaningful set of features we use in this paper. A spectrogram is a visual representation of the spectrum of frequencies of a signal (audio signal here) as it varies with time. We feed such images into VGGNET directly and extract the features from the FC7 layer. Later, we will see how beneficial each of the aforementioned approaches is to predict the speakers' identity.

Feature Selection: The more features we have, the higher the probability of encountering overfitting problems, which is also known as the Curse of Dimensionality. This can be resolved to some extent by making use of feature selection (FS) algorithms. Feature selection approaches carefully select the most relevant and important subset of features with respect to the classes (speakers' identities). We choose a wrapper-based feature selection by exploiting the lib-SVM kernel to evaluate the subsets of features. After applying feature selection, the dimensionality of the dataset decreases significantly. In our work, we apply FS two times in the model; once for each of the modalities separately, and once again after concatenating them together (before feeding the resultant integrated feature vector to the SVM classifier.

2.3.3 Extended video Speaker Recognition

To do the task of video speaker recognition, we have divided our architecture into three main steps as presented in Figure 2.1. The following sections explain each step in closer scrutiny.

Step One: Inevitably, learning to differentiate between genders is notably more straightforward for the network compared to distinguishing between the identities. The former is a binary classification problem, while the latter is a multi-label classification problem with 40 labels (in our dataset). On the other hand, facial expressions and audio frequencies of the two genders differ remarkably in some aspects. For instance, a woman usually has longer haircuts, a smaller skull, jewelry, and makeup. Men, on the contrary, occasionally have a mustache and/or beard, colored ties, and tattoos. Other than facial characteristics, males mostly have deep, low pitched voices while females have high, flute-like vocals. Such differences triggered the notion of designing the first step of our framework to distinguish the speakers' genders.

Basically, the objective of this step is to classify the input into Male and Female labels. This will greatly assist in training specialized and accurate models for each class. Since the gender classification is an easy task for the VGGNET, we just use the facial features in this step. In fact, we pass the speakers' images to the VGGNET and based on the resultant identified gender class extracted from the model, we decide whether to use the network for Male speakers or Female ones. In our dataset, such a binary classification yields 100% accuracy on the test set. Later, we will see the effect of this filtering step on our results.

Step Two: In this step, we take the separated datasets of men and women as inputs to the networks. For each category, we apply two VGGNETs, one for speaker images and another for their voices. Thus, in total, we have five VGGNETs in the first and second steps. Indeed, we had one singular modality VGGNET, for filtering out the speakers' genders in the first step, and we have two VGGNETs for each gender (four VGGNETs in total) in the second step. Note that the pipeline always uses the VGGNET specified for the

gender recognition in the first step; Afterwards, based on the output gender, it chooses whether to use the parallel VGGNET model for women or men, but not both simultaneously.

In the given dataset, we have 20 unique females and 20 unique males. Following this, we train the images and audio of males and females separately on two parallel VGGNETs and extract the result of each network's FC7 layer. Each extracted feature vector consists of $4,096$ features, which is passed to the next step.

The second step may change a little if we use the non-visualized vocal features of either MFCCs, DMFCCs, or the F-Bank approach. In this case, we only need VGGNET for the speakers' face images to extract the dense features; Following this, we concatenate the resultant feature vector with the one we already have from vocal feature extractors to generate the final unified dataset for each gender.

Step Three: After receiving the feature vectors for each modality of each gender, we apply a classifier to recognize the speaker. Since the built-in neural network of VGGNET is not powerful enough for identity detection, we try a couple of classifiers on the resultant feature vector of the previous step to find the best classifier for our architecture. Nonetheless, before we feed the data into the classifiers, we need to ensure the amount of contribution each modality makes to the final result. As the contribution of each modality on the final result can vary according to the density and descriptivity of its features, we need to filter out the unnecessary features from each modality. To do so, we apply feature selection on each modality separately to allocate an appropriate number of features for each of them. Afterwards, we concatenate them together as a unified 1-D vector. We apply feature selection again on the unified vector and use the final selected features as input to the classifiers. The specific number of data samples, the number of epochs for each stage, and the results at each checkpoint are discussed in the Experiments section of this chapter.

2.4 Experiments

2.4.1 Dataset

We have used the VoxCeleb2 dataset proposed in (J. S. Chung et al., 2018), which originally has more than $7,000$ speakers, $2,000$ hours of videos, and more than one million utterances. We use an unbiased sub-sample (Shenavarmasouleh & Arabnia, 2019) of that with $20,000$ video samples in total, with almost $10,000$ sample speakers per gender. The metadata of the VoxCeleb2 dataset has gender and ID labels; the ID label is connected to the VGGFace2 dataset. The first step we have to go through is to bind the two metadata sets together and segregate the labels correctly. The way their dataset is arranged is that a celeb ID is assigned to multiple video clips extracted from several YouTube videos, which is almost unusable. Hence, we unfolded this design and assigned a unique ID to each video to make them meet our needs.

As mentioned earlier, we selected 40 random speakers from the dataset, which included 20 male and 20 female speakers. Thereupon, one frame per video was extracted where the speaker's face was clearly noticeable. The voice was also extracted from a 4-second clip of the video. Finally, the image-voice pairs

were shuffled to create training, validation, and test sets of $14,000$, $3,000$, and $3,000$ samples, respectively, for the whole dataset, i.e., both genders together.

2.4.2 Classifiers

Random Forests: Random forests (Ho, 1995) or random decision forests are an ensemble learning method for classification, regression, and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests prevent overfitting, which is common in regular decision tree models.

Gaussian Naïve Bayes: Gaussian Naïve Bayes (John & Langley, 1995) was first introduced in the 1960s (though not under that name), and it is still a popular (baseline) method for classification problems. With appropriate pre-processing, it is competitive in the domain of text categorization with more advanced methods, including support vector machines. It could also be used in automatic medical diagnosis and many other applications.

Logistic Regression: Logistic regression is a powerful statistical model that basically utilizes a logistic function to model a binary dependent variable, while much more complicated versions exist. In regression analysis, logistic regression (Kleinbaum et al., 2002) (or logit regression) estimates the parameters of a logistic model. Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail, which is represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name. The unit of measurement for the log-odds scale is called a logit, from the logistic unit, hence the alternative names. Some applications of logits are presented in (Amini et al., 2016; Karami et al., n.d.; Shahabinia et al., n.d.).

Support Vector Machine: A Support Vector Machine (Hearst et al., 1998) is an efficient tool that helps to create a clear boundary among data clusters in order to aid with the classification. The way this is done is by adding an additional dimension in cases of overlapping data points to obtain a clear distinction, and then projecting them back to the original dimensions to break them into clusters. These transformations are called kernels. SVMs have a wide range of applications from finance and business to medicine (Maddah & Beigzadeh, 2020) and robotics (Khayami et al., 2014). An example of its applications can be found in (Asali, Valipour, Afshar, et al., 2016; Asali, Valipour, Zare, et al., 2016; Asali et al., 2018), where they detect the opponent team's formation in a Soccer 2D Simulation environment using various approaches, including SVM. We use a linear kernel in our experiments, and we have Linear-SVM as a part of our proposed pipeline.

2.4.3 VGGNET Architecture

This section briefly explains the layers of our VGGNET architecture. Among the available VGGNET architectures, we have chosen the one containing a total of 13 convolutional and 3 Dense layers, famed as VGG-16 (Simonyan & Zisserman, 2014b). The architecture includes an input layer of size $224 \times 224 \times 3$ equal to a $2D$ image with 224 pixels in width and the same height, including RGB channels. The input layer is followed by two convolutional layers with 64 filters each and a max pooling layer with a window of size 2×2 and a stride of 2. Then another pair of convolutional layers of size 112×112 with 128 filters each and a max pooling layer are implemented. Afterwards, in the next three stages, the architecture uses three convolutional layers and one pooling layer at the end of each stage. The dimensions of the convolutional layers for these steps are $56 \times 56 \times 256$, $28 \times 28 \times 512$, and $14 \times 14 \times 512$, respectively. Finally, it has three dense layers of size $1 \times 1 \times 4096$ followed by a softmax layer. Since the output of the softmax layer specifies the output label (e.g., the speaker's name), its size must be equal to the number of classes. Also, notice that all convolutional and dense layers are followed by a ReLU function to protect the network from having negative values. Moreover, the first Dense layer is usually referred to as the FC7 layer (Fully Connected layer 7) that contains an extracted flattened feature vector of the input.

2.4.4 Implementation

In the Proposed Method section, we explained how we create the dataset, and now we elucidate the steps taken to produce the results. In order to train the parallel VGGNET for each gender, we divide the dataset into two parts: the samples of the 20 Male speakers and the samples of the 20 Female speakers. Thereafter, each of the two partitions is fed into a dual-channel VGGNET consisting of the image and audio streams. When the training process finishes, the architecture learns to extract meaningful features from the input data. Now, we can generate a new dataset for each gender by passing the face and Spectrogram's train, validation, and test images through their corresponding VGGNETs and fetching their FC7 layers' feature vectors.

Afterwards, using the linear SVM feature extractor library in Scikit learn - Python, we can extract almost $1,053$ features for Male images, 798 features for Male voices, $1,165$ features for Female images, and 792 features for Female voices. Then, we concatenate the resultant feature vectors for each gender and feed them again to the same feature extractor to summarize them once more. The final size of the merged feature vectors for the Males is $1,262$, and for Females is $1,383$. Note that the reported number of voice features are related to the Spectrogram feature extractor, which is the one we elected among the available options. The last step is to train the Linear SVM classifier and get its result.

The very first baseline architecture that we are going to compare our results with, does not segregate genders, uses only one modality (i.e., either the face or the voice data), uses the plotted waveform of the voice data, and does not use any feature selection approach. To compare the effect of any changes to the baseline, we have accomplished an extremely dense ablation study process. The ablation study results are discussed in the next section.

2.4.5 Ablation Study

To check the effect of each contribution, we perform an ablation study by training and testing the dataset in various conditions. The following sections briefly discuss the impact of each contribution on the final result.

Feature Extraction and Selection: Feature Extraction (FE) is highly crucial in the learning process. The main contribution of deep learning pipelines over classical machine learning algorithms is their ability to extract rich, meaningful features out of a high-dimensional input. Here, VGGNET plays this role for the face images of the speakers and also for the visualized vocal features. On the other hand, Feature Selection (FS) can prevent the model from being misled by irrelevant features. As previously mentioned, we have used linear-SVM feature selection in this work.

To evaluate the advantage of using FE and FS when dealing with auditory data and also to compare the performance of diverse FE algorithms, we apply each algorithm to Male, Female, and the whole dataset. Then, we apply FS on top of it; then, we examine each algorithm with four different classification methods, including Random Forests, Gaussian Naïve Bayes, Logistic Regression, and Support Vector Machines. Finally, we compare the results for both cases of using and not using FS. Table 2.1 shows the results for all the situations. As the results represent, the best test accuracy is achieved when we utilize the Spectrogram feature extractor combined with the linear-SVM feature selection approach.

Table 2.1: The accuracy for single/multi modality with/without feature selection.

Audio FE Algorithm	Random Forest	Gaussian Naïve Bayes	Logistic Regression	Support Vector Machines
Spectrogram(M) (%)	45.40	19.06	54.33	50.53
Spectrogram(M) + FS (%)	45.93	25.93	52.86	56.26
Spectrogram(F) (%)	44.26	21.53	52.26	48.46
Spectrogram(F) + FS (%)	42.66	29.40	51.20	53.30
Spectrogram(all) (%)	37.16	14.96	48.40	43.60
Spectrogram(all) + FS (%)	38.03	21.60	46.50	49.30
Waveform(M) (%)	30.53	16.6	32.26	29.26
Waveform(M) + FS (%)	30.46	17.20	29.93	32.13
Waveform(F) (%)	22.06	14.08	27.73	21.40
Waveform(F) + FS (%)	22.00	13.93	23.26	23.60
MFCC(M) (%)	11.93	25.33	5.46	9.46
MFCC(M) + FS (%)	11.46	24.20	5.33	9.20
MFCC(F) (%)	10.80	21.86	5.93	9.46
MFCC(F) + FS (%)	10.80	21.53	5.93	9.73
Filter bank(M) (%)	32.33	19.13	42.06	36.60
Filter bank(M) + FS (%)	33.00	24.00	42.26	40.46
Filter bank(F) (%)	33.66	18.06	43.20	38.06
Filter bank(F) + FS (%)	33	25.46	41.93	41.06

To analyze the efficacy of FS on the face frames, we train VGGNET and extract the FC7 layer feature vector. We then apply FS and eventually, train on four different classifiers. Table 2.2 represents the test accuracy for each classifier with and without FS. As the results demonstrate, the highest accuracy for each dataset is achieved for the case in which we have used FS on top of VGGNET and for the SVM classifier.

Table 2.2: The accuracy of the speakers' face images for four different classifiers associated with different feature extractors with/without Feature Selection (FS).

Face FE Algorithm	Random Forest	Gaussian Naïve Bayes	Logistic Regression	Support Vector Machines
VGG(M) (%)	91.00	49.66	93.60	91.66
VGG(M) + FS (%)	91.26	66.73	92.53	94.20
VGG(F) (%)	86.26	55.33	90.93	87.33
VGG(F) + FS (%)	85.66	62.20	88.13	91.26
VGG(total) (%)	88.03	50.10	91.53	88.70
VGG(total) + FS (%)	88.06	58.43	90.40	91.90

Gender Detection: As discussed earlier in detail, the first step of our pipeline is to segregate speakers by their gender. Instead, we could train a model with 40 classes consisting of all men and women speakers. To see how the first step improves the overall performance of the model, we examined both cases and compared their results. The test accuracy of Male speakers, Female speakers, the average test accuracy of Male and Female speakers, and the test accuracy of the whole dataset (containing both genders) are reported in Table 2.3. The results show that the average accuracy increases when we perform gender segregation regardless of whether we use feature selection before and/or after concatenating the face and audio modalities or not. Also, notice that according to Table 2.3, we can achieve the highest accuracy when we perform feature selection, specifically before the concatenation step. According to the table, the average accuracy has been improved by almost 4% using our proposed method (the last row) compared to the baseline approach (the first row).

Table 2.3: The accuracy of the whole dataset for the SVM classifier associated with the Spectrogram feature extractor combined with feature selection.

Feature Fusion Approach	Male	Female	Avg. of Male and Female	Total (Genderless)
Simple concatenation	91.20	87.87	89.54	89.27
FS + concatenation	95.13	91.87	93.50	92.97
concatenation + FS	94.67	91.87	93.27	92.53
FS + concatenation + FS	95.07	91.93	93.50	93.03

Single modality Vs. Multimodality: One of the greatest contributions of DeepMSRF is taking advantage of more than one modality to recognize the speaker efficiently. Each modality comprises of

unique features that lead the model to distinguish different individuals. To show how multimodality can overcome the limitations of single modality, we carry out a comparison between the two, reported in Table 2.4. According to the results, using both visual and auditory inputs together can improve the accuracy of the task of speaker recognition.

Table 2.4: The Accuracy for single/multi modality with/out feature selection

Feature Selection Involvement	**Face Frames**	**Audio (Spectrogram)**	**Multimodality**
Without Feature Selection	88.70	43.60	89.00
With Feature Selection	91.90	49.30	93.03

2.4.6 Time Complexity

In the previous section, we saw the benefit of utilizing feature selection on the model's accuracy. Additionally, there exist one more criteria to consider which is the training time. Although the training process is being performed offline and in the worst case, training the SVM classifier over our dataset finishes in almost 20 minutes (for the whole dataset), it is noteworthy to see how feature selection can influence the training time. Figure 2.2 depicts the training time required for the SVM in the last step (step 3) of our pipeline for the experiments shown in Table 2.3. According to Figure 2.2, the required training time for each gender is approximately one-third of the corresponding time required to train the whole dataset.

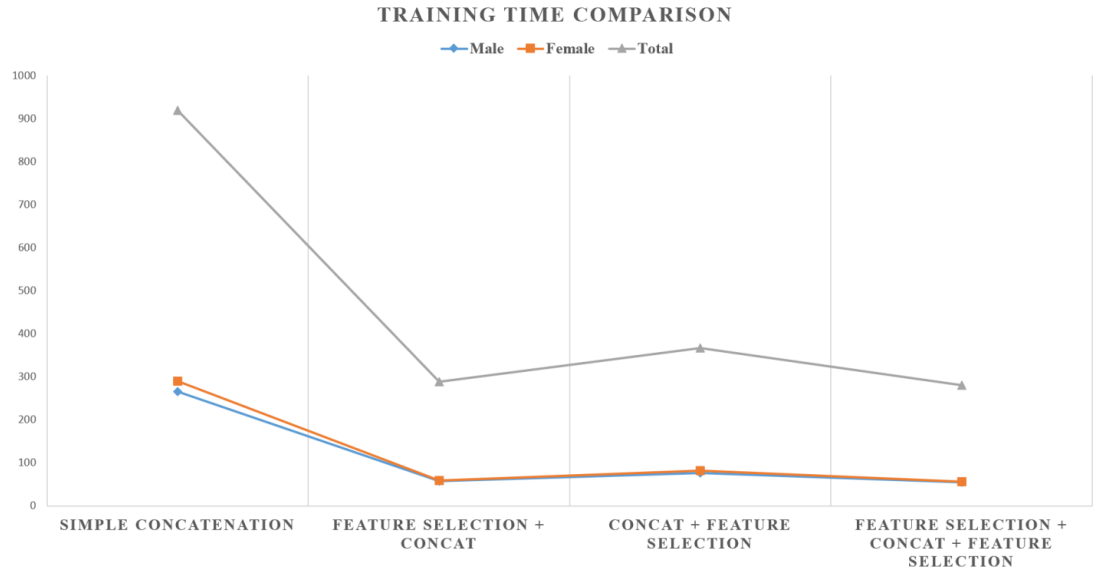

Figure 2.2: Training time Comparison (Male Vs Female) for DeepMSRF with SVM classifier.

Next, the time required to perform Step 3 of DeepMSRF, including the time needed for feature selection and the training time, is reported in Table 2.5. The results can be discussed from two different

points of view: (i) Among the examined methodologies, the shortest training duration is for the case in which we apply Feature Selection (FS) after the concatenation of the two modalities' feature vectors. On the contrary, the worst time performance is for the situation of applying FS before and after concatenation. (ii) The time is significantly shorter when we segregate the genders. Each gender's dataset needs less than one-third of the required time for the whole dataset. Training two separate models (one per gender), together, requires less time than training a general model that contains both genders.

Table 2.5: The training time (seconds) of different feature fusion approaches for each gender and both genders.

Feature Fusion Approach	**Male**	**Female**	**Total (Genderless)**
Simple concatenation	265.37	290.94	919.46
FS + concatenation	209.91	195.49	883.23
concatenation + FS	179.61	197.27	729.60
FS + concatenation + FS	296.93	291.56	1,208.52

2.5 Future Work

The future work entails using multiple datasets to test the robustness of the system and adding more functionalities to the model, such as recognizing the facial features of each individual person and predicting the extent of overlap between different people. Moreover, some aspects of the model can be investigated further. For instance, here we have used VGGNET; there remains a myriad of object detection pipelines to be tested. Another example of possible future probes is to try disparate approaches to feature selection. Altogether, this architecture has numerous applications beyond speaker recognition and is to be explored in the upcoming works.

2.6 Concluding Remarks for DeepMSRF

This paper takes a trip down the novelty lane by adding multimodality to improve the robustness of the recognition and overcome the limitations of single modality performance. From the results of the experiments above, we can infer that the hypothesis made about the multimodality improving over the single modality results for person recognition using deep neural networks was nearly conclusive. Among other challenges, this paper also solves the dimensionality challenge arising from using multimodality input streams. Exploiting feature extraction has provided a deep insight into how significant features to train the network are to be extracted to obtain a well-trained model. We can see that although the images provide a high accuracy over speaker recognition, audio stream input reinforces the performance and provides an additional layer of robustness to the model. In conclusion, we state that the unique framework

used in this paper, DeepMSRF, provides an efficient solution to the problem of speaker recognition from video streams. At last, DeepMSRF is a highly recommended framework for those researchers who deal with video analysis.

Chapter 3

Integration and evaluation of a low-cost intelligent system and its parameters for monitoring three-dimensional features of broiler chickens

Effective monitoring systems are crucial for improving poultry management and welfare. However, despite the enhanced analytics provided by 3D systems over 2D, affordable options remain limited due to unresolved design and algorithm challenges. The objective of this study was to develop a low-cost intelligent system to monitor the 3D features of poultry. The system consisted of data storage, a mini computer, and electronics housed in a plastic box, with an RGB-D camera externally connected via USB for flexible installation. Python scripts on a Linux-based Robotic Operating System (ROS Noetic) were developed to automatically capture 3D data, transfer it to storage, and notify a manager when storage is full. Various 3D cameras, installation heights (2.25, 2.50, 2.75, and 3.00 m), image resolutions, and data compression settings were tested using a robotic vehicle in a 1.2 m × 3.0 m pen to simulate broiler movement in controlled environments. Optimal configurations, based on the quality of 3D point clouds, were tested in several broiler trials, including one containing 1,776 Cobb 500 male broiler chickens. Results showed that the integrated L515 camera provided clearer features and superior 3D point cloud quality at 2.25 m, capturing an average of 1,641 points per frame. Additionally, data compression reduced RGB frame storage by 75%, enabling efficient long-term storage without compromising data quality. During broiler house testing with 1,776 Cobb 500 male broilers, the system demonstrated stable and reliable operation, recording 1.65 TB of data daily at 15 FPS with a 20 TB hard drive, allowing for 12 consecutive days of uninterrupted monitoring. Among object detection models tested, YOLOv8m (a medium-sized version of the YOLO version 8 model) outperformed other models by achieving a precision of 89.2% and an accuracy of 84.8%. Depth-enhanced modalities significantly improved detection and tracking

performance, especially under challenging conditions. YOLOv8m achieved 88.2% detection accuracy in darkness compared to 0% with RGB-only data, highlighting the advantage of integrating depth information in low-light environments. Further evaluations showed that incorporating depth modalities also improved object detection in extreme lighting scenarios, such as overexposure and noisy color channels, enhancing the system's robustness to environmental variations. These results demonstrated that the system was well-suited for accurately capturing 3D data across diverse conditions, providing reliable detection, tracking, and trajectory extraction. The system effectively extracted 3D walking trajectories of individual chickens, enabling detailed behavioral analysis to monitor health and welfare indicators. The system, costing approximately $\$1,221$, integrates cost-effective hardware with a scalable software architecture, enabling precision monitoring in large-scale operations. By reducing storage costs to \$28 per day and compressing data without losing critical details, the system is well-suited for practical deployment in poultry farms. The outlined evaluation and customization process ensures that this framework can be adapted to other agricultural or industrial applications, paving the way for intelligent and scalable monitoring systems. These advancements provide a robust foundation for improving animal management, enhancing productivity, and addressing welfare concerns in precision agriculture.

3.1 Introduction

The United States is the largest broiler producer in the world, with over 9.16 billion broiler chickens (approximately 27.0 million metric tons) harvested in 2023, valued at 42.6 billion USD (USDA National Agricultural Statistics Service, 2024). Despite supplying affordable proteins for humans, the intensive broiler industry is under pressure to provide birds with larger living spaces. A modern broiler house commonly accommodates $17,000$ to $48,000$ birds (Lei et al., 2022). The high rearing stocking density is favorable for economic profits, but makes it increasingly difficult for producers to inspect birds in detail on a daily basis. Growers commonly need to check birds twice a day and 7 days a week in a production cycle of 6–8 weeks, which is time-consuming and laborious (Certified, 2019). Flock inspection becomes even more challenging with the prediction of a labor shortage of farm jobs in the near future (Call & Stuesse, 2024). It is also inconvenient for growers to visit farms during pandemics (e.g., COVID-19) (Attia et al., 2022). Smart monitoring systems are urgently needed to assist growers in performing daily bird inspections, as they can continuously monitor birds 24 hours a day and 7 days a week.

Smart monitoring systems have been researched and developed to monitor various components related to poultry. Smart monitoring systems have been integrated into various commercialized robotic systems, such as ChickenBoy Robot, Octopus SCARIFIER, and T-MOOV (G. Ren et al., 2020; D. Wu et al., 2022). These commercialized robots can aerate litter, disinfect air, monitor bird status, and bird movements. For instance, ChickenBoy, a ceiling-suspended movable vision system, can measure environmental conditions (temperature, relative humidity, CO2, and air speed), examine excrements using artificial intelligence, find dead birds and defective nipple drinkers, and identify wet spots in the litter (Hartung et al., 2019). However, they require considerable investment or modification of housing systems

for accommodation, which is too costly for the poultry industry, which typically has low profit margins, or poultry research with limited budgets.

Intelligent portable units feature low cost with great mobility and scalability for monitoring key poultry parameters. (Ji et al., 2016) designed and evaluated the performance of an upgraded portable monitoring unit for measuring ammonia (NH3) and carbon dioxide (CO2) in commercial layer houses and demonstrated the feasibility of the upgraded unit in multiple-point measures for air quality in large-scale barns. (So-In et al., 2014) integrated cloud services as database and computational offloading, mobile phone and wireless sensor networks for data transmission, and image filtering into a unit to classify environment conditions and population density in broiler houses. The unit achieved 80% accuracy in classifying bird population density. (Lee et al., 2019) developed an automated chicken weighing system using a wireless sensor network for broiler farmers, and the weighing scales were equipped with a wireless data transfer system to enable automatic data transfer to the Cloud using a Wi-Fi module. While those smart systems assisted in monitoring critical components in poultry houses, they cannot directly monitor bird-based metrics such as behaviors and activities.

Vision systems can directly monitor, visualize, and analyze bird-based indicators. One well-known vision system for precision poultry farming is the two-dimensional (2D, RGB) camera system embedded with the software, eYeNamic. The system can monitor group-level activities and provide animal-based insights for flock distribution (De Montis et al., 2013). However, it cannot measure the information of individual birds. Meanwhile, the system, along with other poultry robotics systems, mainly uses 2D vision systems, which are subject to environmental challenges and have inherent limitations for animal phenotypic trait measurement (Viazzi et al., 2014). They are also expensive for low-profit poultry growers.

Alternatively, three-dimensional (3D) cameras obtain additional information (beyond 2D images), the distance between the camera and objects in the scene at each pixel. Because 3D cameras operate by emitting their own light onto the scene and measuring 3D structure (Ji et al., 2024), they are also more resistant to certain changes in the environment that would otherwise affect 2D image acquisition (G. Li et al., 2023). (Aydin, 2017) used a 3D vision camera system to automatically assess the level of inactivity in broiler chickens. (Mortensen et al., 2016) deployed a 3D computer vision system to predict the live body weight of broiler chickens. (G. Li et al., 2023) applied 3D vision to track and characterize spatiotemporal and three-dimensional locomotive behaviors and gait scores of individual broiler chickens. The previous studies clearly demonstrated the potential of 3D vision systems as valuable tools for bird monitoring in commercial and research facilities. However, there are no affordable, portable 3D monitoring systems for broiler chickens, and critical aspects such as system design and customization, key parameter configurations, sensing modalities, algorithm implementation, and core behavioral indicator extraction remain unexplored.

Several studies have applied deep neural networks for pose estimation and tracked broiler chickens' skeletal movements, enabling behavior classification such as standing, walking, and preening (Fang et al., 2021; G. Li, Hui, et al., 2020). Transfer learning techniques enhanced multi-animal pose estimation, as demonstrated by the Multi-Chicken Pose system, which achieved robust posture recognition for multiple chickens in group settings (Fang et al., 2024). Similarly, studies used RGB-D (R=Red, G=Green, B=Blue,

and D=Depth) data to estimate poultry movement trajectories and cluster behaviors into activity levels, offering insights into animal welfare status (Campbell et al., 2024; Peña Fernández et al., 2018). Advances like FCS-Net (Feather Condition Scoring Network) integrated RGB and infrared imaging to score feather conditions accurately, improving assessments of physical health (X. Zhang et al., 2024). While these techniques marked considerable progress, many systems rely on 2D imaging, which limits precision in occluded or crowded environments (Olanrewaju et al., 2024). Moreover, 3D vision-based systems have shown promise, with methods like latency-to-lie measurements correlating strongly with gait scores to identify lameness (Aydin, 2017), but such systems often lack the affordability and scalability required for widespread adoption. By integrating affordable 3D vision and automated processing, this study aims to bridge these gaps, offering an accessible solution tailored to the practical needs of large-scale poultry operations.

State-of-the-art object detection models have been widely applied in various domains, offering robust solutions for tasks requiring accuracy and speed. You Only Look Once Version 8 (YOLOv8; (Jocher et al., 2023)), is known for its real-time performance and has been widely adopted in dynamic applications such as autonomous vehicles, surveillance, and agricultural monitoring. Single Shot MultiBox Detector (SSD; (W. Liu et al., 2016)) is a lightweight, single-stage framework that efficiently detects object classes and bounding boxes, making it suitable for real-time applications in resource-constrained environments. EfficientDet (EfficientDet: Efficient Object Detection Network; (Tan et al., 2020)) incorporates compound scaling and a Bidirectional Feature Pyramid Network (BiFPN) for feature fusion, achieving an optimal balance between computational efficiency and detection accuracy, and has been applied in tasks requiring scalable and accurate object detection. Faster R-CNN (Faster R-CNN: Faster Region-based Convolutional Neural Network; (S. Ren et al., 2017)) utilizes a two-stage architecture to achieve high precision and accuracy, making it ideal for complex tasks, though its slower inference speed limits its use in real-time applications. These models, each with distinct strengths and trade-offs, provide valuable benchmarks for developing and evaluating advanced object detection systems across various applications, including precision poultry monitoring. All models were trained using default hyperparameter settings in TensorFlow (Python).

Object detection models, particularly YOLO (You Only Look Once) and Faster R-CNN, have significantly advanced poultry farming by addressing challenges related to health monitoring, behavior analysis, and welfare assessment. YOLO-based models, such as YOLOv8 and YOLOv5, have demonstrated exceptional accuracy in detecting broiler abnormalities like lethargy, stress, and slipped tendons using visual and thermal imagery, even under challenging lighting conditions (Elmessery et al., 2023; Khairunissa et al., 2021; Nakrosis et al., 2023). Faster RCNN and other CNN-based models were employed for tasks such as counting chickens and identifying abnormalities in smart poultry farms, enabling optimized management through edge AI technologies that reduce reliance on cloud-based systems (Čakić et al., 2022; Lubich et al., 2019). Advanced methods, including super-resolution fused with YOLOX, improved detection accuracy for small objects and addressed challenges like occlusion and variability in free-range environments (Z. Wu et al., 2023). These models were integrated with multi-object tracking algorithms to analyze poultry movement patterns, aiding in early detection of stress or health issues within flocks (Khairunissa et al.,

2021). Beyond behavior and health monitoring, object detection techniques have been applied to classify poultry droppings for non-invasive health assessment (Nakrosis et al., 2023) and to automate tasks such as egg detection and classification (G. Li, Chesser, et al., 2021; G. Li, Xu, et al., 2020), ensuring higher efficiency and precision in production chains (Fang et al., 2022). Together, these applications highlight the transformative impact of object detection models in modern poultry farming, enhancing scalability, precision, and automation in farm management practices. However, newer object detection models like YOLOv8 have yet to be thoroughly evaluated or effectively integrated into 3D sensing frameworks to achieve mobile and robust bird detection and kinematic features extractions.

Despite the growing use of smart monitoring systems in precision poultry farming, existing solutions face several critical limitations. Many commercially available systems are prohibitively expensive, require extensive modifications to existing housing structures, and primarily rely on 2D imaging, which suffers from limitations in capturing detailed phenotypic and behavioral traits in challenging poultry environments. Furthermore, these systems often rely on continuous internet connectivity and cloud-based storage, which are not feasible in many broiler production settings due to infrastructure constraints in rural areas. Additionally, while some research has explored 3D vision systems, these efforts typically lack affordability, portability, or scalability, making them unsuitable for large-scale or budget-constrained operations. Appropriate sensing modalities integrated into detection algorithms, seamless integration between 3D sensing systems and control schemes, and effective algorithms for extracting movement characteristics of individual animals remain under exploration. This study addresses these gaps by developing a cost-effective, portable, and scalable 3D monitoring system tailored to the specific environmental and operational conditions of broiler production in the United States. By integrating advanced RGB-D imaging with optimized data management and automated alert systems, this work offers a practical and innovative solution that advances the field by bridging the gap between high-performance monitoring and real-world affordability and accessibility.

The objective of this study was to develop a low-cost intelligent system for monitoring the 3D features of poultry. The system was designed to accommodate the specific characteristics of US broiler housing (especially along broiler belts), such as limited Internet access, dense bird populations, and dusty and humid environments. A series of systematic experiments and evaluations were performed and reported, including comparing various RGB-D camera models, testing multiple installation heights, and assessing the impact of different data modalities under varying environmental conditions, including both normal and extreme lighting scenarios. Additionally, field testing was conducted to validate the system's performance in real-world broiler chicken monitoring scenarios. These evaluations establish a comprehensive framework for analyzing the relationship between system parameter settings and effective 3D feature monitoring.

3.2 Materials and Methods

3.2.1 System Design and Integration

An intelligent system was designed to support real-time data recording, processing, and display via wired/wireless data transfer. A plastic box (13 cm deep × 21 cm wide × 29 cm long) was used to house all necessary components, including a mini-PC, extension cable, HDMI cable, power cords, and small fans (Table 3.1).

Table 3.1: List of unit prices and components used to build the intelligence system. (Note: * represents that the item is optional.)

Component	**Model number**	**Function**	**Unit price ($)**
3D camera	Intel RealSense L515	Sensing and image/video acquisition	589.00
Mini-PC	CyberGeek Nano J1	Data acquisition and edge computing	145.99
Junction box	QL-281913AGH	Housing components	56.69
Extension Cable	B08QRP6CGF	Extending connection between the box and camera	15.99
USB Hub	UGREEN CM481	Expanding number of connections with the PC	6.99
Camera Wall Mount Arm	Selens B09GBGD2SH	Adjusting installation height of the 3D camera	45.99
Power Extension Cord	TROND B0B497LDDW	Powering the device	6.11
HDMI Cable	Snowkids B099ZR82TM	Connecting the PC with the touchable screen	9.99
Power Strip	Woods 0600 W	Extending the power connection	9.50
External HDD	WDBWLG0200HBK-NESN	Offline data storage	334.89
Touchable Monitor*	ZSCMALLS ZSCM18	Visualizing running programs and interfaces	99.99
Wireless Keyboard w/ Touchpad*	Logitech K400 Plus	Inputting commands	27.30
		Total cost w/o optional items	1,221.14
		Total cost (all items)	1,348.43

The specifications of the mini-PC (11 cm long × 12 cm wide × 4 cm high) weighing at 940 g included the processor of 2 GHz Celeron, 8 GB LPDDR4 RAM, 256 GB SSD hard drive, Intel UHD Graphics Coprocessor, 32 MB Graphics Card Ram Size, and Linux operation system (Ubuntu 20.04 LTS, known as Focal Fossa). The 3D camera (Intel RealSense L515) uses LiDAR to sense objects ranging from 0.25 to 9.00 m and has maximal depth outputs of 1024×768 pixels and maximal RGB outputs of 1920×1080 pixels. The maximal sampling rate is 30 fps, and it weighs 100 g. The intelligent system featured lightweight and portability. With all necessary components integrated into the plastic box, the 3D camera can be connected via extension cables and mounted at any desired location. ROS Noetic, integrated with the camera drivers, was used to run the camera and to capture, save, and replay the RGB-D streams. Figure 3.1 shows an example of the intelligence system with a 3D camera attached to the ceiling with the assistance of a wall mount.

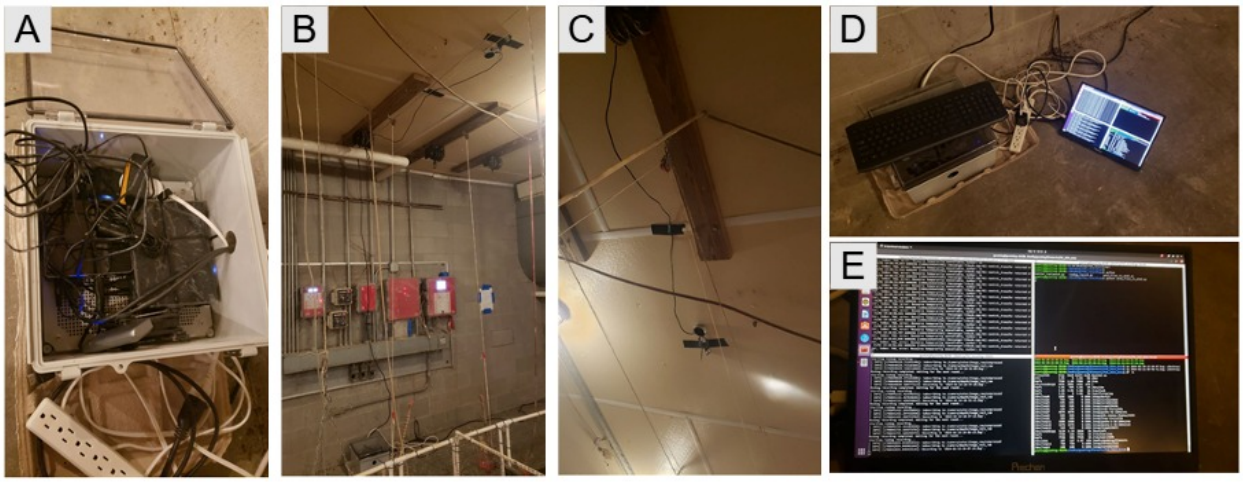

Figure 3.1: Component integration for the three-dimensional (3D) monitoring system in a broiler house. A. Integrated unit; B. Installation of the unit on the farm; C. 3D camera installed on the ceiling; D. & E. Touchable screen to visualize and debug the running program inside the unit.

3.2.2 Evaluating Setup Configurations

To determine the optimal configuration for recording 3D features of broiler chickens, various setup configurations were initially evaluated, including the camera models used, the quality of RGB frames captured, and the camera installation heights. Two models of Intel RealSense cameras, D435i and L515, were comparatively evaluated. These cameras were selected for their ability to provide high-quality RGB and depth data, essential for accurately capturing the three-dimensional features of moving subjects. The quality of RGB frames was analyzed under two conditions: compressed and uncompressed. Compressed frames aimed to reduce storage requirements without significantly compromising the detail needed for effective analysis, while uncompressed frames provided the highest quality data. ROS uses a JPEG compression algorithm, which is a lossy compression method that reduces file size by approximating the image data but may lead to some loss of quality, especially in high-frequency details like edges and textures. It should also be noted that the recorded depth information was not influenced by the RGB frame quality, as RGB frames may be used to visualize objects recorded or assign textures to them. Installation heights were set at four levels: 2.25, 2.50, 2.75, and 3.00 m. These heights were chosen to assess trade-offs between the field of view, depth detail accuracy, and practical feasibility in an operational farm environment. A robotic vehicle (31.5 cm long × 19.8 cm wide × 12.7 cm high) was employed to simulate the movement of broiler chickens in a controlled setting, allowing for precise evaluation of RGB and depth data quality from each setup configuration. Figure 3.2 shows samples of RGB frames of the robotic vehicle taken from the four heights using the Intel RealSense L515 camera.

This approach ensured that equipment and setting selections were grounded in robust, empirical data, reflecting real-world conditions as closely as possible. Figure 3.3 provides a visual representation of these

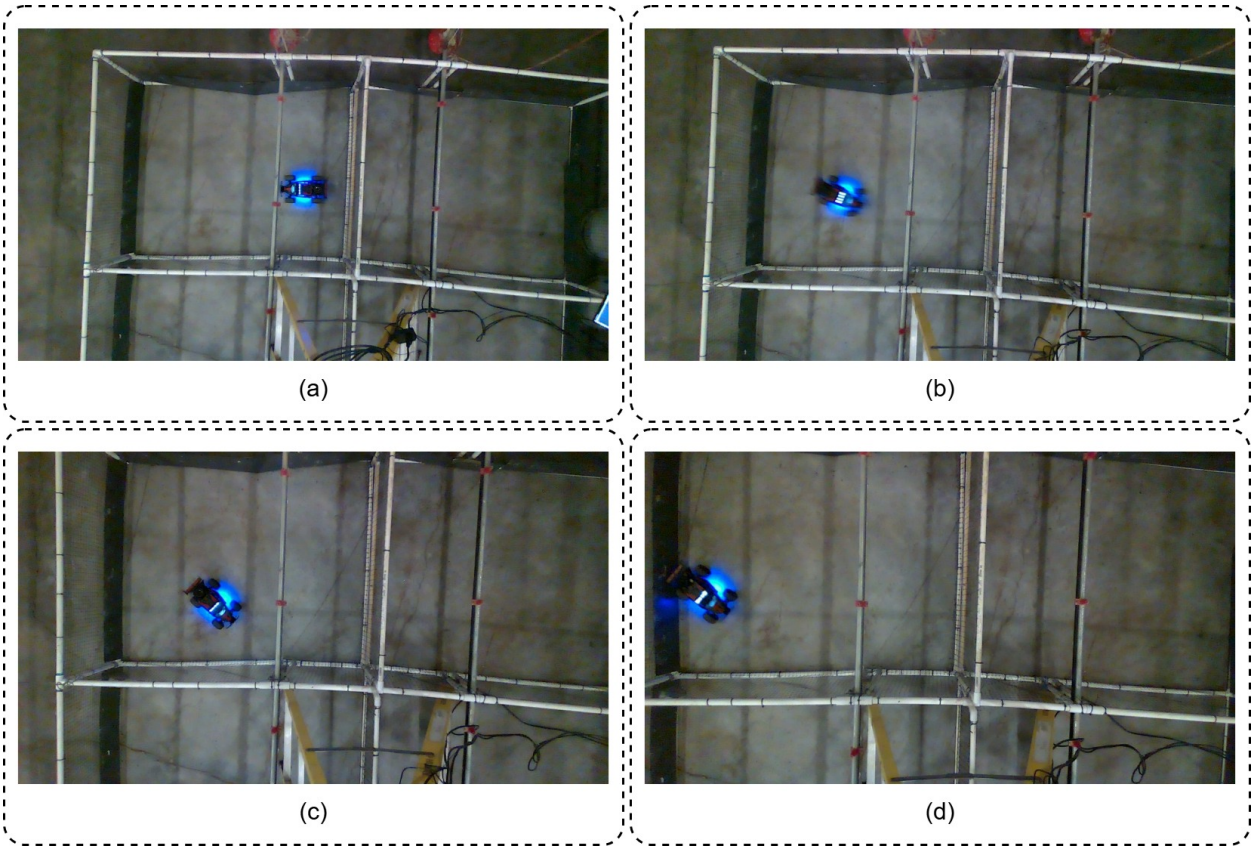

Figure 3.2: RGB frames capturing a robotic vehicle moving in an empty broiler pen. An Intel RealSense L515 camera was used to record these frames from four different heights: 3.00 m (a), 2.75 m (b), 2.50 m (c), and 2.25 m (d).

setup configuration choices, illustrating the experiments and parameters tested. Each configuration was systematically assessed to ensure the selected setup would maximize the quality and reliability of the data collected.

To select the best setup configuration for the intelligent system, a structured experimental process was employed as detailed in the flowchart of Figure 3.4. The process was iterative, enabling a comprehensive evaluation and comparison of configuration options to ensure the highest data quality and system efficiency. The following steps outline the methodology used to determine the optimal setup configurations:

- **Experiment Configuration:** Each experimental round began by setting up a distinct combination of the camera model (D435i or L515), camera installation height (2.25 m, 2.50 m, 2.75 m, or 3.00 m), and RGB image quality (compressed or uncompressed). This step was critical for assessing the impact of different variables on the system recording performance and data quality.

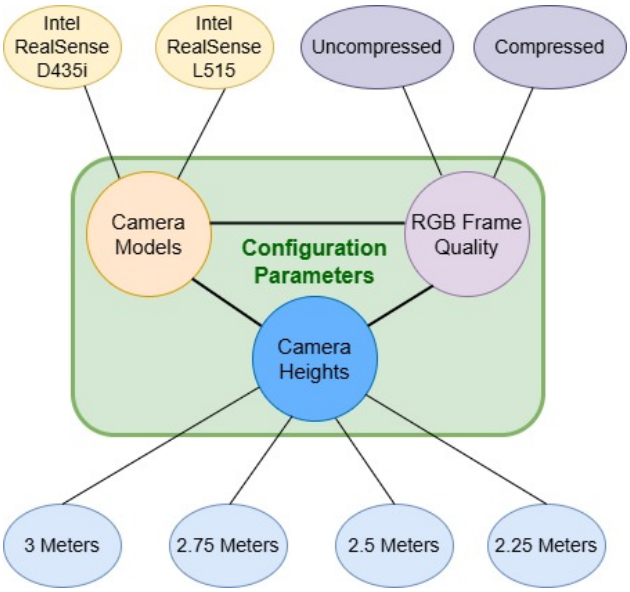

Figure 3.3: Different parameters examined to find the best combined configuration for setting up the intelligent system for monitoring three-dimensional features for broiler chickens.

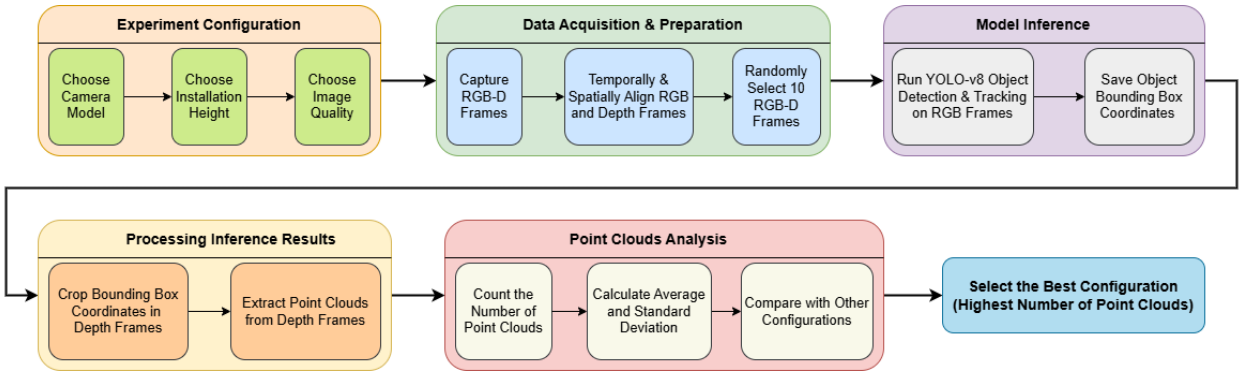

Figure 3.4: Overall algorithm workflow for analyzing and comparing different experiment configurations to find the best combined configuration for the intelligent system for monitoring three-dimensional features for broiler chickens.

- **Data Acquisition and Preparation:** With the experimental setup in place, the data acquisition phase commenced by gathering RGB-D (color and depth) data with the mini-PC listed on Table 3.1. Data acquisition was conducted in an experimental broiler room containing empty pens (1.2 m wide × 3.0 m long × 1.0 m high for each pen). Temporal and spatial alignment of the corresponding recorded RGB and Depth frames was performed to ensure consistency in subsequent analyses. From the aligned dataset, we randomly selected 10 representative pairs of RGB and Depth frames for further processing.

- **Model Inference:** Using the recent object detection and tracking technology, You Only Look Once Version 8 (YOLOv8; (Jocher et al., 2023)) and ByteTrack (Y. Zhang et al., 2022), the RGB frames were processed to identify and track objects within the video stream. The output from this step included coordinates for bounding boxes of the tracked robotic vehicle consecutively throughout the frames, which were saved for further analysis. It should be noted that the pretrained object detection model was used for this purpose, as it was previously trained on large datasets to recognize common objects like cars. In this case, the robotic vehicle was identified and tracked with 100% accuracy.

- **Processing Inference Results:** The bounding box coordinates from the RGB data were used to crop and map corresponding areas in the depth frames. This allowed for the targeted extraction of point clouds within these regions for focused analysis.

- **Point Clouds Analysis:** Within the cropped areas, the point clouds were analyzed by counting the total number of points and calculating both the average and standard deviation of point counts across the selected 10 frames. The results were presented as average ± standard deviation. This statistical analysis was crucial to evaluate the quality and consistency of the data captured under each configuration.

- **Comparison and Selection:** After analyzing the data from various configurations, the average number of point clouds (Eq.3.1) and their distribution characteristics across all tested setup configurations were compared. The configuration that yielded the highest average number of point clouds, indicating the configuration that captured the most detailed and reliable data, was selected as the optimal choice.

$$\text{Number of points} = \sum_{i=1}^{N} \mathit{1}\{(x_i, y_i, z_i) \neq 0\} \tag{3.1}$$

  where, i is the i-th point in the point cloud; *1* is the number, 1; (x_i, y_i, z_i) are the 3D coordinates in a point cloud.

This meticulous process ensured that the chosen configuration was grounded in empirical evidence, demonstrating optimal performance in terms of data quality and operational feasibility for broiler chicken monitoring in a practical setting. Multiple Python packages and libraries were used to perform the experiments, including (but not limited to): `ultralytics`, `cv2`, `rosbag`, `cv_bridge`, `numpy`, `json`, `argparse`, and `os`.

All experiments and system examinations were conducted in the Poultry Research Center at the University of Georgia. The center is located in the southern part of Athens, Georgia. Experiments took place between February and June 2024.

3.2.3 Determining Optimal Modality Combination

Three-dimensional data files are typically very large (>60 GB per hour at 15 fps) as additional depth information is included in every frame's spatial location. It is necessary to evaluate whether depth modality is needed for the recording. To effectively integrate depth data with the YOLOv8m object detection and tracking model, which supports only three-channel data inputs, RGB and Depth data were adapted into several three-channel formats with various modality combinations. These adaptations preserved depth information while conforming to the architectural constraints of the YOLOv8m model. The modality combinations tested included standard RGB (Red, Green, and Blue), DGB (Depth, Green, Blue), RDB (Red, Depth, Blue), and RGD (Red, Green, Depth). Datasets were created for each of the modality combinations under four lighting and noise conditions to evaluate their performance in typical scenarios of broiler farms. Four scenarios were deployed to examine the model's performance under the four modality combinations (depicted in Figure 3.5).

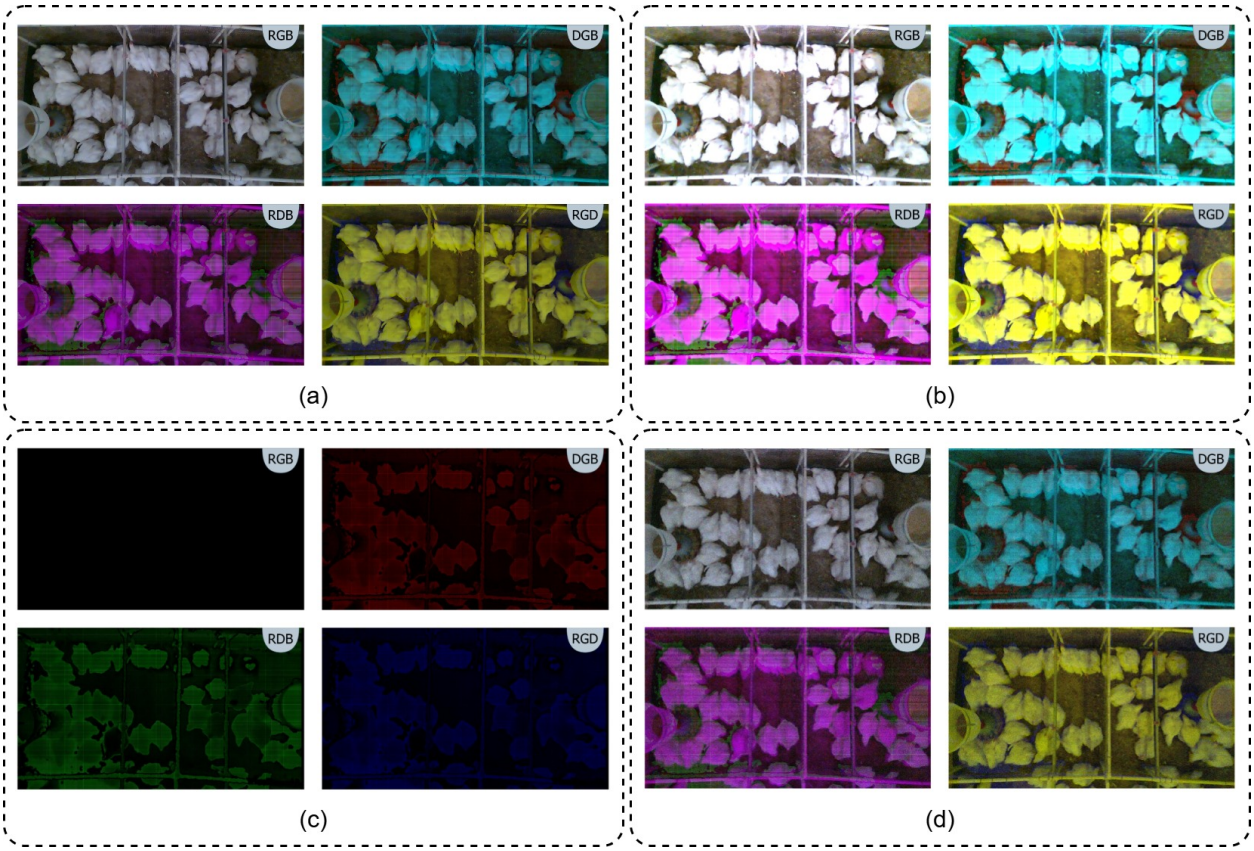

Figure 3.5: Samples of recorded frames of broiler chickens across various lighting conditions and data modality combinations: (a) Frames with normal lighting with no noise, highlighting how the absence of color channels affects the image hue but retains sufficient information for broiler tracking; (b) Frames with overexposed color channels; (c) Frames under deficient light intensity (<1 lux); (d) Frames with noisy color channels (Gaussian noise).

Normal conditions served as the baseline and included typical light intensities (10–20 lx) without noise, which simulated general recording conditions in broiler houses. Overexposed scenarios were created by suddenly doubling the light intensity in the color channels, which simulated the excessive lighting conditions when the lights were suddenly turned on or off on farms. Dim light conditions were extracted from nighttime recordings with less than 1 lx of light intensity. Under such conditions, depth data captured via infrared was the only reliable source to monitor objects of interest. The scenario simulated farm situations such as sudden power outages or nighttime. The fourth scenario was to include artificial noises in the frame to assess each modality combination's resilience against electronic noises, which can degrade image quality. The addition formula is presented in Eq.3.2.

$$N(x, y) = I(x, y) + M(x, y) \cdot G(0, 25) \tag{3.2}$$

where $N(x, y)$ represents the intensity value at coordinates x and y in a noisy image; $I(x, y)$ is the intensity values in an original image; $G(0, 25)$ denotes the Gaussian noise (salt-and-pepper type) with a mean of 0 and a standard deviation of 25; and $M(x, y)$ is the intensity value in a mask generated by Bernoulli process where each pixel is assigned 1 with a probability of 0.1, otherwise 0. The inclusion of $M(x, y)$ ensures that only a certain percentage of the pixels, specifically 10%, are affected by the noise, reflecting the random nature of electronic noise in imaging systems.

To evaluate model performance for bird detection on these datasets, a pilot recording was performed in an experimental pen (1.2 m wide × 1.5 m long) with twenty 28-day-old Cobb 500 male broiler chickens. A total of 165 frames, including $6,347$ bird instances, were labeled using a semi-automatic labeling platform, Roboflow. Each chicken was annotated with a bounding box and assigned the class label 'chicken'. The dataset was split into 80% for training and 20% for testing. The YOLOv8m was trained with 300 epochs to optimize bird detection performance. The loss functions of bounding box detection and class identification, along with precision, recall, mAP50, and mAP50-95 curves, were reported during training and validation. precision, recall, F1-score, and accuracy were used to assess the test performance of the YOLOv8m detection and tracking pipeline under the abovementioned challenging conditions. The formulas for these evaluation metrics are provided in Eq.3.3 through Eq.3.6:

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{average\_precision}(i) \tag{3.3}$$

$$\text{IoU} = \frac{(\text{TrueBoundingBoxes}) \cap (\text{PredictedBoundingBoxes})}{(\text{TrueBoundingBoxes}) \cup (\text{PredictedBoundingBoxes})} \tag{3.4}$$

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}} \tag{3.5}$$

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}} \tag{3.6}$$

where i represents the ith class, N is the total number of classes, *mAP* denotes mean Average Precision, and *IoU* refers to Intersection over Union. mAP50 and mAP50-95 correspond to IoU thresholds of 0.5 and 0.95, respectively.

In object detection, the classification confidence score quantifies the likelihood that a detected object belongs to a specific class. This score is derived from the neural network's output and reflects the model's confidence in its classification. For each detected object, the network outputs a confidence score between 0 and 1, with higher scores indicating stronger confidence in the classification result. The classification confidence was computed using the softmax activation function, which converts the raw output scores (logits) of the neural network into a probability distribution across all possible classes. The softmax function for a class c is defined as:

$$P(c) = \frac{\exp(z_c)}{\sum_{k=1}^{C} \exp(z_k)} \tag{3.7}$$

where $P(c)$ represents the probability or confidence score for class c, z_c is the raw output (logit) of the neural network for class c, and C is the total number of classes. The numerator, $exp(z_c)$, calculates the exponential of the logit for class c, while the denominator sums the exponentials of the logits for all classes. This operation ensures that the confidence scores for all classes sum to 1, forming a valid probability. In the context of this study, the classification confidence score is displayed in Figures 3.12 to 3.15 corresponding to $P(c)$, providing an interpretable measure of how likely the detected object belongs to the assigned class. For example, a confidence score of 0.95 for a detected object bird indicates a 95% likelihood that the object belongs to the specified classchicken. These confidence scores are crucial for evaluating the reliability of the model's classifications and are particularly useful in applications where the accuracy of object classification directly impacts downstream analyses or decision-making processes.

To determine the best configuration, we employed ROS along with several Python packages, including sys, subprocess, re, signal, and os, RGB-D recording; cv2, numpy, and random for selecting 10 random RGB-D frames; ultralytics for running YOLOv8m and ByteTrack; and cv2, os, and numpy for processing inference results. Point cloud analysis was performed using numpy and various built-in Python functions.

3.2.4 Comparing Cost of Data Storage

The proposed intelligent system for monitoring 3D poultry features comprises multiple components that can be assembled quickly (less than 5 min per unit). Although the listed items (Table 3.1) perfectly worked for the experiments, each item can be replaced by other alternatives depending on the needs of the project or the availability of certain products. One of the priciest items was data storage, which may require frequent replacement as storage fills up with ongoing recordings. The cost of the offline recording method was compared to four commonly used cloud storage services: Amazon S3, OneDrive, Google Drive, and Dropbox. Amazon S3 is a scalable cloud storage solution by Amazon Web Services that offers data availability, security, and performance, ideal for businesses needing extensive data management capabilities. OneDrive, provided by Microsoft, integrates with Microsoft Office and Windows, supporting file access, sharing, and collaboration across devices. Google Drive, offered by Google, allows seamless file storage and

real-time collaboration, enhanced by its integration with Google's productivity tools. Dropbox facilitates file synchronization and sharing, with a user-friendly interface suitable for both individual and business use, ensuring secure and accessible data across platforms. The comparison was based on a daily recording volume of 1.65 TB.

A multi-dimensional qualitative comparison was also conducted for local and cloud storage methods, including average cost, data security, latency, setup complexity, monitoring complexity, scalability, accessibility, backup and recovery, and customization. Average cost is defined as the average cost of saving the recorded data per system per day in US Dollars. Data security measures the effectiveness of mechanisms in place to protect data against unauthorized access and threats. Latency is crucial for applications needing real-time processing and represents the time it takes to transfer the data from local storage to external storage. Setup complexity indicates the ease of installation and configuration, while monitoring complexity describes the effort required to maintain optimal system performance. Scalability is the system's ability to expand and handle increased data volumes. Accessibility gauges how easily data can be retrieved from the system. Backup and recovery assess the strategies for data duplication and restoration capabilities in the event of data loss. Finally, customization refers to the system's flexibility to be tailored to meet specific project needs. Collectively, these criteria help determine the most suitable storage option for ensuring effective data management for monitoring systems in broiler chickens.

3.2.5 Implementing Broiler House Testing

The intelligent system was tested in a broiler house experiment involving two rooms, with each measuring 17.2 m long × 11.4 m wide. Each room was divided into two rows, and each row contained 12 identical pens (1.2 m × 3.0 m). Each pen accommodated 37 Cobb 500 male broiler chickens, and a total of 1,776 birds were used for the 48 pens in two rooms. For optimal point cloud data acquisition, L515 Intel RealSense cameras were strategically installed in three pens per row at a height of 2.25 m. As explained previously, this height was determined as the most effective following extensive testing with the robotic vehicle to ensure the best captured data quality. Figure 3.6 shows the schematic layout of the 3D camera in a broiler room. The even distribution of these cameras ensured sufficient variation in lighting and ventilation was captured for testing the robustness of the intelligent system. Broiler management followed the industry guidelines. All procedures were approved by the Institutional Animal Care and Use Committees at the University of Georgia (protocol number: A2023 07-016-Y1-A0).

Above the specific pens, RGB-D cameras were mounted on the ceilings to continuously record detailed 3D data of broiler chickens. Each camera was connected to the intelligent system as described in the System Design and Integration section. Offline recordings were conducted to ensure that data can be appropriately secured without the influence of the Internet. Data were recorded consecutively over three days a week. Daily reports were tagged with the respective mini-PC names, enhancing clarity in communications sent to the system administrator. Figure 3.7 illustrates one of the two broiler rooms used for field tests.

Figure 3.8 shows the logic of 3D data acquisition and sending warnings. Data were recorded in 5-minute intervals, with each chunk approximately 5.7 GB in size. Initially, the data was stored locally on a

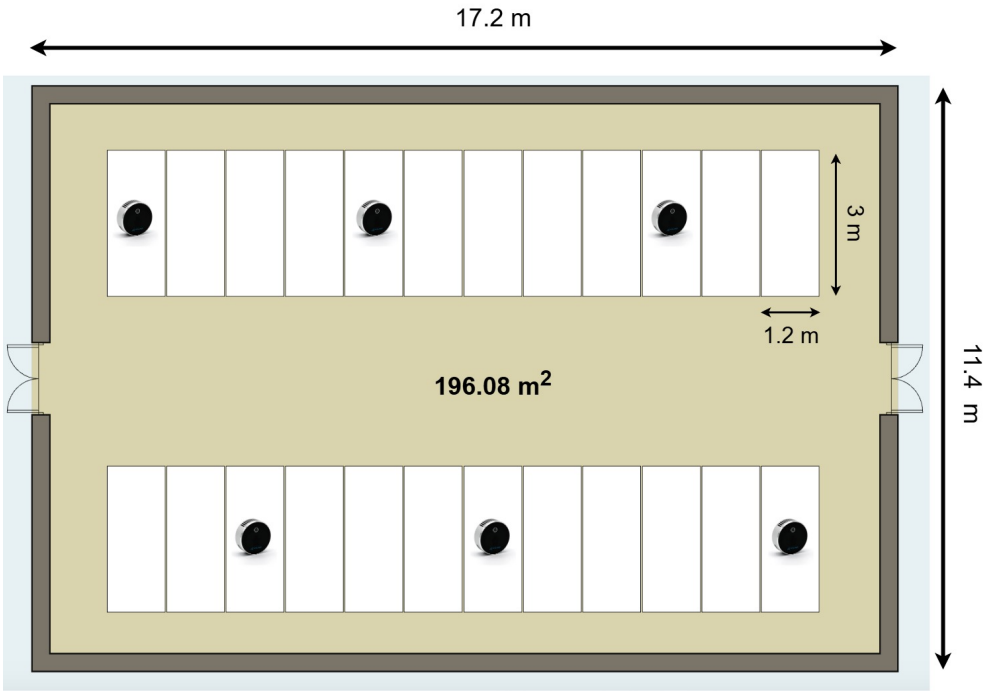

Figure 3.6: Schematic layout of a broiler house illustrating the distribution of pens and cameras.

mini-PC. A Python script continuously monitored the local directory, automatically transferring new files to an external hard drive to mitigate local storage constraints. Each transferred file was logged, ensuring precise tracking of the data migration. The system actively monitored the storage capacity of both the local and external drives. It employed remaining memory storage thresholds to trigger alerts when storage space was low, preventing data overwrite and loss. Specifically, if the local or external hard drive spaces became critically low (50 GB), the system sent a critically low space alert and halted the recording process to safeguard the integrity of the data. Conversely, if only low space (1 TB) was detected on the external drive, the program sent a low space alert to the system administrator but continued recording.

Additionally, the system generated a daily report every midnight, detailing the hard drive usage and remaining disk space on the external drive. This report offered a comprehensive summary of system status and was sent to the system administrator for review. The entire monitoring infrastructure was highly customizable. System administrators could choose their preferred method of notifications, such as email, Telegram, or Slack, based on their operational needs and preferences. Each notification and report included identifiers such as the PC name associated with the pen number being monitored, allowing for easy reference and management. This meticulous approach to data management ensured continuous and reliable monitoring of the chicken pens, enabling timely interventions and system adjustments based on real-time data storage and management needs.

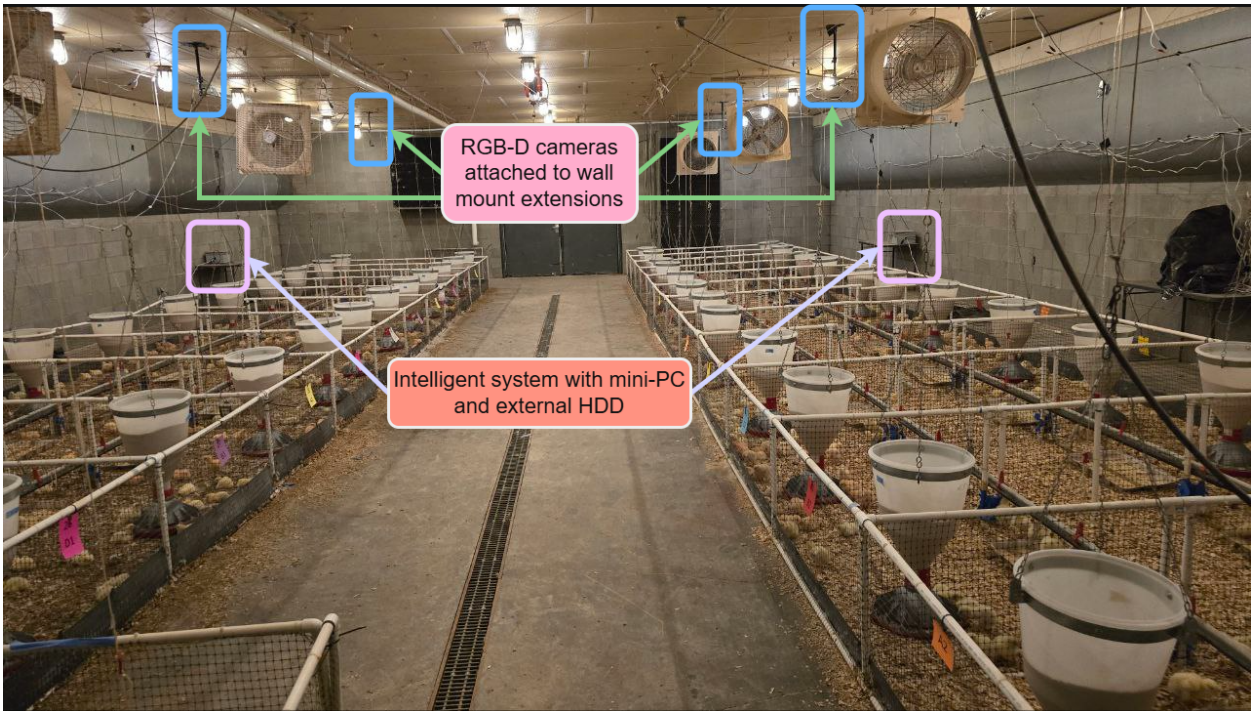

Figure 3.7: Operational setup within a broiler house showing the installation of the intelligence system and cameras.

ROS and several Python libraries were used, including (but not limited to) subprocess, sys, re, signal, os, shutil, time, datetime, smtplib, requests, socket, mimetypes, message, and netifaces for continuously recording the RGB-D streams, transferring them to the external storage, and sending warning alerts and daily reports to the system administrator.

3.2.6 Extracting three-dimensional movement trajectories of individual broiler chickens

The process of extracting 3D movement trajectories of individual broiler chickens from recorded RGB-D data followed a series of carefully designed steps to ensure maximum accuracy and relevance of the collected data. Initially, the RGB frame was processed using a YOLOv8m object detector that identified the broiler chickens and determined their bounding box coordinates. These coordinates were crucial as they directly corresponded to the targeted area in the depth image where point cloud data needed to be extracted. To align the RGB and depth frames effectively, intrinsic camera parameters were utilized. The RGB camera parameters were specified by the camera matrix (K_{color}, Eq.3.8) and distortion coefficients (D_{color}, Eq.3.9).

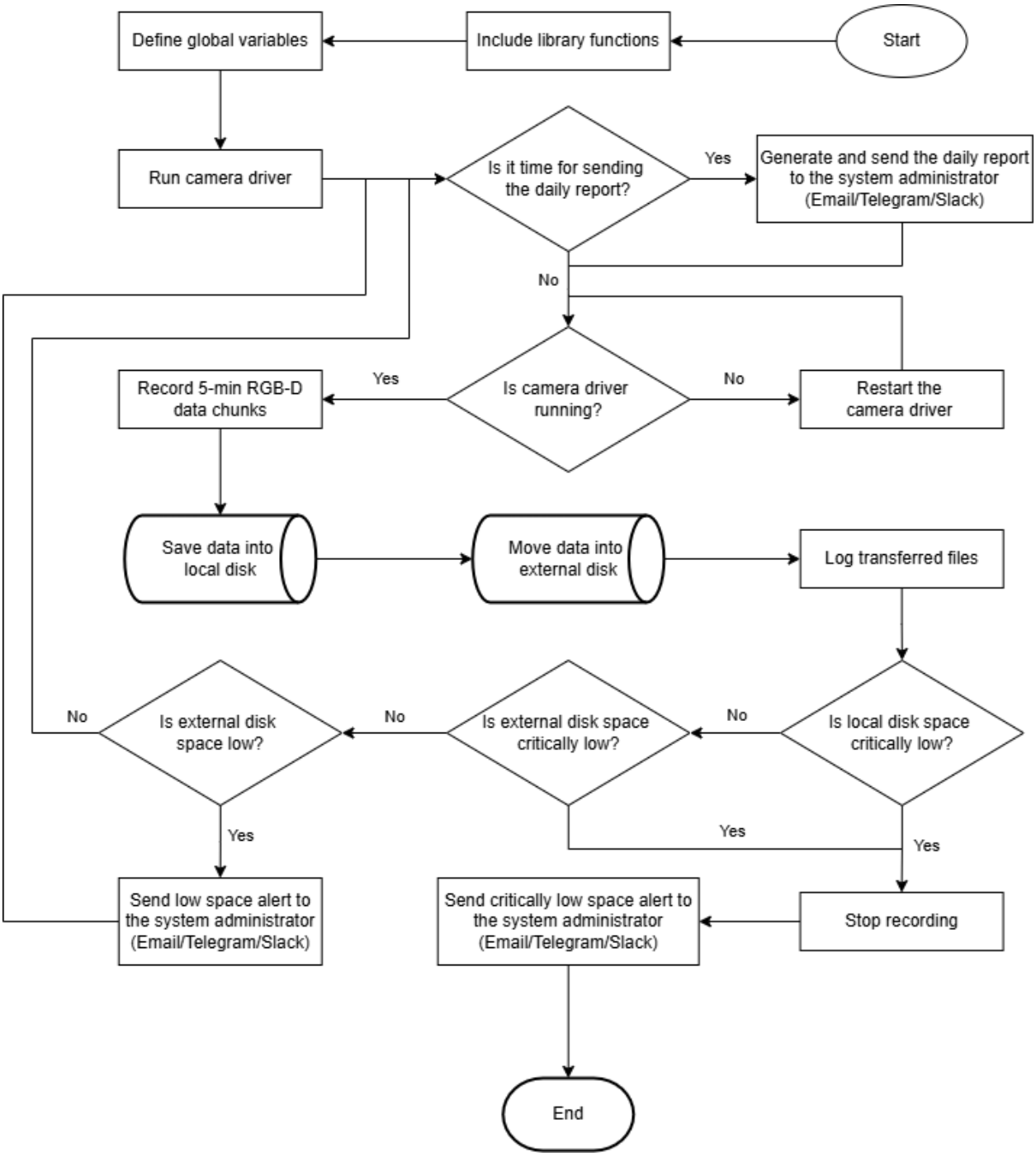

Figure 3.8: Flowchart for the logic of three-dimensional data acquisition and sending warnings. The threshold for low memory alert is 1 TB and the critically low memory threshold is 50 GB.

$$K_{\text{color}} = \begin{bmatrix} 901.3604 & 0 & 644.2163 \\ 0 & 901.9478 & 354.4529 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

$$D_{\text{color}} = [0.1605,\ -0.4705,\ -0.00099,\ 0.00157,\ 0.4175] \tag{3.9}$$

Similarly, the depth camera utilized its own set of parameters, comprising another camera matrix (K_{depth}, Eq.3.10) and distortion coefficients (D_{depth}, Eq.3.11).

$$K_{\text{depth}} = \begin{bmatrix} 456.4648 & 0 & 324.0938 \\ 0 & 457.5429 & 257.5586 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

$$D_{\text{depth}} = [0,\ 0,\ 0,\ 0,\ 0] \tag{3.11}$$

Subsequent to these configurations, Eq.3.12 is used to map pixel coordinates from the RGB frame (p_{color}) to the depth frame (p_{depth}):

$$p_{\text{depth}} = K_{\text{depth}}^{-1} \times p_{\text{color}} \tag{3.12}$$

Meanwhile, depth value (Z) can be extracted from the detected bounding box centroid coordinates in RGB frames ($D(x, y)$) following Eq.3.13:

$$Z = D(x, y) \tag{3.13}$$

And the point cloud data is converted into 3D space using Eq.3.14:

$$P = Z \times p_{\text{depth}} \tag{3.14}$$

The transformation formulas enable alignment of depth data with the RGB image coordinate system. RGB-D frames were aligned using intrinsics from the Intel RealSense cameras, recorded with the frames to ensure synchronization. Empirical checks on paired frames confirmed alignment accuracy, though no additional algorithmic verification was performed. Following alignment, the point cloud data was extracted from the depth image. To minimize background noise (e.g., ground surface point clouds), a threshold was set based on the camera's height minus 3 cm. This filtering ensured that only relevant point clouds were analyzed. With the system set up, YOLOv8m and ByteTrack continued to detect and track the broiler chickens across subsequent frames, allowing for ongoing point cloud extraction and counting. The process of 3D data extraction and visualization of the movement trajectories for the top-5 chickens with the highest total distance travelled in the pen is presented in pseudocode form as Algorithm 1. The algorithm was implemented using multiple Python packages, including but not limited to: `ultralytics`, `cv2`, `os`, `numpy`, `argparse`, `math`, `matplotlib`, `json`, and `rosbag`.

To evaluate the system performance during the broiler house testing, the following metrics were directly recorded by Python Scripts, including data (GB) recorded over 24 h of a day, file counts over 24 h of a day, and remaining space (TB) from June 7th to June 27th, 2024. The processing time was computed in Eq.3.15.

$$Processingtime = T_{\text{end}} - T_{\text{start}} \tag{3.15}$$

where T_{end} and T_{start} are the ending and starting timestamps reported by the Python time package. Processing speed was calculated by dividing the total processing time by the number of processed files.

Algorithm 1: Visualizing 3D Movement Trajectories of Chickens

Data: Synchronized RGB and Depth frames, camera intrinsic parameters
Result: Visualized 3D movement trajectories of chickens on XY and YZ planes

Initialization:
Set `rgb`, `depth`, `cam_int`, `track_IDs` as arrays ; // Use arrays for sequential data
Set `dets`, `cents`, `dists` as dictionaries ; // Use dictionaries for unordered data
Read RGB data array, `rgbs` ; // Save RGB frames in `rgbs` array
Read Depth data array, `depths` ; // Save Depth frames in `depths` array
Obtain `rgbs` length, `l` ; // Obtain number of RGB frames
Load YOLOv8 model with ByteTrack, `yolo_bt` ; // Load detection & tracking models
for $i \leftarrow 1$ **to** l **do**
    Set `temp_dets` as dictionary ; // Use a dictionary to save temporary detections
    Set `Temp_dist` as 0 ; // Set temporary distance to the previous centroid to 0
    `rgb, depth = alignFrames(rgbs[i], depths[i], cam_int)` ; // Align frames
    `temp_dets = detectAndTrack(yolo_bt, rgb, depth)` ; // Detection & tracking
    Obtain `temp_dets.keys()` length, `ld` ; // Obtain number of detected chickens
    for $j \leftarrow 1$ **to** ld **do**
        `cents[i][j] = findCentroid(dets[i][j])` ; // Find centroid of BBox
        if $i > 1$ **then**
            `Temp_dist = calcDistance(cents[i][j], cents[i-1][j])`;
            `dists[j] = dists[j] + Temp_dist` ; // Find & sum traveled distances
`track_IDs = sortIDsByDistance(dets, dists)` ; // Sort IDs by distance (descending)
for $i \leftarrow 1$ **to** 5 **do**
    `plotTrajectory(cents, 'XY')` ; // Plot top-5 movement trajectories in XY plane
    `plotTrajectory(cents, 'YZ')` ; // Plot top-5 movement trajectories in YZ plane

3.3 Results

This section summarizes the systematic evaluation of the proposed system in different scenarios. The findings begin with an assessment of hardware configurations, focusing on camera models and installation heights to maximize data capture quality. Next, object detection performance is evaluated across various data modalities under diverse lighting conditions to ensure robustness. This is followed by a comparison of storage options to optimize cost-effectiveness in data management. The system's reliability and efficiency were then tested under real-world field conditions. Finally, the extraction of 3D movement trajectories demonstrates the practical application of the selected configuration.

3.3.1 Equipment comparison and selection for system configuration

This section begins by evaluating hardware configurations to identify the best setup for capturing 3D poultry data. To find the most suitable camera for poultry monitoring, we compared key specifications of the Intel RealSense L515, Microsoft Azure Kinect, and other widely available 3D cameras, including the D435i, which includes an Inertial Measurement Unit (IMU) for motion tracking. Table 3.2 summarizes the evaluation, emphasizing criteria most relevant to this application, such as depth frame quality, size, environmental resistance, and additional sensing capabilities.

Table 3.2: Comparative analysis of RGB-D cameras for poultry monitoring.

Criteria	**Intel RealSense L515**	**Microsoft Azure Kinect**	**Zed 2**	**Oak-D Pro**	**Intel RealSense D435i**
Depth Sensing Technology	LiDAR	Time-of-Flight (ToF)	Passive Stereo Vision	Active Stereo Vision	Active Stereo Vision
Depth Frame Quality	High (low noise, close range)	High (low noise, mid-range)	Moderate to High (varies)	Moderate (low in low-light)	Moderate (more noise)
RGB Resolution (pixels)	1920 × 1090	3840 × 2160	4416 × 1242	1920 × 1080	1920 × 1080
Depth Resolution (pixels)	1024 × 768	1024 × 1024	4416 × 1242	1280 × 720	1280 × 720
Field of View (H/V, °)	70 / 43	120 / 120	110 / 70	81 / 52	87 / 58
Depth Range (m)	0.25–9	0.5–12	0.2–20	0.2–4	0.1–10
Power Consumption	Low	High	Moderate	Moderate	Low
Size and Weight (g)	Compact, 100	Larger, 440	Compact, 159	Compact, 150	Compact, 72

The Intel RealSense L515 was selected due to its superior close-range depth accuracy, compact size, and low power consumption, all of which are critical for monitoring small or closely spaced objects in poultry farm environments. While the Microsoft Azure Kinect offers a broader field of view, higher RGB resolution, and better water resistance, these features are less essential for this specific application. The D435i, though more affordable and equipped with an IMU, exhibited higher noise levels in depth frames and reduced depth accuracy compared to the L515, particularly in low-light conditions. The modularity of the proposed system allows for integration of cameras like the D435i or Azure Kinect if motion tracking or extended environmental resistance is prioritized in future applications. Due to practical constraints, we

used two of the cameras listed in Table 3.2, the Intel RealSense L515 and D435i, to represent distinct price points, depth-sensing capabilities, and underlying technologies. The L515 was chosen for its superior depth range and minimal sensing noise, while the D435i provided a more budget-friendly alternative with lower depth sensing performance. This comparison allowed us to evaluate distinct depth-sensing technologies and understand trade-offs between cost and performance for practical RGB D monitoring applications.

Comparison of Depth-Sensing Cameras in the Integrated System: The integrated system utilized a mini-PC and Intel RealSense cameras (L515 with LiDAR, D435i with stereo vision) to capture RGB-D data, enabling a comparison of depth-sensing performance across four installation heights (2.25, 2.50, 2.75, 3.00 m). Point cloud extraction began with identifying the robotic vehicle in RGB frames using the YOLOv8m object detector and ByteTrack algorithm. This state-of-the-art object detection model accurately positioned the coordinates of the bounding box encapsulating the vehicle. These coordinates were then used to locate the corresponding regions in the point cloud images derived from the depth data. Within this designated area, we extracted the point clouds, which consisted of sets of data points in space representing the surface of the vehicle. The extraction was meticulously carried out to ensure that only the points within the bounding box are considered, thereby isolating the vehicle from its surrounding environments. Subsequently, we counted the number of point clouds present in this isolated area to assess the density and distribution of the data points, which were critical for analyzing the vehicle's 3D features. The point cloud extraction process was also depicted and explained in Figure 3.9.

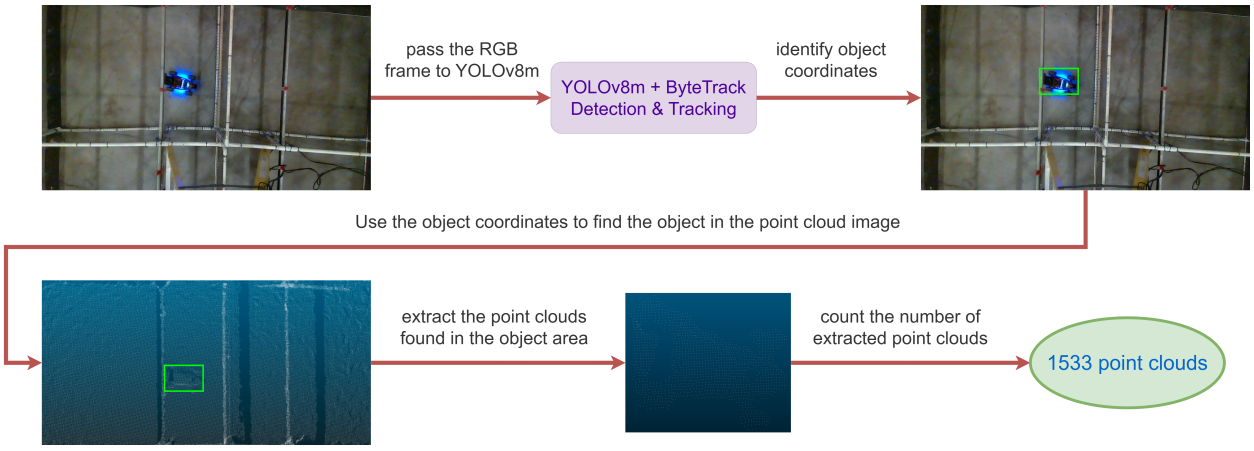

Figure 3.9: Workflow for extracting point cloud data from RGB-D frames using YOLOv8m object detection and tracking.

The average number of reconstructed point clouds for the vehicle at installation heights of 3.00, 2.75, 2.50, and 2.25 m was 951 ± 42, $1,208 \pm 91$, $1,435 \pm 101$, and $1,641 \pm 134$, respectively, for 10 trials using the L515 camera. Under similar conditions, the D435i camera captured 688 ± 54, 836 ± 82, $1,087 \pm 175$, and $1,255 \pm 208$ point clouds, respectively. These results are also presented in Table 3.3.

Table 3.3: Average number of points in the reconstructed vehicle point clouds (unitless).

Camera Installation Height (Meters)	**Intel RealSense D435i**	**Intel RealSense L515**
3.00	688 ± 54	951 ± 42
2.75	836 ± 82	1208 ± 91
2.50	1087 ± 175	1435 ± 101
2.25	1255 ± 208	$\mathbf{1641 \pm 134}$

In summary, the Intel RealSense L515 with an installation height of 2.25 was selected as the setup configuration, as it can capture more detailed 3D data. The prices of D435i and L515 are $334 and $589, respectively. Some sample depth frames are also presented for the two Intel RealSense cameras with four different installation heights in Figure 3.10 to show the data quality in different configuration setups.

Storage Optimization of Depth Data: Following the evaluation of depth-sensing performance, we assessed the strategies to manage the storage demands of depth data generated by the system. Depth files, stored as .npy files, can be compressed using various lossless and lossy compression techniques to optimize storage while preserving essential data integrity. We evaluated three compression methods:

- **Zlib Compression:** Applying zlib with Python's gzip module reduces the file size by approximately 30–40% without loss of accuracy.
- **Blosc Compression:** Using Blosc, optimized for NumPy arrays, achieves a reduction of up to 60%, depending on the sparsity and structure of the data.
- **Quantization with NumPy arrays:** Quantizing the depth data to 8-bit integers (instead of 32-bit floats) and then compressing with zlib results in a size reduction of up to 75%, with a negligible loss in depth accuracy (mean squared error < 0.01 for most datasets).

Preliminary experiments showed that compressing 1.5 TB of raw .npy depth files using zlib reduces the storage requirement to 914 GB, while Blosc compression reduces it further to 608 GB. Quantization followed by zlib compression reduces storage to 414 GB, demonstrating substantial efficiency gains. However, for tasks requiring fine-grained spatial precision in behavioral analysis, even minimal distortion could compromise accuracy. Therefore, compression was excluded for depth data. Only RGB frames were compressed to reduce storage utilization, as their major function was to assign textures to point clouds or visualize target objects.

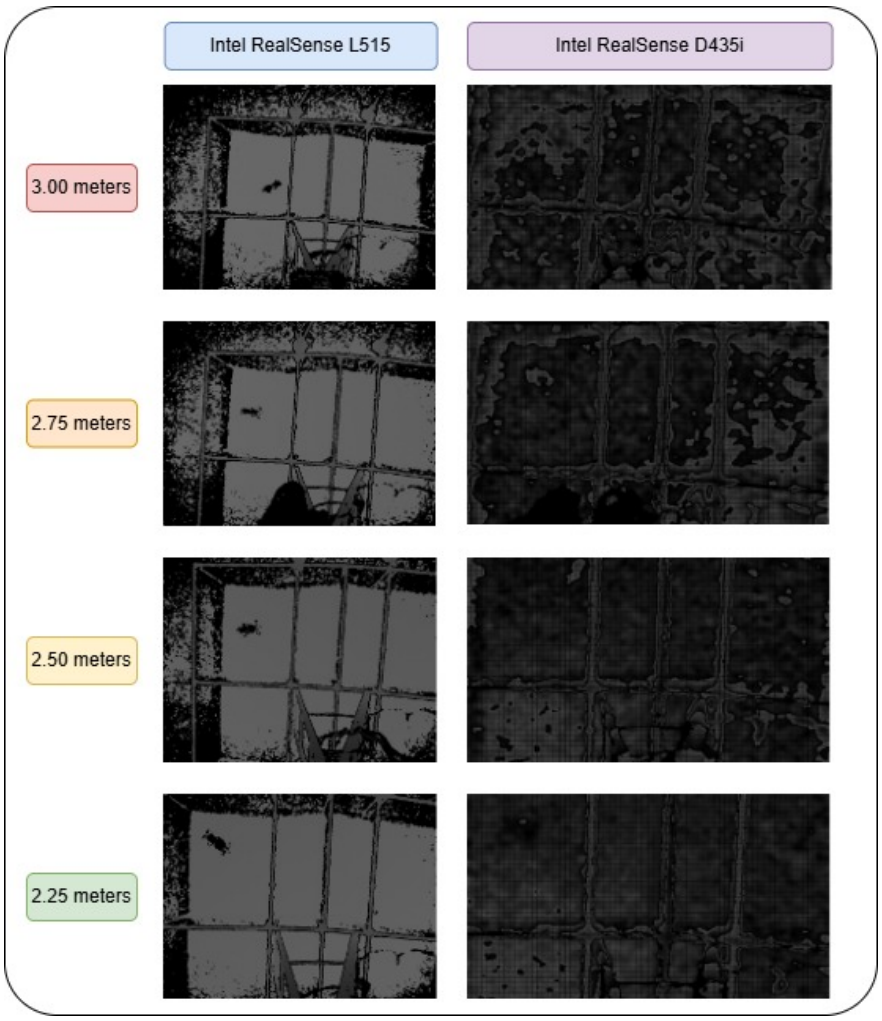

Figure 3.10: Sample depth frames from two Intel RealSense cameras in the robotic vehicle experiment at four installation heights (2.25, 2.50, 2.75, and 3.00 m from the ground floor).

3.3.2 Comparison of Intelligent System Performance in Monitoring Three-Dimensional Characteristics of Broiler Chickens Across Color Channel and Lighting Combinations

Following the identification of the best hardware setup, this section evaluates object detection performance across data modalities under varying conditions.

YOLOv8 was trained with a custom dataset including 165 images containing 6, 347 labeled chickens to identify broiler chickens to evaluate the performance of four modalities: RGB, DGB, RDB, and RGD under four testing scenarios: standard lighting without significant noise, extreme light overexposure, extremely low light intensity, and normal lighting conditions affected by Gaussian noise.

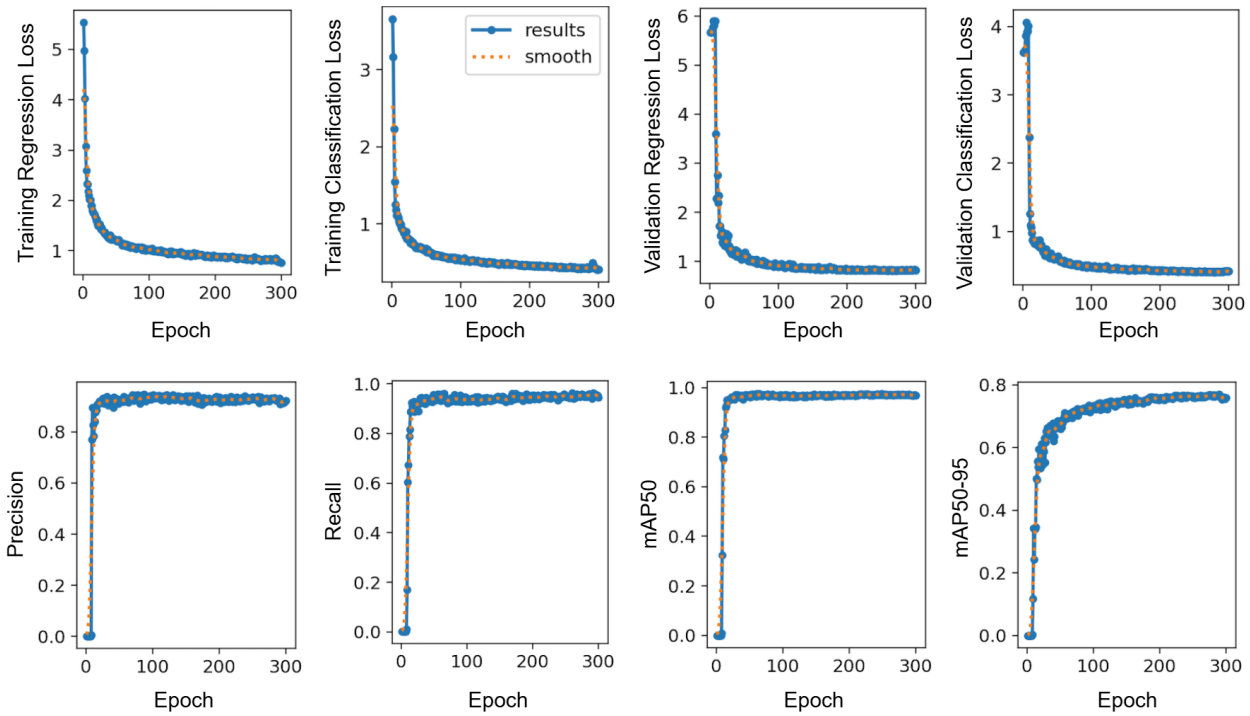

Figure 3.11: Performance Metrics of the Detection Model Across Training and Validation Phases

Figure 3.11 provides a comprehensive overview of the acquired performance on the evaluation metrics used to assess the YOLOv8m object detection model trained on the broiler dataset. These metrics include box regression and classification loss for both phases, alongside precision, recall, mean Average Precision (mAP) at IoU = 0.50, and mAP over IoU = 0.50 to 0.95. Each row of charts offers insight into different aspects of the model's performance across 300 epochs. The performance tended to converge around 100 epochs. Precision and recall were close to 1.0 at the end of the training, indicating high accuracy in identifying broiler chickens. It also suggested that the model was robust enough to examine the performance of various modality combinations.

The visual detection results are also shown in Figure 3.12. Each frame in the figure represents a different modality, showcasing bounding boxes around detected chickens. Adjacent to each bounding box label, 'chicken', the confidence scores are displayed, quantifying the model's certainty in its classifications. These scores ranged up to a maximum of 1.0, where a higher value indicates a stronger likelihood that the bird has been correctly identified by the model.

Table 3.4 outlines the model's performance in normal lighting conditions. It was observed that the RDB modality slightly outperformed the traditional RGB setup, achieving an accuracy of 89.4% and an F1-score of 87.2%. This improvement suggests that the addition of depth information alongside red and blue color channels can enhance the model's ability to discern and accurately identify broiler chickens in

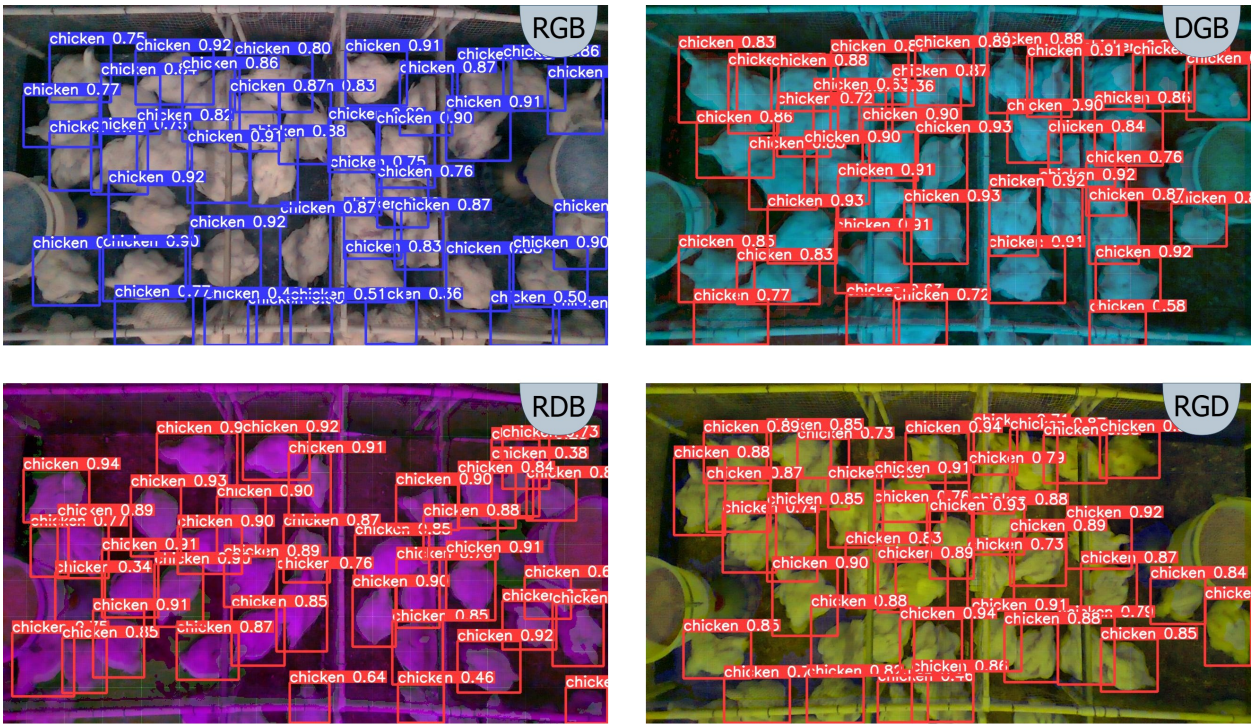

Figure 3.12: Illustration of chicken detection across four data modalities in normal light condition, as performed by the custom-trained YOLOv8m object detection model. R=Red, G=Green, B=Blue, and D=Depth. There is a number for each bounding box, representing the classification confidence score within the range of 0 to 1, meaning how much the object detection model is confident about its classification accuracy. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

crowded environments. The RGD modality also showed competitive results, although it demonstrated a slight decrease in precision compared to RGB and RDB, potentially indicating some trade-offs when all three channels are used. The depth component in RDB and RGD appears to provide a distinct advantage over RGB in environments where lighting and background conditions were controlled and consistent without noise and interference.

Table 3.5 showcases the resilience of the model with the four modalities and under conditions of extreme light overexposure. Here, the inclusion of depth information in the DGB modality proved particularly beneficial, achieving an accuracy rate of 88.2% and an F1-score of 85.5%. The RDB and RGD modalities maintained robust performance levels as well, with a higher recall of 88.4% and F1-score of 86.2%.

These enhancements indicate that depth channels can help mitigate the adverse effects of overexposure that typically wash out color details, leading to loss of feature information critical for bird detection. Figure

Table 3.4: Performance criteria of YOLOv8m object detection model applied on four modalities when the lighting condition is normal.

Criteria	**RGB**	**DGB**	**RDB**	**RGD**
Accuracy (%)	89.2	88.0	89.4	88.0
Precision (%)	84.8	83.4	86.4	84.8
Recall (%)	89.5	88.2	89.5	88.2
F1-score (%)	86.9	85.5	87.2	87.2

Table 3.5: Performance criteria of YOLOv8m object detection model applied on four modalities in extremely high light exposure.

Criteria	**RGB**	**DGB**	**RDB**	**RGD**
Accuracy (%)	86.8	88.2	87.7	87.5
Precision (%)	82.8	83.3	84.7	85.2
Recall (%)	87.1	88.4	88.4	87.5
F1-score (%)	85.1	85.5	86.2	86.2

3.13 illustrates the object detection bounding boxes detected by the custom-trained YOLOv8m model applied on a sample test frame in four modalities.

Table 3.6: Performance criteria of YOLOv8m object detection model applied on four modalities in extremely low light intensity.

Criteria	**RGB**	**DGB**	**RDB**	**RGD**
Accuracy (%)	0	88.2	86.0	88.2
Precision (%)	0	86.2	83.1	85.8
Recall (%)	0	87.8	86.0	88.4
F1-score (%)	0	86.4	84.4	86.5

Table 3.6 delves into performance metrics in extremely low light intensity situations, a condition that poses significant challenges for any vision-based system. The failure of the RGB modality, with 0% performance across all metrics, starkly highlights its dependency on adequate lighting. Conversely, modalities incorporating depth data showcased remarkable robustness; particularly, the RGD configuration stood out with an F1-score of 86.5%, suggesting that depth sensors that do not rely on visible light can provide critical information even in the dark.

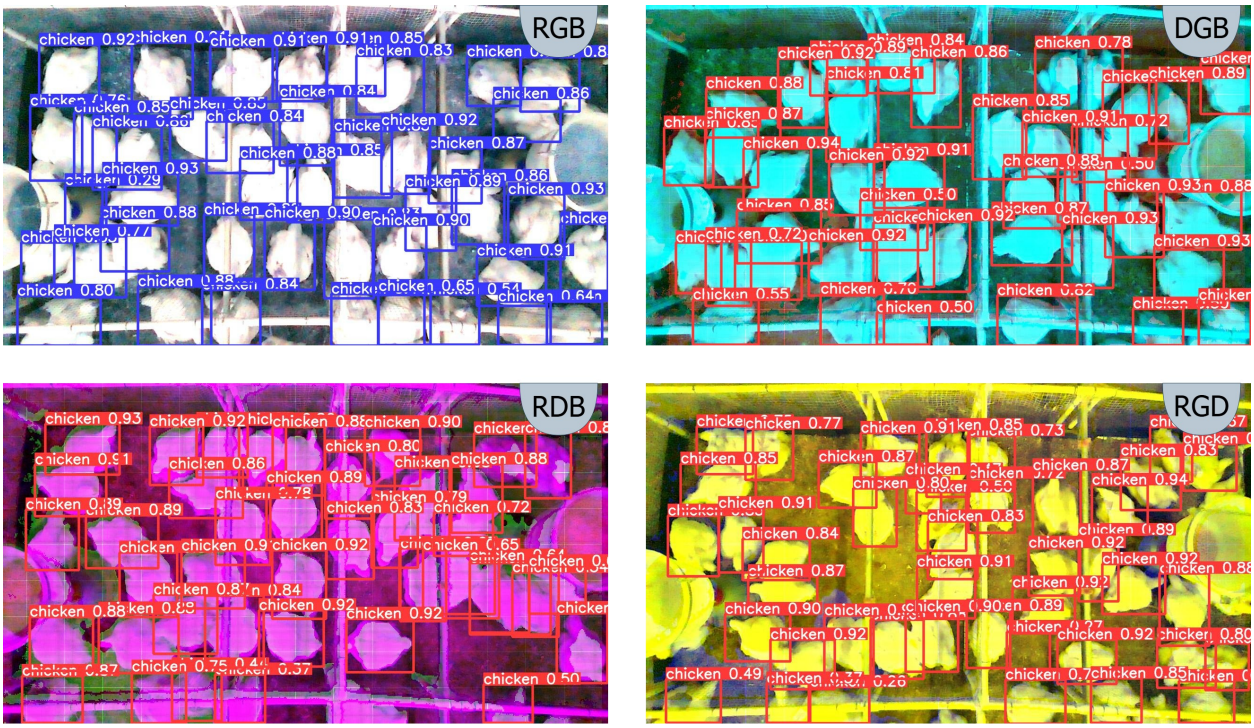

Figure 3.13: Demonstration of chicken detection across four distinct data modalities in overexposed lighting condition, as performed by our custom-trained YOLOv8m object detection model. R=Red, G=Green, B=Blue, and D=Depth. There is a number for each bounding box, representing the classification confidence score within the range of 0 to 1, meaning how much the object detection model is confident about its classification accuracy. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Figure 3.14 demonstrates an example of YOLOv8m object detection model performance on samples with extremely low light intensity in the four modalities.

Table 3.7 examines the impact of Gaussian noise on the color channels under normal lighting conditions, simulating scenarios where signal integrity might be compromised due to electronic noise or sensor anomalies. The depth-inclusive modalities again outperformed the RGB setup, with RGD showing the highest F1-score of 86.7%.

This superior performance can be attributed to the depth channel's ability to provide structural information about the environment and objects, which helps in compensating for the loss of detail in noisy color data. The enhanced robustness of RGD, RDB, and DGB in this scenario underscores the potential of depth data to augment color information, ensuring that object detection systems remain effective even when color channels are significantly degraded. Figure 3.15 shows the detected chicken bounding boxes in testing noisy samples in each of the four diverse modalities.

Table 3.7: Performance criteria of YOLOv8m object detection model applied on four modalities when the color channels are noisy.

Criteria	**RGB**	**DGB**	**RDB**	**RGD**
Accuracy (%)	85.1	86.5	88.4	88.4
Precision (%)	82.5	83.5	85.1	85.7
Recall (%)	85.4	87.5	88.1	88.5
F1-score (%)	83.7	85.5	86.3	86.7

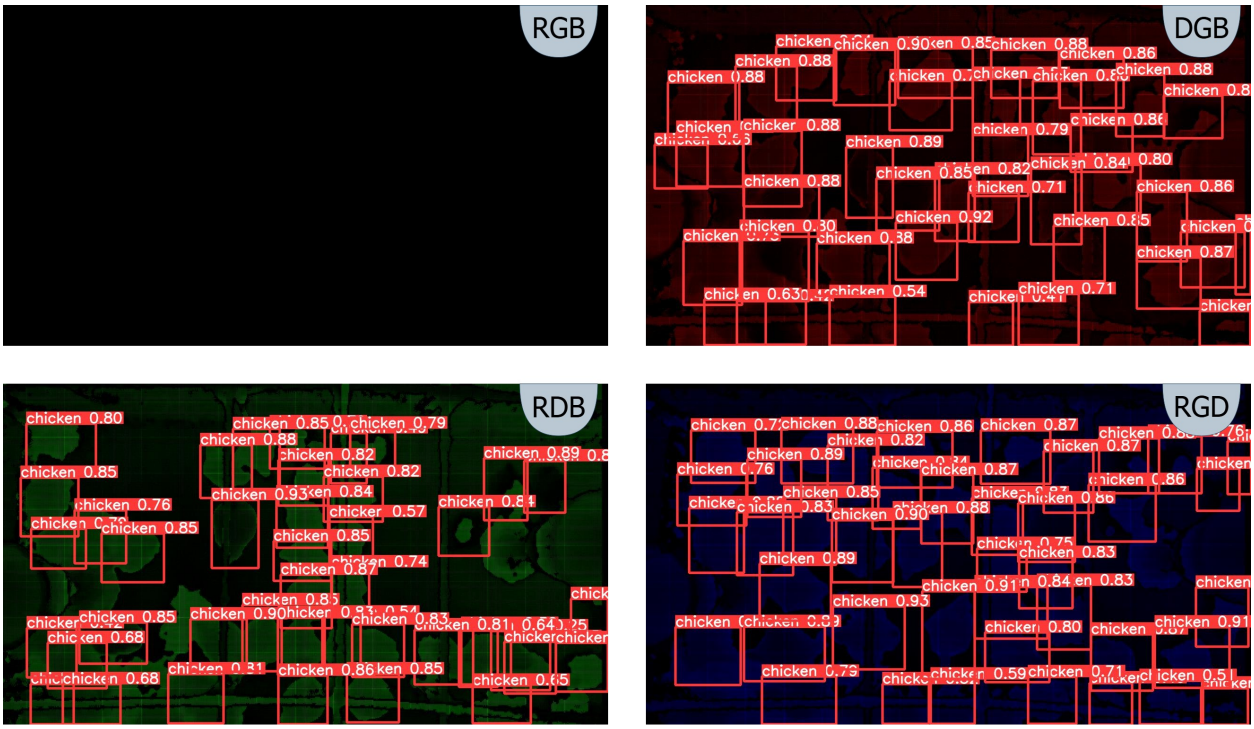

Figure 3.14: Visualization of chicken detection across four distinct data modalities in a very low light intensity condition, as performed by our custom-trained YOLOv8m object detection model. R=Red, G=Green, B=Blue, and D=Depth. There is a number for each bounding box, representing the classification confidence score within the range of 0 to 1, meaning how much the object detection model is confident about its classification accuracy. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

As mentioned before, the YOLOv8m model demonstrated robust performance in the chicken monitoring domain, achieving an accuracy of 89.2%, precision of 84.8%, recall of 89.5%, and an F1-score of 86.9% on the test set. We compared its performance in normal lighting conditions with three different state-of-the-art object detection models: SSD, EfficientDet, and Faster-RCNN. The training data and

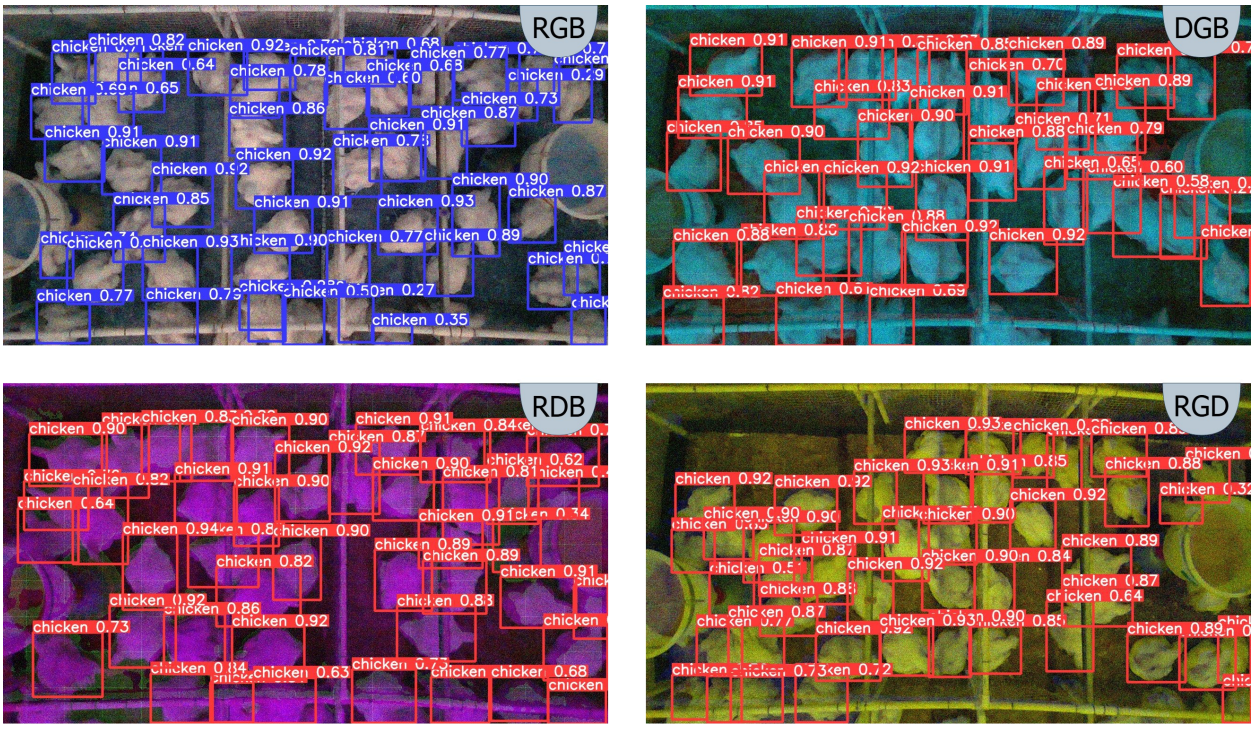

Figure 3.15: Depiction of chicken detection across four data modalities when color channels are with Gaussian noises, as performed by our custom-trained YOLOv8m object detection model. R=Red, G=Green, B=Blue, and D=Depth. There is a number for each bounding box, representing the classification confidence score within the range of 0 to 1, meaning how much the object detection model is confident about its classification accuracy. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

hyperparameters were all the same for a fair comparison. In comparison, SSD exhibited an accuracy of 83.4%, precision of 79.2%, recall of 81.5%, and an F1-score of 80.3%, reflecting its challenges in detecting small or complex objects in dynamic environments. EfficientDet, optimized for computational efficiency, achieved an accuracy of 85.1%, precision of 81.6%, recall of 83.8%, and an F1-score of 82.7%, showcasing its balance between performance and speed, though slightly underperforming in recall compared to YOLOv8m. Faster R-CNN, while known for high localization accuracy, achieved an accuracy of 87.3%, precision of 83.1%, recall of 84.6%, and an F1-score of 83.8%, with its two-stage detection process impacting real-time responsiveness. The results are also presented in Table 3.8. These comparisons highlight YOLOv8m's superior balance of accuracy, precision, and recall, making it a particularly effective model for poultry monitoring applications, especially in scenarios requiring real-time detection and tracking.

Table 3.8: Bird detection performance with various object detection models under normal lighting conditions.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
YOLOv8m	89.2	84.8	89.5	86.9
SSD	83.4	79.2	81.5	80.3
EfficientDet	85.1	81.6	83.8	82.7
Faster R-CNN	87.3	83.1	84.6	83.8

3.3.3 Cost Comparison for Data Storage

Building on data capture capabilities, this section assesses storage options to ensure cost-effective data management. The cost analysis for various storage options for our poultry monitoring system highlights significant differences in daily expenditures. Based on the calculations from the Python script, local external hard drives had a daily cost of approximately $28, making them one of the more expensive options compared to some cloud services. In stark contrast, OneDrive offered the most economical solution at just $4 per day, followed closely by Google Drive at $5 per day, reflecting their efficiency and affordability for extensive data storage over long periods. Dropbox, priced at $21 per day, presents a mid-range option, while Amazon S3 stands as the costliest at $38 per day. The comparison of the cost of data storage per day is visualized in Figure 3.16. In a typical 49-broiler production cycle, if the data are consecutively recorded, it may cost $1,372 for external hard drives, $1,862 for Amazon S3, $196 for OneDrive, $245 for Google Drive, and $1,029 for Dropbox.

Table 3.9 provides an in-depth comparison of local storage devices with cloud storage services. Choosing between local and cloud storage requires a careful assessment of various factors beyond cost, such as data security, latency, accessibility, and scalability. While cloud storage options like OneDrive and Google Drive offered remarkable cost efficiency and accessibility, they may involve higher latency and potential risks in data security due to reliance on third-party management. Local storage, though more expensive per day, provided advantages in latency reduction, crucial for real-time data processing and immediate access needs. Additionally, local storage ensured enhanced control over data security, as the data remains confined to physical devices within the premises, minimizing external breach risks, for instance, outage and Internet disruption. In sum, based on our priorities of low-latency access and data security, local storage was selected. However, users with stable internet access and cost constraints may opt for cloud-based alternatives.

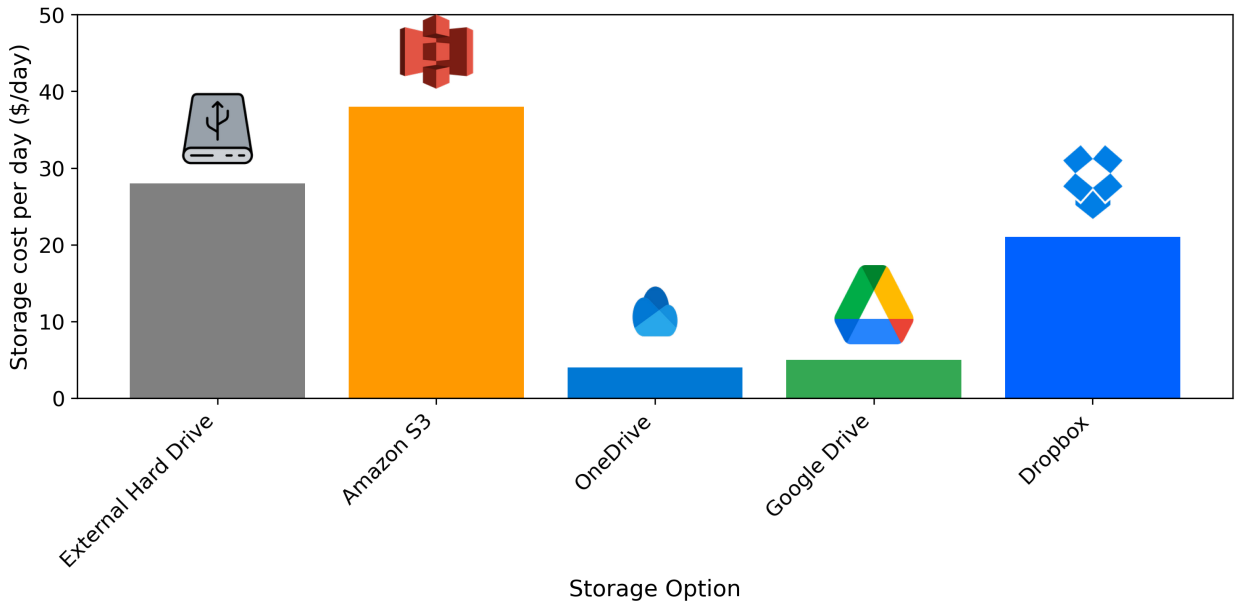

Figure 3.16: Comparison of daily storage costs for different data storage solutions used in 3D sensing systems for poultry, illustrating the cost-effectiveness of external hard drives versus various cloud storage options.

3.3.4 Broiler house testing performance

Performance of the three-dimensional monitoring device in a broiler house testing: With the system configuration and storage defined, this section evaluates its performance in real-world chicken farm conditions. In order to ensure efficient data management and system health monitoring, and to help the system administrator of the 3D sensing system have full control over the data recording process in all systems, a daily report process was designed and developed. In this system, each email, automatically generated and dispatched shortly after midnight, contained a detailed update on the specific monitoring system from which it was sent. The subject line includes the system's name, enhancing clarity and organization in the administrator's inbox. The body of such an email provided succinct details such as the IP address of the system, ensuring traceability and easy access to the system's network location, if needed. Furthermore, the email highlighted the remaining storage capacity on the external hard drive, expressed in gigabytes, which is crucial for preempting storage capacity issues and managing data overflow effectively. In a specific email, there were three charts indicating storage performance. The first chart details the volume of data recorded daily, broken down by hours, offering insights into the data acquisition process. Figure 3.17.a shows 70 GB of recorded 3D data hourly with minimal variations. The second chart illustrates the number of bag files recorded each hour, offering a granular view of data collection frequencies and timings. Figure 3.17.b shows 12 files recorded hourly with each being 5 minutes. The third chart tracks the remaining space on the external hard drive over three weeks, with each bar representing a day, thereby

Table 3.9: A comparison of local storage devices and cloud storage services in various criteria as viable options for saving poultry monitoring data. (Note: 'O' means the criterion is supported, and '×' means it is not supported.)

Criteria	**Local Storage Device**	**Cloud Storage Service**
Average Cost	×	O
Data Security	O	×
Latency	O	×
Setup Complexity	×	O
Monitoring Complexity	×	O
Scalability	O	×
Accessibility	×	O
Backup & Recovery	O	O
Customization	O	×

facilitating a long-term view of data storage trends and helping anticipate the need for data offloading or additional storage provisions. Figure 3.17.c indicates a trend of remaining space in one month, with three-day recordings every week that reduced remaining space linearly.

To examine the efficiency of the system even further, we recorded the data transfer and processing duration in different steps. The camera driver took about 4 seconds to start and can stay on for as long as needed. The data recording can start instantly in a few milliseconds with 30 frames per second for the RGB frames and 15 frames per second for the depth frames. Such frame rates can be increased with better performance of cameras. Once the data recording was finished after 5 minutes, the next recording started automatically. Each recorded bag file was sent to the external hard drive in parallel with the new data being recorded, and each file took about 61 seconds to be transferred. Storage was checked after each data transfer, and in case of low storage, the appropriate warning report was sent to the user in less than a second. Daily reports were generated in about 2 seconds and take less than a second to be sent every midnight to reduce data transferring burdens during the day.

The operational duration of the system without manual data management depends on the capacity of the external hard drive used. With a data generation rate of 1.65 TB per day and the current external hard drive capacity of 20 TB, the system can operate continuously for approximately 12 days before requiring manual intervention to replace the external hard drive. The duration of continuous operation can be extended with a larger storage space (e.g., >20 TB).

Three-Dimensional Movement Trajectories: Finally, this section demonstrates the system's ability to extract 3D movement trajectories, a key outcome of the evaluated setup. To extract 3D movement

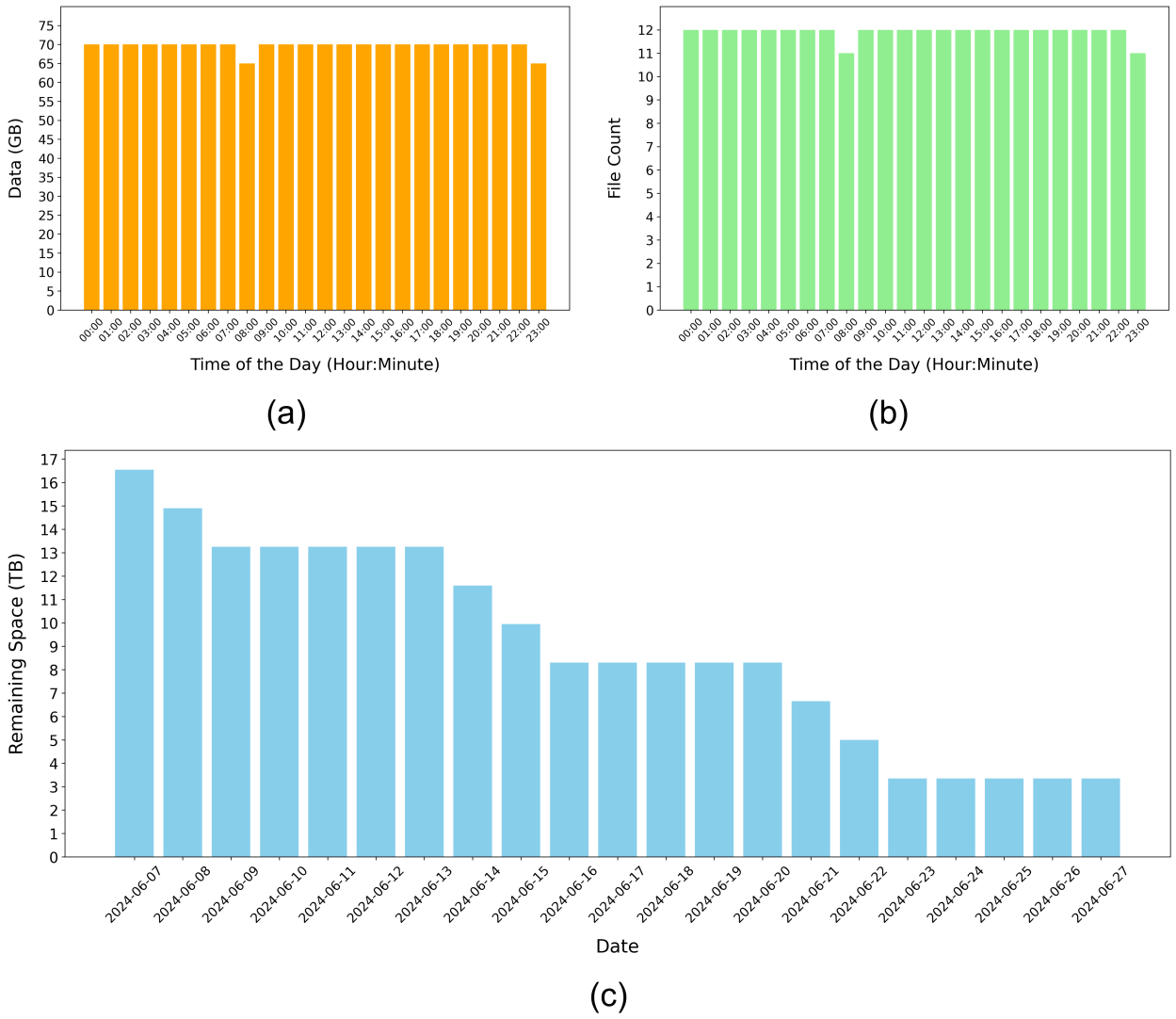

Figure 3.17: Data recording metrics from one of the 12 monitoring systems over 24 hours: (a) the amount of data recorded per hour in gigabytes, illustrating the consistent data capture throughout the day; (b) the number of bag files created per hour, indicating the frequency of data storage events; and (c) the remaining space on an external hard drive over three weeks, with each bar representing the storage space left in terabytes (TB) at the end of each day.

trajectories of chickens, compressed RGB and uncompressed depth frames were recorded in normal lighting conditions using an Intel RealSense L515 camera mounted at a height of 2.25 m above the ground. Once the data was transferred to the external hard drive, it was ready to be extracted and processed. The extraction time may vary based on the file size, but for a 5-minute bag file, it took about 244 seconds to extract the Red, Blue, and Depth channels (to make RDB frames) from 9,000 RGB and 4,500 depth frames. Once the data was extracted, the object detection model took about 42 milliseconds to run

on each RDB frame, making it take about 378 seconds for each 5-minute video to be processed by the object detection model. The RDB modality was chosen because it demonstrated higher object detection performance under normal lighting conditions (Table 3.4) and performs comparably well under other extreme conditions. The point cloud creation took an average of 68 milliseconds per frame, which was 612 s for a 5-minute video. Eventually, the movement trajectory was generated in about $1,874$ milliseconds. Therefore, the data extraction and processing were about $1,236$ seconds per video for an average of 34 chickens per frame.

The YOLOv8m object detection model, enhanced with ByteTrack for robust tracking, was employed to identify and track each chicken, assigning unique identifiers and tracking their movement across successive frames. The centroids of bounding boxes associated with each chicken ID were extracted in every RDB frame, allowing for the precise calculation of 3D movement distances and the visualization of movement trajectories. Figure 3.18 shows an example of a 5-minute-long 3D movement. The back height of a standing broiler at 28 days old is nearly 30–40 cm, and it is below 30 centimeters when a bird sits. Based on these, we observed that the five broiler chickens spent most of their time sitting within the 5 minutes and presented sparse trajectory patterns in the YZ-plane.

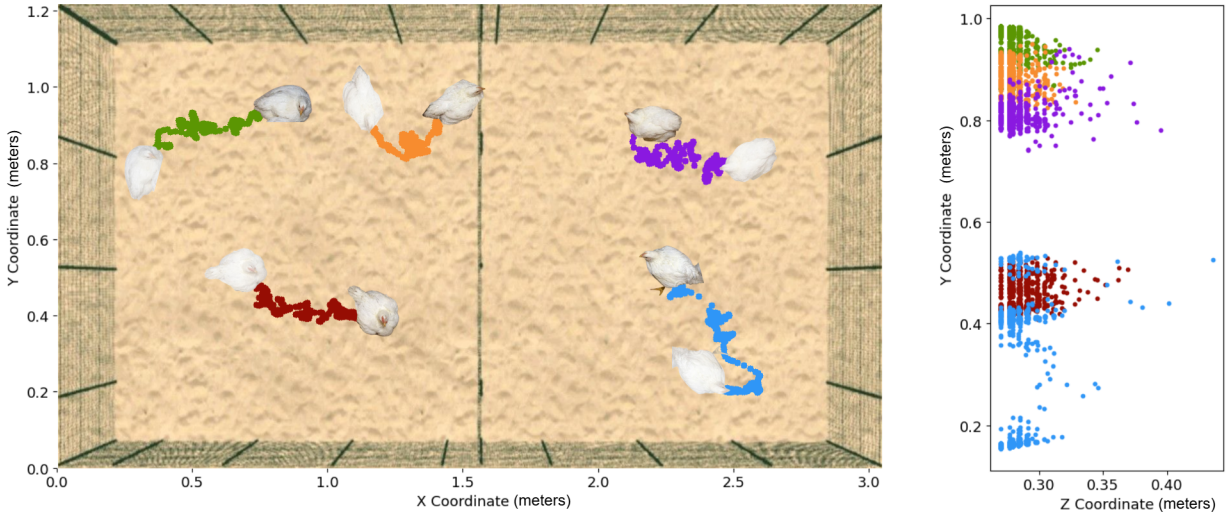

Figure 3.18: Visualization of three-dimensional movement trajectories of five chickens in XY (left) and ZY (right) planes. The trajectories were extracted from a 5-minute RGB-D video. Along the same trajectory highlighted with the same color, a brighter chicken demonstrates the initial status, and a darker bird shows the ending status.

3.4 Discussion

This section synthesizes the advancements in poultry monitoring and activity classification, situating our work within the broader landscape of existing research. By examining various approaches to behavior, health, and environmental monitoring, we highlight the contributions of our low-cost, 3D intelligent

system and its advantages over prior methods. The following subsections explore key themes, including monitoring techniques, sensor technologies, data modalities, and system comparisons, while addressing the challenges and future directions for precision poultry farming.

3.4.1 Overview of Poultry Monitoring Approaches

Behavior monitoring and activity classification in poultry farming have been addressed through various approaches. Some studies focused on monitoring and classifying activity levels using trajectory-based clustering (Campbell et al., 2024) and analyzing chicken behavior for surveillance using machine vision systems (Mohialdin et al., 2023). Others have targeted the tracking and characterization of locomotive behaviors to evaluate gait scores using deep learning models (G. Li et al., 2023) or assessing inactivity levels to detect lameness with 3D vision systems (Aydin, 2017). Health and welfare monitoring is another critical area, with work on early detection and disease monitoring through machine vision (Okinda et al., 2019), developing early warning systems for detecting abnormalities in broiler distribution and equipment malfunctions (Kashiha et al., 2013), and classifying poultry droppings for early health prediction (Nakrosis et al., 2023). Additionally, kinematic analysis has been employed to quantify gait abnormalities associated with lameness (Caplen et al., 2012). Environmental monitoring and control have been implemented through systems that monitor and regulate conditions such as temperature and humidity via wireless sensor networks (Murad et al., 2009) and IoT-based solutions (Orakwue et al., 2022). Other efforts focused on weight and physical assessment, including detecting live broiler weight using computer vision and support vector regression (Amraei et al., 2017), employing 3D computer vision for automatic weight prediction (Mortensen et al., 2016), and estimating carcass and cut weights using online vision systems (Nyalala et al., 2021). Specific tasks such as assessing plumage condition in laying hens (Lamping et al., 2022) and detecting stunning effectiveness (ChangWen et al., 2018) have also been explored.

3.4.2 Contribution of the Proposed System

Our work fits within the broader category of behavior monitoring and activity classification while also intersecting with health and welfare monitoring. Specifically, we developed and evaluated a low-cost intelligent system to monitor the three-dimensional features of poultry, enabling the acquisition of detailed behavioral and phenotypic traits. This approach provided significant advantages over previous methods by offering a more comprehensive analysis through 3D feature extraction. Unlike 2D or single-modality methods, our system captured the full spatial context of poultry movements and physical characteristics, facilitating the detection of subtle behavioral changes and phenotypic variations critical for early intervention and enhanced farm management.

3.4.3 Sensor Technologies in Poultry Monitoring

Various sensor types were employed in studies to monitor poultry behavior, health, and environmental conditions. RGB cameras were commonly used for activity tracking and floor distribution (Campbell

et al., 2024; G. Li et al., 2023), while depth cameras like the Kinect provided enhanced accuracy in measurements and 3D reconstructions (Mohialdin et al., 2023; Mortensen et al., 2016). Environmental monitoring often utilized specialized sensors such as DHT11 for temperature and humidity, MQ135 for gas detection, and infrared sensors for motion detection (Murad et al., 2009; Nyalala et al., 2021; Orakwue et al., 2022). These sensors were frequently integrated into IoT frameworks or microcontrollers like Arduino or Raspberry Pi for real-time processing. In our work, we used an RGB-D camera to capture three-dimensional poultry features, enabling precise analysis of behavior and phenotypic traits. This multimodal approach improved monitoring quality, opening numerous opportunities for precision poultry management.

3.4.4 Sensor Positioning Strategies

Sensor positioning played a crucial role in monitoring system effectiveness. Many studies used top-down views to monitor large areas, track movement, and reduce occlusion (Campbell et al., 2024; G. Li et al., 2023). Side-view placements were used for specific tasks such as gait analysis or capturing depth data for 3D modeling (Aydin, 2017; Mortensen et al., 2016). In our setup, the RGB-D camera was positioned in a top-down view at a height of 2.25 meters, covering a full pen and optimizing the field of view. This strategic placement captured comprehensive data while reducing obstructions, enabling accurate behavior analysis and phenotypic trait monitoring.

3.4.5 Data Modalities and Processing

Studies employed various data modalities to achieve their monitoring objectives. RGB images were widely used for behavior analysis, activity classification, and health monitoring (Campbell et al., 2024; G. Li et al., 2023; Mohialdin et al., 2023). Depth data from devices like Kinect and Intel RealSense cameras added spatial understanding for 3D modeling, gait analysis, and weight prediction (Aydin, 2017; Caplen et al., 2012; Mohialdin et al., 2023; Mortensen et al., 2016). Environmental parameters such as temperature, humidity, and gas levels were integrated into IoT-based frameworks for real-time monitoring (Murad et al., 2009; Nyalala et al., 2021; Orakwue et al., 2022). Specialized data modalities, such as infrared images for stunning detection (ChangWen et al., 2018) or droppings classification for health detection (Nakrosis et al., 2023), further diversified monitoring approaches. In our work, we collected RGB and depth data at 15 FPS and explored four modalities (RGB, RGD, RDB, DGB). This analysis enabled the identification of optimal modalities for accurate object detection and tracking, ensuring robust monitoring under diverse environmental conditions, especially under darkness.

3.4.6 Data Storage and Algorithmic Approaches

Data storage solutions in poultry monitoring varied widely. Local storage options, such as SD cards in microcontroller systems, were used for environmental data (Murad et al., 2009; Orakwue et al., 2022), while hard drives in PCs stored RGB and depth data (Mortensen et al., 2016; Nyalala et al., 2021). Cloud storage facilitated remote access and scalability but raised concerns about security and cost (Amraei et al.,

2017; Nyalala et al., 2021). In our work, we stored terabytes of RGB and depth data on external hard drives, balancing high capacity with immediate access. Regarding algorithms, studies employed traditional models like support vector regression (Amraei et al., 2017) and Bayesian artificial neural networks (Mortensen et al., 2016) for weight detection. Deep learning methods, such as ResNet (Nakrosis et al., 2023), U-Net (Lamping et al., 2022), and YOLO (Okinda et al., 2019), were favored for behavior classification and real-time applications. We trained YOLOv8m integrated with ByteTrack for object detection and tracking, providing a robust solution combining speed and precision.

3.4.7 Comparison with Commercial Systems

While commercial poultry monitoring systems (e.g., ChickenBoy Robot, Octopus SCARIFIER, and T-MOOV) offer advanced capabilities for monitoring and management in broiler production, they have several limitations when compared to the proposed 3D monitoring system. Table 3.10 outlines a detailed comparison of these systems based on key qualitative evaluation factors, including cost, portability, scalability, and functionality. This comparison highlights that compared to these systems, the proposed system excels in its cost-effectiveness, portability, and scalability, making it accessible to both small-scale and large-scale poultry operations. The advanced RGB-D imaging allows for precise 3D monitoring of behavior and phenotypic traits, a capability that other systems lack. Furthermore, its offline functionality and customizable design ensure it can adapt to various farm conditions without requiring extensive modifications or continuous internet access. However, unlike some existing systems that monitor environmental parameters such as humidity or CO2 levels, our system focuses primarily on vision-based monitoring. However, these additional sensors can be easily integrated into the system if needed, offering flexibility for diverse monitoring needs.

Table 3.10: Comparative analysis of the proposed 3D monitoring system against existing broiler monitoring systems across key qualitative evaluation factors.

Evaluation Factors	**ChickenBoy Robot**	**Octopus SCARIFIER**	**T-MOOV**	**Proposed System**
Cost	High	High	Moderate	Low
Portability	Low	Low	Moderate	High
Data Modality	2D + environment data	2D + litter aeration	Movement-focused	Advanced RGB-D data
Scalability	Low	Moderate	Moderate	High
Internet Dependence	Cloud-dependent	Requires connectivity	Requires connectivity	Offline enabled
Target Parameters	Environment monitoring	Litter aeration	Bird movement	Behavior & phenotypic traits
Customizability	Limited	Limited	Moderate	High
Installation Process	Complex	Complex	Moderate	Simple

3.4.8 Applications and Limitations

The methods proposed in poultry monitoring studies offered broad applications. IoT-based systems optimized environmental control (Murad et al., 2009; Orakwue et al., 2022), while object detection

models supported sorting and grading (G. Li et al., 2023; Okinda et al., 2019). Health monitoring systems using image processing and deep learning enhanced disease monitoring and behavior analysis (Lamping et al., 2022; Nakrosis et al., 2023). Similarly, our system enabled automated grading and real-time monitoring through high-resolution RGB and depth data. The integration of YOLOv8m with ByteTrack supported selective breeding and farm management optimization. While our system addressed many limitations of existing methods, challenges such as ongoing storage costs and initial setup complexity persisted. Despite these challenges, its adaptability and scalability ensured that it remained a robust, future-proof solution for poultry monitoring.

3.4.9 Sensor Technology Considerations

The proposed 3D monitoring system employed advanced sensor technology, notably the Intel RealSense L515 camera, to achieve precise monitoring of poultry behavior and phenotypic traits. While the L515 was chosen for its high-quality depth frame accuracy and close-range precision, its field of view (FoV) imposes some limitations on the area that can be monitored with a single unit. This may require deploying multiple systems to cover larger poultry houses effectively, which could increase setup complexity and cost. The camera's depth range, although well-suited for typical poultry farm layouts, affects the height at which it can be installed above chicken pens, which may need adjustment based on specific housing configurations. Additionally, while the system focuses on vision-based monitoring, it does not currently incorporate environmental sensors for parameters such as temperature, humidity, or CO2 levels, which are important for comprehensive farm management. However, the system's modular design allows for straightforward integration of these sensors, offering flexibility for future enhancements. The L515's IP50 dust resistance ensures reliable operation in standard farm conditions, but protective enclosures may be necessary in environments with excessive humidity or water exposure.

3.4.10 Challenges and Future Directions

Beyond sensor-related considerations, other factors should also be noted. The machine learning algorithms employed for behavioral analysis perform robustly under typical farm conditions but may face challenges in scenarios involving dense clustering of birds or highly variable lighting. Additionally, while the system has been validated under continuous operation for 7 weeks without significant performance degradation, further testing across diverse global environments, such as tropical climates or high-altitude farms, remains an area for exploration. These considerations represent opportunities for iterative refinement to ensure broader applicability and even greater reliability of the system in precision poultry farming.

3.4.11 Evaluation scope and methodology

Testing and reporting results for all combinations of factors—comprising $1,280$ possibilities derived from variations in camera models, heights, RGB frame qualities, data modalities, lighting conditions, and object

detection models—were beyond the scope of this study. Instead, we analyzed each factor's impact individually to ensure a rigorous yet practical evaluation. While normal lighting conditions were prioritized due to their prevalence and critical influence on system performance, extreme lighting conditions were addressed selectively, focusing on the robustness of YOLOv8m with different data modalities, as detailed in Section 3.3.2 and Tables 3.4–3.7.

3.5 Conclusion

This study integrated and evaluated a low-cost intelligent system for monitoring the three-dimensional features of broiler chickens. The system integrated an RGB-D camera, ROS, and a portable hardware setup, effectively capturing and processing high-quality 3D data. The optimal configuration—a camera height of 2.25 m with compressed RGB and uncompressed depth data—enabled detailed point cloud reconstruction and efficient storage. The system demonstrated continuous operation for 12 days using a 20 TB external hard drive, minimizing manual intervention. In addition to testing various camera models, installation heights, and data compression methods, the study evaluated the system's robustness under diverse lighting conditions, highlighting the superiority of depth-enhanced modalities for bird detection and tracking in low-light and challenging scenarios. Among the tested object detection models, YOLOv8m outperformed others in terms of precision and accuracy, particularly under normal lighting conditions. The integration of these optimized components ensured high detection accuracy and the reliable extraction of 3D walking trajectories, enabling detailed behavioral analysis. Integrating depth modality enhances bird detection accuracy, especially in the dark. This cost-effective solution advances precision livestock farming by enabling real-time poultry monitoring, improving animal welfare, and optimizing farm management.

Chapter 4

A Novel Three-dimensional Deep Learning Approach for Auditing Gait Scores of Individual Broiler Chickens

Manually auditing gait scores of broiler chickens is labor-intensive and subjective, necessitating the development of an automated and objective alternative. This study presents a novel three-dimensional (3D) deep learning pipeline to assess the walking ability of broilers by predicting gait scores ranging from 0 (optimal mobility) to 2 (severely impaired). A total of 540 broiler chicken videos, sampled from 6 to 7 weeks of age, were recorded as the chickens traversed a 1.75-meter wooden platform. An Intel RealSense L515 LiDAR camera, mounted at a 2.5-meter height, was used to capture synchronized RGB-Depth data. The data were recorded using the Robot Operating System (ROS) Noetic, ensuring efficient and structured data acquisition. The proposed pipeline consisted of multiple sequential steps: (1) RGB-Depth frame extraction and synchronization, (2) pose estimation using a custom-trained YOLOv11-based chicken pose detection model, (3) back-projection of 2D keypoints into 3D space using camera intrinsics, (4) frame validation using a convolutional neural network to filter out occlusions and artifacts, (5) platform orientation detection via Hough line transformation, (6) segmentation of the chicken's body using the Segment Anything Model (SAM) to extract 3D point clouds, and (7) kinematic feature extraction for gait analysis. Key 3D features, including velocity, acceleration, and head turn frequency, were fed into a multi-layer perceptron classifier to predict gait scores. The classifier predicted broiler gait scores with 93.34% accuracy, 95.56% precision, 91.16% recall, and 93.31% F1-score. The system demonstrated robustness in various environmental conditions, offering a scalable and cost-effective solution for poultry welfare monitoring. The entire system was developed at an approximate cost of $\$1,483 \pm \$127$, making it an affordable alternative for large-scale poultry research and field sampling operations. This research highlights the potential of integrating 3D vision with deep learning to enhance automated gait scoring,

ultimately improving assessment reliability, reducing labor costs, and contributing to the advancement of precision poultry farming.

4.1 Introduction

Broiler chicken production is one of the largest and fastest-growing sectors in the global livestock industry, with an annual output exceeding 100 million metric tons of meat worldwide (D. Pereira, Nääs, & da Silva Lima, 2021). The United States, Brazil, and China are the leading producers, accounting for a significant share of global broiler production, with the U.S. alone producing approximately 20.2 million metric tons per year (D. Pereira, Nääs, & da Silva Lima, 2021). The rapid growth and high stocking densities of commercial broiler chickens have led to increasing concerns regarding animal welfare, particularly the prevalence of disproportional body weight distribution, unsupportive bone density and structure, and locomotion issues, leading to impaired walking ability and further lameness (Van Hertem et al., 2018). Gait scoring is a method to evaluate bird walking ability and welfare status. Given that over 50% of broilers in commercial settings may exhibit some degrees of gait impairment (van der Sluis et al., 2021a), it is necessary to conduct timely evaluation of gait scores and provide precise management strategies to prevent the problems from being worse.

Gait scoring is to assign specific ordinal scores to chickens based on their walking ability, such as the widely adopted six-point gait scoring system and three-point scoring system in the United States (Certified, 2019; Webster et al., 2008). While this method provides a baseline for assessing mobility impairments, the manual methods of gait scoring have several drawbacks (Riber et al., 2021; Wurtz & Riber, 2024). Variability in scoring can arise due to differences in observer expertise, environmental conditions, and bird behavior, leading to inaccurate or inconsistent assessments (Wurtz & Riber, 2024). Furthermore, manual gait scoring is time-consuming and impractical for large-scale commercial poultry farms, where thousands of birds need evaluation within limited timeframes (Van Hertem et al., 2018). These limitations hinder effective monitoring and delay interventions for lameness (Nääs et al., 2018). Such labor-intensive tasks become more challenging with the predicted labor shortage of the farm jobs in the near future (Zahniser et al., 2018). Given these challenges, there is a critical need for automated, scalable, and objective gait assessment solutions to enhance poultry health monitoring and management.

The integration of smart monitoring technologies in poultry farming has revolutionized livestock management by enabling real-time, non-invasive, and objective assessments of bird health and behavior. Modern systems leverage computer vision, sensor networks, and machine learning to monitor locomotion patterns, detect anomalies, and provide early warnings of welfare issues (Van Hertem et al., 2018). Automated monitoring systems offer consistent, scalable, and cost-effective solutions for large-scale poultry production and overcome the drawbacks of manual methods as mentioned above (Silvera et al., 2017). These technologies can track key gait parameters such as walking speed, stride length, step frequency, and lateral body oscillation, allowing precise classification of mobility impairments (Fodor et al., 2024). Additionally, deep learning models trained on large datasets can outperform human assessors in detecting subtle locomotor abnormalities, contributing to improved flock health management (G. Li et al., 2023).

By integrating automated gait assessment with real-time monitoring, poultry producers can enhance early disease detection, reduce labor costs, and improve overall animal welfare, aligning with the growing demand for precision livestock farming (Nääs et al., 2018).

The integration of three-dimensional (3D) cameras in poultry gait analysis has improved mobility assessment by enabling depth-aware movement tracking, eliminating limitations associated with 2D vision-based systems (Okinda et al., 2019). Unlike traditional RGB cameras, which are prone to occlusions, perspective distortions, and shadow artifacts, 3D depth sensors such as Intel RealSense LiDAR, Microsoft Azure Kinect, and stereoscopic vision systems allow for precise reconstruction of broiler motion in 3D space, ensuring accurate gait scoring across diverse environmental conditions (Goyal et al., 2024). A notable study by Li et al. (G. Li et al., 2023) utilized 3D motion tracking to analyze the locomotive behaviors of individual broilers in a three-point gait-scoring system. Their approach involved keypoint-based pose estimation and depth-aware spatiotemporal feature extraction, demonstrating that 3D skeletal tracking provides significantly better differentiation of gait scores than conventional 2D methods. The study highlighted that step height, limb angular displacement, and velocity fluctuations were more reliably quantified using 3D pose data, reinforcing the superiority of depth-based gait monitoring for precise mobility impairment detection.

Other recent works have leveraged LiDAR-based depth sensors to enhance 3D skeleton reconstruction, allowing for fine-grained movement analysis and minimizing errors due to occlusions and overlapping body parts. Additionally, stereo vision-based gait tracking has been employed to further improve classification accuracy and robustness, particularly in high-density flock environments (Faysal et al., 2021). Studies also indicate that 3D point cloud-based motion tracking provides enhanced feature extraction, enabling early detection of subtle locomotor abnormalities that might be overlooked in 2D image-based assessments (Abd Aziz et al., 2021). Furthermore, integrating multi-modal fusion techniques, combining 3D depth data with hyperspectral and thermal imaging, has shown improved sensitivity in detecting early-stage mobility impairments, offering a more comprehensive view of gait abnormalities (George & George, 2023). Deep learning models trained on 3D skeletal data have also outperformed traditional 2D CNN-based classifiers, demonstrating the effectiveness of depth-aware neural networks for gait scoring (Supriyanto et al., 2023). Given these advancements, 3D vision-based gait assessment is emerging as a highly scalable, non-invasive, and objective solution for poultry welfare monitoring, paving the way for precision livestock farming with greater automation and reliability (Goyal et al., 2024).

Object detection models have been progressively deployed in poultry farming and enabled automated health monitoring, behavior analysis, and welfare assessment. Among these, the YOLO (You Only Look Once) family (Redmon et al., 2016) stands out due to its real-time processing capability, high accuracy, and robustness under challenging conditions (Elmessery et al., 2023; Khairunissa et al., 2021). Unlike traditional two-stage models such as Faster R-CNN (S. Ren et al., 2017), which, while precise, suffers from slow inference speeds, YOLO operates as a single-stage detector, making it ideal for high-throughput real-time poultry surveillance (Nakrosis et al., 2023). Other alternatives like SSD (Single Shot MultiBox Detector) (W. Liu et al., 2016) are lightweight and efficient but often struggle with detecting small objects and occlusions, which are common challenges in poultry farming environments (Ariza-Sentís et al., 2024).

YOLO has been successfully applied in broiler health monitoring, detecting abnormalities such as stress indicators, lethargy, and skeletal deformities from both RGB and thermal imagery (Čakić et al., 2022). It has also been integrated with multi-object tracking algorithms to analyze flock movement patterns, aiding in the early detection of disease outbreaks and behavioral anomalies (Khairunissa et al., 2021). In addition, YOLO-based models have demonstrated superior performance in classifying poultry droppings for non-invasive health assessments and automating egg detection, improving efficiency in poultry production chains (G. Li, Chesser, et al., 2021). In our study, we leverage YOLOv11 (Jocher et al., 2023), the recent state-of-the-art iteration, to enhance object detection and pose estimation for poultry gait analysis. YOLOv11 incorporates advanced transformer-based architectures, dynamic anchor assignment, and improved feature extraction, significantly boosting detection precision while maintaining real-time performance. This ensures accurate pose estimation of broilers, even under variable lighting and occlusion scenarios, making it the most suitable model for precision poultry monitoring.

Along with object detection and pose estimation, accurate object segmentation is critical in poultry gait analysis, particularly for isolating birds from their surroundings to extract precise 3D kinematic features. Various deep learning-based segmentation models have been explored for this task, including Mask R-CNN, U-Net, and DeepLabV3+. While Mask R-CNN (He et al., 2017) performs well for instance segmentation, its reliance on region proposal networks (RPNs) makes it computationally expensive and slower for large-scale datasets (Okinda et al., 2019). Similarly, U-Net (Ronneberger et al., 2015), though efficient in medical imaging applications, struggles with generalization in unconstrained environments, leading to poor segmentation in cluttered farm settings (Abd Aziz et al., 2021). DeepLabV3+, which utilizes dilated convolutions for multi-scale feature extraction (Chen et al., 2018), has demonstrated high segmentation accuracy but is prone to over-segmentation in overlapping objects, making it less reliable in high-density poultry farming environments (Goyal et al., 2024). To overcome these challenges, we adopted the Segment Anything Model (SAM), a state-of-the-art foundation model designed for general-purpose object segmentation (Kirillov et al., 2023). Unlike previous models, SAM is pre-trained on massive datasets, enabling it to generalize well across diverse farm conditions without extensive re-training. Its ability to perform zero- or few-shot segmentation allows for flexible adaptation to different poultry breeds, lighting conditions, and backgrounds, making it more robust than task-specific models (Goyal et al., 2024). Additionally, SAM's efficient prompt-based segmentation mechanism eliminates the need for manual annotation and fine-tuning, significantly reducing the computational burden while improving segmentation quality.

While significant progress has been made in automating poultry gait assessment, many existing approaches remain limited in scalability, robustness, and generalization to real-world farm conditions. One major limitation is the reliance on 2D motion analysis, which misses critical 3D spatial features such as step height, lateral oscillation, and angular displacement, reducing accuracy in gait score differentiation (D. Pereira, Nääs, & da Silva Lima, 2021). Many methods are also designed for controlled environments, requiring specific lighting, uniform backgrounds, and marker-based tracking, making them impractical for large-scale commercial farms (G. Li et al., 2023). Additionally, several studies suffer from limited feature extraction, relying primarily on velocity and acceleration, without incorporating a more compre-

hensive set of kinematic descriptors in 3D spaces (Nääs et al., 2018). This results in poor generalization, particularly when applied to flocks with diverse walking behaviors or variable environmental conditions (Abd Aziz et al., 2021). Even 3D-based tracking solutions, while promising, have been constrained by small datasets, often using fewer than 20 birds, increasing the risk of overfitting and reduced scalability (G. Li et al., 2023). Another key challenge is the computational efficiency of existing deep learning models. Two-stage detectors such as Faster R-CNN offer high accuracy but lack real-time processing capability, making them impractical for continuous monitoring of broiler locomotion (Goyal et al., 2024). Alternatives like SSD and traditional CNN-based models struggle with small-object detection and occlusion, which are common in high-density poultry farming environments (Okinda et al., 2019). To overcome these limitations, our study introduced YOLOv11-based 3D pose estimation, integrating multi-keypoint detection and depth-aware motion analysis to provide precise, real-time gait assessments in uncontrolled farm conditions. By eliminating lighting dependencies, expanding dataset scalability, and incorporating richer kinematic features, our approach offers a more robust, automated, and scalable solution for large-scale poultry gait assessment.

This study aimed to develop a scalable and automated 3D gait scoring system to address the limitations of previous approaches in poultry locomotion assessment. We introduced a multi-stage deep learning pipeline that integrated RGB-Depth data acquisition, multi-keypoint pose estimation using YOLOv11, depth-aware motion analysis, platform orientation detection, body segmentation with SAM, and 2D/3D kinematic feature extraction. A diverse set of 3D kinematic features was extracted and used to train a multi-layer perceptron classifier to predict gait scores on a scale from 0 (optimal mobility) to 2 (severely impaired mobility). Unlike prior 2D-based methods, our system utilized 3D spatial information to enhance gait assessment while also incorporating a frame validation model to identify and exclude occluded or noisy frames, improving the robustness of the dataset. Additionally, our approach reduced strict setup requirements, making it more adaptable for practical use in commercial farm environments. The proposed system was designed to enhance gait scoring accuracy, automation, and scalability, contributing to a cost-effective and objective solution for poultry welfare monitoring.

4.2 Materials and Methods

4.2.1 System and Experiment Setup

A portable intelligent system was designed to capture and store RGB and depth data for subsequent analysis. The core components were enclosed within a compact plastic box (13 cm × 21 cm × 29 cm), ensuring protection and easy transportation. The system included a mini-PC, extension cables, HDMI connections, power cords, and small cooling fans to maintain optimal operation (Table 4.1). At the heart of the system was a lightweight mini-PC (11 cm × 12 cm × 4 cm, 940 g), featuring a 2 GHz Intel Celeron processor, 8 GB LPDDR4 RAM, a 256 GB SSD, and an Intel UHD Graphics Coprocessor with 32 MB of dedicated graphics memory. The device operated on Ubuntu 20.04 LTS (Focal Fossa), providing a stable Linux environment for data acquisition and processing. The Intel RealSense L515

3D camera, utilizing LiDAR technology, was employed for precise depth sensing. This sensor supported an operational range of 0.25 to 9.00 m, offering depth resolutions up to 1024×768 pixels and RGB resolutions up to 1920×1080 pixels at a maximum frame rate of 30 fps. Weighing only 100 g, the camera provided a highly portable solution for capturing detailed spatiotemporal gait data. To facilitate flexible positioning, the 3D camera was mounted on a wooden tower with 250 cm height and 60 cm width, providing a top-down view of the recording area. The mini-PC remained housed within the plastic enclosure on the ground, with the camera connected via a high-speed USB 3 extension cable, ensuring stable data transmission. The recording area featured a rectangular wooden platform with dimensions of 50 cm (width) × 310 cm (length) × 40 cm (height), allowing for uninterrupted movement tracking of broilers. The Robot Operating System (ROS) Noetic, integrated with camera drivers, was utilized to manage data acquisition, enabling efficient recording, storage, and replay of RGB-D streams. To perform the experiments, multiple Python packages and libraries were used, including but not limited to ultralytics (v8.3.77), python-opencv (v4.10.0.84), ROS Noetic, numpy (v1.26.4), json5 (v0.9.25), and configargparse (v1.7). Figure 4.1 illustrates the complete setup, highlighting the 3D camera mounted on the wooden tower at a height of 245 cm for optimal field coverage.

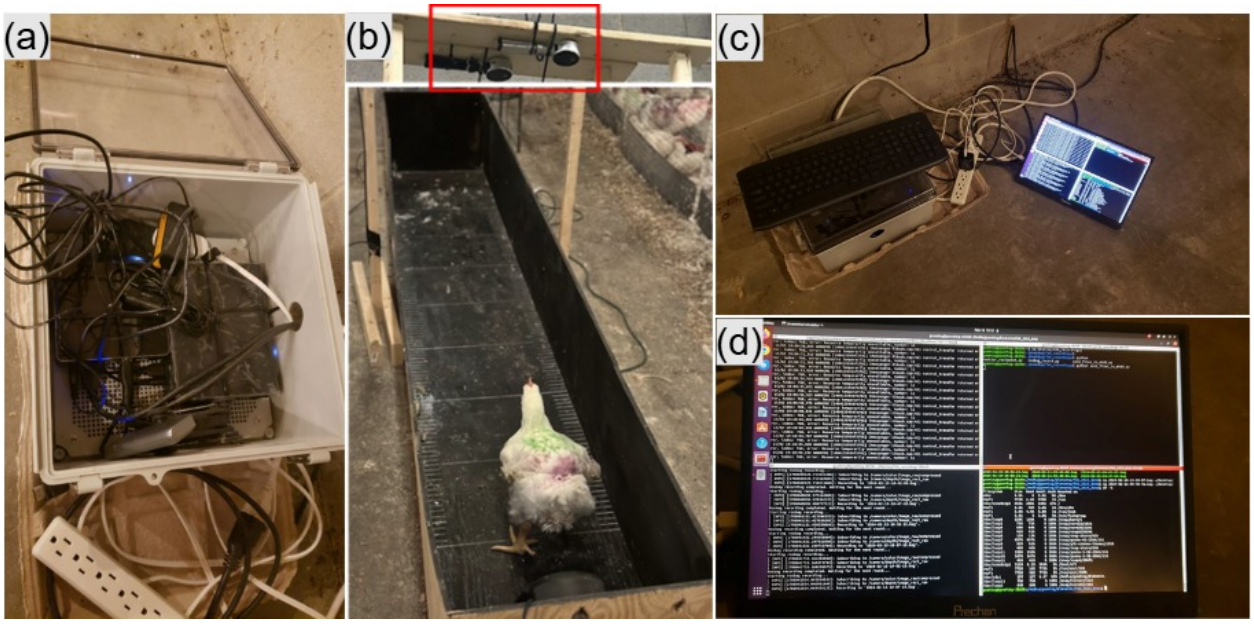

Figure 4.1: System integration for the 3D gait scoring setup in a broiler house. (a) Integrated unit; (b) 3D camera mounted on a wooden tower overlooking the walking platform with a broiler for gait assessment; (c) & (d) Touchscreen interface for real-time visualization and debugging

The total system cost for data collection and storage was approximately $1483 ± $127, as detailed in Table 4.1.

Once data collection was completed and the data was stored on external hard drives, a separate, high-performance desktop PC running Windows 11 Enterprise was used for model training and testing. The

Table 4.1: Detailed list of components used to set up the gait scoring system in a broiler house. The price estimates were collected in March 2025. (Note: * represents that the item is optional.)

Component	Model number	Function	Unit price ($)
3D camera	Intel RealSense L515	Sensing and image/video acquisition	680.00
Mini-PC	CyberGeek Nano J1	Data acquisition and edge computing	145.99
Junction box	QL-281913AGH	Housing components	56.69
Extension Cable	B08QRP6CGF	Extending connection between the box and camera	15.99
USB Hub	UGREEN CM481	Expanding number of connections with the PC	6.99
Wooden Platform	N/A	Provided area for the chicken to walk in and to be assessed for its movement ability	65.00
Wooden Tower	N/A	Adjusting installation height of the 3D camera	25.0
Power Extension Cord	TROND B0B497LDDW	Powering the device	6.11
HDMI Cable	Snowkids B099ZR82TM	Connecting the PC with the touchable screen	9.99
Power Strip	Woods 0600 W	Extending the power connection	9.50
External HDD	WDBWLG0200HBK-NESN	Offline data storage	334.89
Touchable Monitor*	ZSCMALLS ZSCM18	Visualizing running programs and interfaces	99.99
Wireless Keyboard w/ Touchpad*	Logitech K400 Plus	Inputting commands	27.30
		Total cost w/o optional items	1,356.15
		Total cost (all items)	1,483.44

system is equipped with a 13th Gen Intel® Core™ i7-13700 CPU and 64 GB of RAM, and it also features an NVIDIA RTX A4500 GPU with 20 GB of GDDR6 memory.

All of the experiments and system examinations were conducted in the Poultry Research Center at the University of Georgia. The center was located at the southern part of Athens in Georgia State. The series of experiments were conducted from February to June 2024. A total of 1,776 day-old Cobb 500 off-sex male broilers, raised and managed according to cobb 500 guideline (2022), were randomly allocated to 48 pens (with an average of 37 birds per pen), each containing either fresh or used litter and receiving one of four copper supplementation levels (5, 125, 250, or 500 ppm). To monitor welfare over time, ten out of 37 birds per pen were randomly chosen, color-marked, and evaluated from weeks 4 to 7. All experimental procedures were performed in compliance with protocols approved by the Institutional Animal Care and Use Committee (IACUC) at the University of Georgia (protocol number: A2023 07-016-Y1-A0).

A code was assigned to each color to quantize it. Green, purple, black, red, green-purple, green-black, green-red, purple-black, purple-red, and black-red were assigned numbers 1 to 10, respectively, for their color code. Individual body weights and litter moisture levels were measured. To measure the individual gait score values, all of the colored chickens from all pens were placed in the wooden platform (Figure 4.1.(b)) one by one, and three humans were observing the chicken movements. After a major voting between them, a score was assigned to the chicken's movement behavior and recorded on a paper sheet. Score 0 indicated the birds can walk 1.5 m without any lameness signs; Score 1 indicated the birds can

walk 1.5 m with slight lameness signs; and Score 2 indicated the birds cannot walk 1.5 m (Kestin et al., 1992). The video recording was also in progress while the chicken was moving along the platform, and an operator was starting and ending the recording process for each chicken. For each chicken, multiple data attributes were recorded, such as assessment date, pen number, color code, copper supplement level, litter condition, body weight, and gait score.

4.2.2 Pipeline Overview

The developed 3D gait scoring system followed a structured multi-stage pipeline that integrated RGB-D data acquisition, preprocessing, 3D pose estimation, segmentation, and gait classification. The process began with data acquisition, where RGB and depth frames and the camera intrinsics information were recorded using the portable unit as mentioned in the previous section. The recorded data was saved in ROS bag files, ensuring synchronized multi-modal storage for further processing. A total of $1,920$ bag files were recorded for 480 chickens during weeks of age 4–7 (once per chicken every week). The average size of each bag file was approximately 2.1 GB. Since the RGB and depth frames were captured at different frame rates, a preprocessing step was applied to extract the stored data and synchronize the frames. To maintain temporal alignment, unsynchronized RGB frames were discarded, and only those that matched the corresponding depth frames were kept.

Once the data was synchronized, 3D pose estimation and reconstruction were performed using YOLOv11, which detected multiple keypoints on the chicken's body from the RGB frames. These 2D keypoints were then back-projected into 3D space using depth data and intrinsic camera parameters, allowing for precise motion tracking. However, not all recorded frames were suitable for analysis due to potential issues such as motion blur, occlusions, or incomplete detections. To address this, a frame validation module was applied to filter out noisy frames using a CNN-based model, ensuring that only high-quality frames were used for subsequent processing.

To maintain a consistent walking perspective across all samples, the platform reorientation module was introduced to correct any misalignment in the walking platform with respect to the camera's field of view. This module first applied Canny edge detection (Canny, 1986), followed by a Hough Line Transform (Duda & Hart, 1972), to identify the platform's upper and lower edges. The dominant slope of these edges was computed across frames, and a geometric transformation was then applied to rotate the frames accordingly, ensuring that the platform was aligned with the camera's perspective. This correction was also applied to the detected 2D keypoints and bounding boxes, preserving their spatial alignment with the corrected platform orientation.

After platform reorientation, chicken body segmentation was performed to accurately extract the broiler from the background. The SAM was used to generate precise segmentation masks, effectively isolating the bird from irrelevant background regions, which improved the robustness of subsequent feature extraction. With the cleaned and aligned data, kinematic features such as velocity and acceleration were computed based on the 3D pose data. These extracted features were then used to classify gait scores using a multi-layer perceptron classifier.

A high-level overview of this pipeline is illustrated in Figure 4.2, which summarizes the data flow from data acquisition to final gait score prediction. The following sections of this paper provide a detailed explanation of each module in the pipeline.

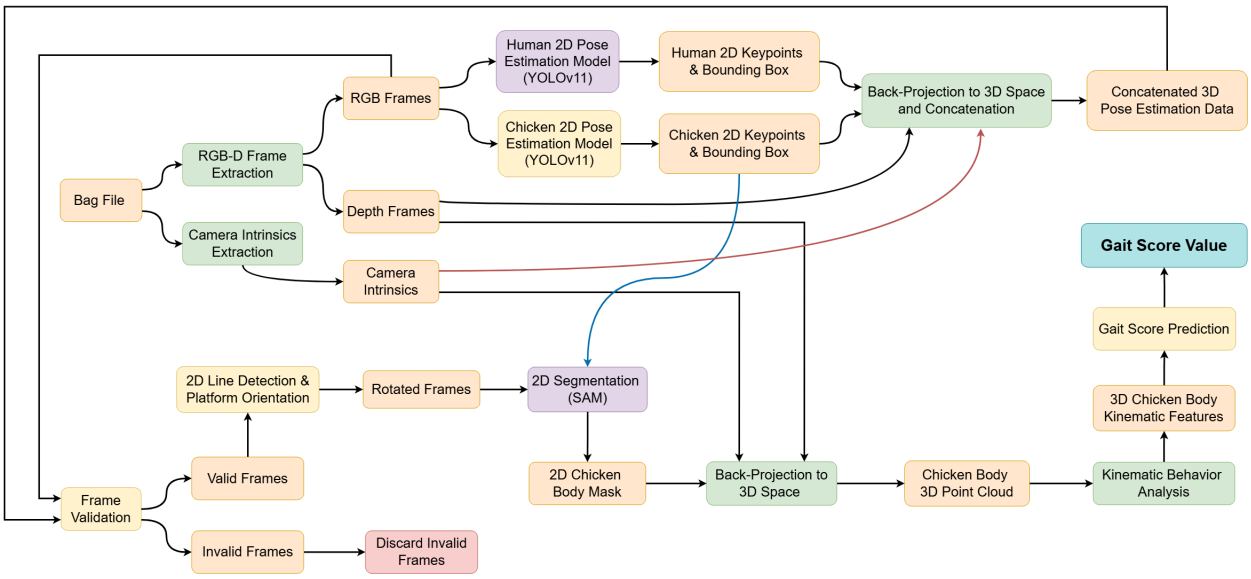

Figure 4.2: Overview of the gait scoring pipeline, from data collection to final classification. Different box colors represent different module types in the pipeline (orange box: data block, green box: mathematical operation, purple box: pre-trained model, yellow box: custom-trained model, red box: discarded data, and pale-blue box: gait score prediction).

4.2.3 Data Preprocessing

The recorded RGB and depth data required preprocessing to ensure proper synchronization and compatibility for further analysis. Since the Intel RealSense L515 camera captured RGB frames at 30 frames per second (FPS) and depth frames at 15 FPS, a temporal alignment process was applied to match the two modalities. Each depth frame was paired with the nearest RGB frame in time, ensuring that both image types were synchronized at a uniform 15 FPS for subsequent processing. If two RGB frames were equally distant from a depth frame (forward and backward), the code chose the earlier RGB frame (the one before the depth frame) to avoid relying on future information. After synchronization, the RGB frames were converted and stored in JPG format, optimizing them for lightweight storage and easy accessibility during later stages of processing. The depth frames, containing essential spatial information, were saved as NumPy arrays (.npy format) to preserve precision while facilitating efficient numerical computations. This structured storage approach allowed seamless access to both image modalities, ensuring compatibility with deep learning models and reducing unnecessary computational overhead. All subsequent analyses were performed on temporally aligned and standardized data, minimizing discrepancies caused by differences in frame rates. A visual example of synchronized RGB and depth frames is illustrated in Figure 4.3, demonstrating the paired frames captured during a broiler's movement on the walking platform.

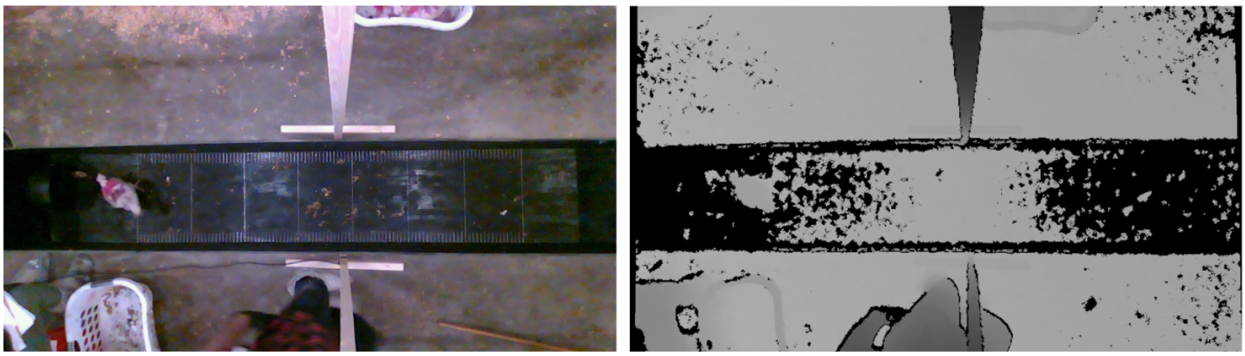

Figure 4.3: An example of synchronized RGB and depth frames captured during a broiler's movement on the walking platform. The RGB frame (a) provides visual context, while the corresponding depth frame (b) preserves spatial and depth information for further processing in pose estimation and gait analysis.

4.2.4 Two-Dimensional Chicken and Human Pose Estimation

Accurately detecting key anatomical landmarks in broilers was essential for analyzing locomotion patterns and extracting meaningful gait features. A multi-keypoint pose estimation model based on YOLOv11 (yolo11m-pose) was used to detect and localize keypoints on chickens from RGB frames. The chicken pose estimation model was trained to identify five distinct chicken keypoints, including the head, tail, left-wing, right-wing, and center of the body, providing a structured representation of the bird's movement. The Roboflow online annotation tool was used to label a total of 700 RGB frames. The data was randomly shuffled, and 500 frames were selected for the training set, while 100 frames were allocated to each of the validation and test sets. The model was trained for 100 epochs with input images resized so that their shortest sides were 640 pixels. The initial learning rate was set to 0.01, with a momentum of 0.937 and a weight decay of 0.0005. A batch size of 32 was used, and Stochastic Gradient Descent (SGD) (Robbins & Monro, 1951) served as the optimizer. Once the training finished, the model with the lowest training loss value was saved in PyTorch model format (.pt file).

Additionally, a pre-trained YOLOv11 pose estimation model (yolo11m-pose) was utilized for human keypoint detection, identifying 17 anatomical landmarks, including the nose, eyes, ears, shoulders, elbows, wrists, hips, knees, and ankles. The human played as an interference factor in predicting the gait scores, and the human keypoint information was later used to help with the frame validation process.

The YOLOv11-based pose estimation framework operated as a single-stage detector, allowing it to simultaneously detect bounding boxes and keypoints for both chickens and humans. An overview of the YOLOv11 architecture used for multi-keypoint pose estimation is shown in Figure 4.4, illustrating the network's components and detection pipeline.

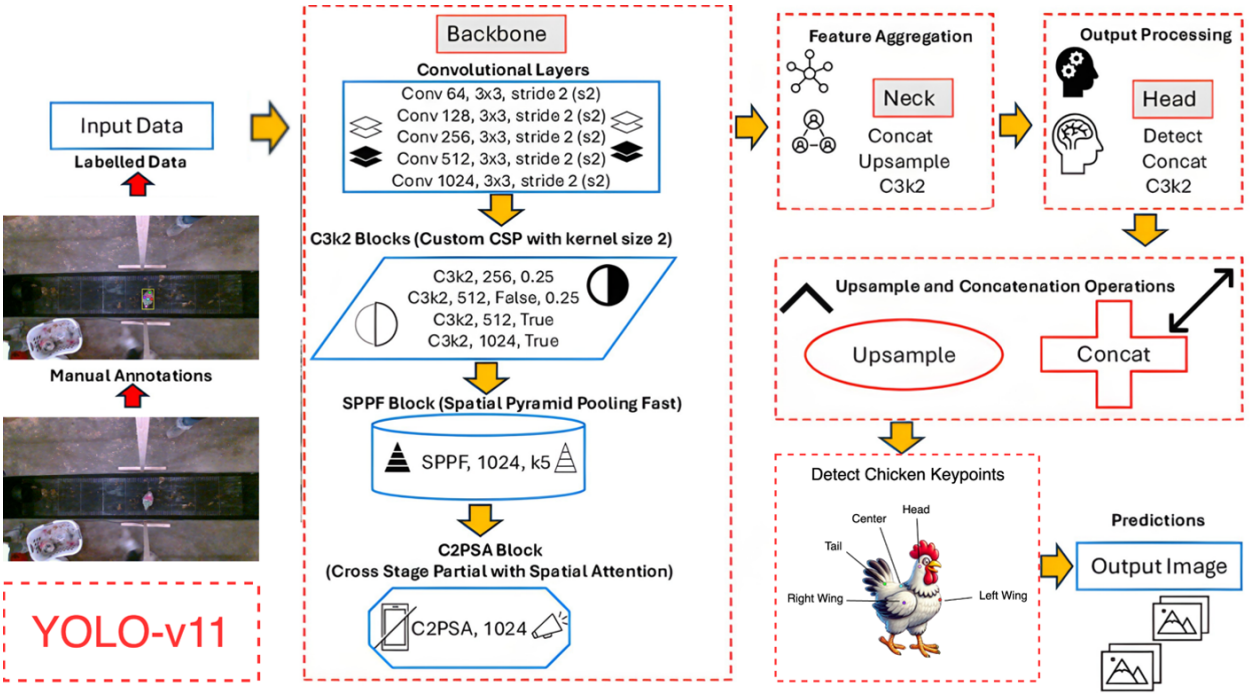

Figure 4.4: Architecture diagram of YOLOv11 used for multi-keypoint pose estimation of broiler chickens. The model detects and localizes keypoints from RGB frames, enabling structured tracking of movement and posture for gait analysis.

Confidence scores were assigned to each detected keypoint, ensuring that low-confidence detections (confidence values less than 0.4) were filtered out to improve overall reliability. To evaluate the accuracy of the pose estimation model, mean Average Precision (*mAP*) and Intersection over Union (*IoU*) were used as performance metrics.

The *mAP* (Eq.4.1) measures the overall detection accuracy by computing the area under the precision-recall curve across multiple confidence thresholds:

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \int_{0}^{1} P_i(R)\, dR \tag{4.1}$$

where $P_i(R)$ represents the precision-recall curve for each keypoint, and $N = 5$ is the number of keypoints of broiler chickens being evaluated. A higher *mAP* indicates a better balance between precision and recall, ensuring fewer false positives and false negatives in keypoint detection. The *IoU* (Eq.4.2) quantifies the overlap between the predicted and ground truth bounding boxes, measuring spatial accuracy:

$$\text{IoU} = \frac{\text{Area of Overlapping}}{\text{Area of Union}} \tag{4.2}$$

where the area of overlap represents the intersection of the predicted and actual bounding boxes, and the area of union is the total area covered by both boxes. A higher *IoU* value indicates a better spatial match between the predicted and ground truth locations.

The detected 2D keypoints were used as input for further processing, where they were back-projected into 3D spatial coordinates using depth data.

4.2.5 Two-Dimension (2D) to Three-Dimension (3D) Back-Projection

To accurately analyze locomotion, the detected 2D keypoints from RGB frames needed to be converted into 3D coordinates using depth data. This transformation allowed for precise tracking of the chicken's motion in 3D space, providing a reliable basis for gait feature extraction. The back-projection process was carried out using the depth values corresponding to each detected keypoint in 2D frames and the intrinsic parameters of the Intel RealSense L515 camera, previously recorded as part of the bag files.

The transformation followed the pinhole camera model with the projection equation formulated in Eq.4.3.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = D \cdot K^{-1} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{4.3}$$

where (X, Y, Z) is a 3D world coordinate; (u, v) represents the 2D pixel coordinates of the detected keypoint in a RGB frame; D is the depth value at that pixel; and K^{-1} is the inverse of the camera's intrinsic matrix, which converts the pixel location into a real-world spatial position.

Once the 3D coordinates were computed for all detected keypoints, they were used to track the chicken's movement over time, enabling the calculation of various kinematic features such as step length, velocity, and body oscillation. Since the depth camera was positioned overhead, the resulting 3D data provided an accurate top-down representation of the broiler's gait dynamics.

4.2.6 Frame Validation

To ensure that only high-quality frames were used for gait analysis, a frame validation module was implemented to automatically filter out frames affected by motion blur, occlusions, and incomplete detections. This step was crucial in maintaining the accuracy of subsequent processing stages, as noisy or occluded frames could lead to incorrect keypoint detection and unreliable gait feature extraction. Figure 4.5 presents examples of both valid and invalid frames, where invalid frames (red box) primarily depict occlusions caused by human intervention during the gait-scoring process, while valid frames (green box) contain clear, unobstructed views of the chicken. These invalid frames needed to be filtered to prevent external influences from affecting locomotion analysis.

To address this, a deep learning-based frame validation model was developed, leveraging both pose estimation-based features and spatial feature extraction from RGB frames to assess frame quality. The validation process began with YOLOv11-based pose estimation models, which extracted human and

Figure 4.5: Examples of valid and invalid frames used in the frame validation process, highlighting human-induced occlusions in invalid frames (red box) and unobstructed frames in valid cases (green box).

chicken keypoints from each RGB frame. The human keypoints, consisting of 17 anatomical joints, were represented by their x, y, and z coordinates along with their corresponding confidence scores, resulting in 68 extracted features. Similarly, the chicken keypoints, including the head, tail, left wing, right wing, and center of the body, were encoded using x, y, and z coordinates and confidence scores, contributing an additional 20 features. Additionally, the 3D bounding box coordinates of both the human and chicken, along with their associated bounding box confidence scores, were extracted. For each 3D bounding box, seven values including the x, y, and z coordinates of the top-left and bottom-right corners of the bounding box and a confidence score for the box were collected. In total, 102 numerical features were derived from the pose estimation output.

In parallel, the RGB frame was resized to 224×224 pixels with three color channels and was passed through a ResNet-18 architecture (He et al., 2016). This model extracted 512 spatial features, capturing high-level visual representations of the frame. The 102 pose-based features were then concatenated with the 512 spatial features, forming a comprehensive 614-dimensional feature vector that integrated both spatial and pose-related information.

This unified feature vector was passed through a fully connected neural network to classify frames as valid or invalid. The network architecture consisted of two hidden layers with 128 and 64 neurons, respectively, followed by a single-node output layer with a sigmoid activation function, which assigned a probability score indicating frame validity. The architecture of the frame validation model is shown in Figure 4.6, detailing the feature extraction, concatenation, and classification process. The validated frames were then passed to the next stages, including platform reorientation and body segmentation.

A total of 2, 000 frames were used for training, with two classes of valid and invalid, while the validation and test sets each contained 250 frames. The learning rate was set to 0.001, with a batch size of 64, and the model was trained for 200 epochs. The model with the lowest training loss was saved in PyTorch format (.pt file).

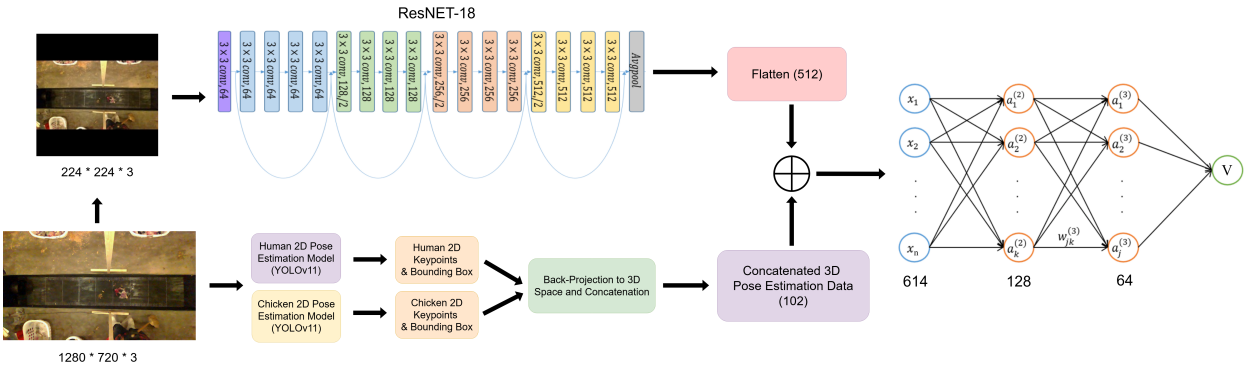

Figure 4.6: Architecture diagram of the frame validation model, representing the integration of pose-based and spatial features, followed by an artificial neural network for classification.

The model was evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score. Accuracy (Eq.4.4) measures the overall correctness of predictions:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.4}$$

where TP and TN represent the number of correctly classified positive and negative samples, while FP and FN denote misclassified instances.

Precision (Eq.4.5) quantifies how many of the predicted positive cases were actually positive. A higher precision value indicated fewer false positives in frame validation:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.5}$$

Recall (Eq.4.6), also known as sensitivity, measures how well the model identifies actual positive cases. A higher recall suggests better detection of invalid frames:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.6}$$

The F1-score (Eq.4.7) provides a balance between precision and recall. This metric was particularly useful in cases where class distribution was imbalanced, ensuring a fair evaluation of model performance:

$$\text{F1}-\text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.7}$$

4.2.7 Platform Reorientation

To ensure consistency in gait analysis, the walking platform needed to be aligned with the camera's perspective in every frame. Variations in the gait scoring wooden platform orientation caused inconsistent

keypoint detections as the spatial space with respect to the camera origin was not the same for all experiments. To correct these misalignments, a platform reorientation module was implemented using edge detection, line detection, and geometric transformations to standardize the walking plane across all frames. Figure 4.7 illustrates several examples of frames where the platform appeared tilted with respect to the camera, demonstrating the necessity of platform reorientation for ensuring consistent spatial references.

Figure 4.7: Examples of frames where the platform is tilted with respect to the camera orientation, demonstrating the necessity of platform reorientation for consistent gait analysis.

The correction process began with Canny edge detection (Canny, 1986), which identified prominent edges in the RGB frames by detecting regions of significant intensity changes. The edge-detected image was then processed using the Hough Line Transform (Duda & Hart, 1972), which extracted the longest horizontal lines in both the upper and lower halves of the frame. These lines were assumed to represent the top and bottom edges of the wooden walking platform. By identifying these platform boundaries, their slopes were computed for each frame. Since individual frame detections could contain noise or slight variations, the mode of the detected slopes across all valid frames in a sequence was calculated to obtain a stable estimate of the platform's orientation.

Once the dominant slope was determined, a geometric transformation was applied to rotate the frame so that the platform aligned horizontally with the camera's field of view. The transformation followed a 2D affine rotation matrix, where each pixel in the original frame (x, y) was mapped to a new location (x', y') using Eq.4.8.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.8}$$

where θ is the calculated average slope of the platform obtained from Hough line detection. After applying this transformation, the detected 2D keypoints and bounding boxes of both the chicken and human were also rotated using the same transformation matrix to maintain spatial consistency. The 3D keypoints and bounding boxes were also rotated along the XY-plane such that the z values remained unchanged but x and y values were rotated using Eq.4.8. The results of this platform reorientation process are illustrated in Figure 4.8. This correction ensured that gait analysis was performed on a consistent spatial reference.

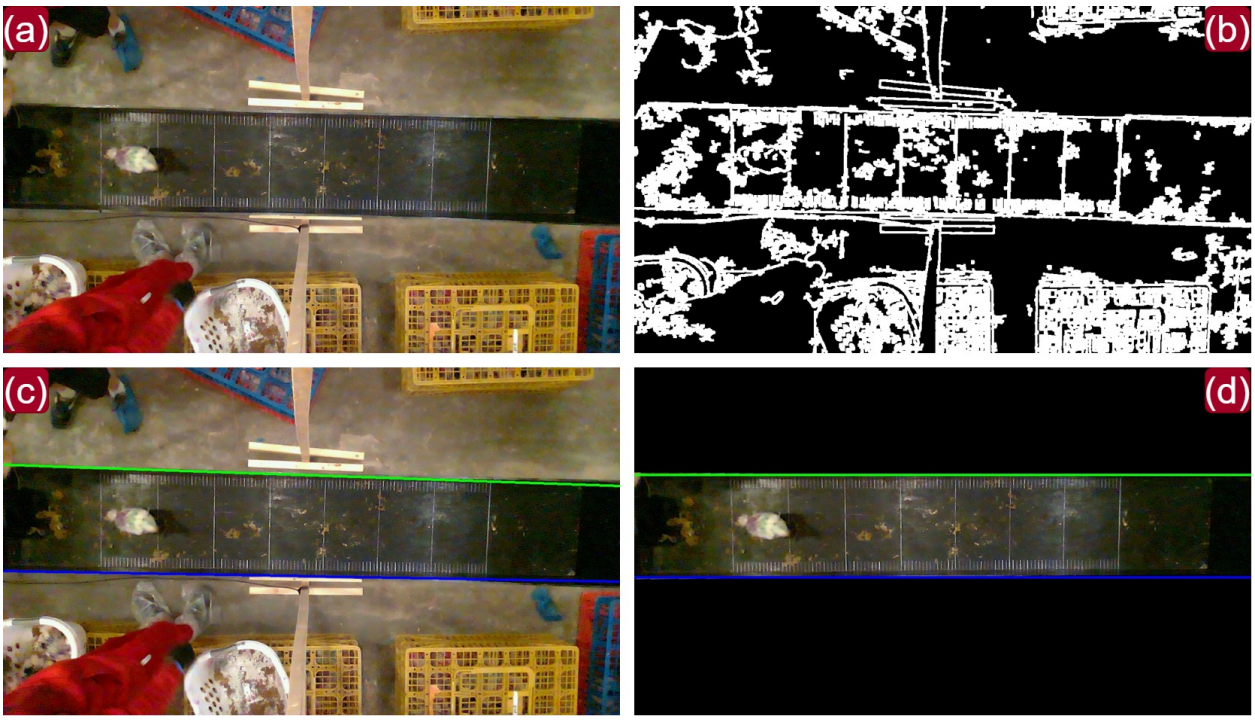

Figure 4.8: Platform reorientation process. (A) Original RGB frame showing a tilted platform. (B) Canny edge detection output highlighting the platform's edges. (C) Hough Line Transform result, where detected edges are marked in green and blue. (D) Reoriented frame after geometric transformation, ensuring the platform is horizontally aligned.

A total of 20,744 was used to evaluate the performance of the platform reorientation. If the edges along the outer boundaries of the walking platform cannot be identified by the proposed algorithm, the corresponding frames were treated as failures. The success rate (accuracy) of edge detection was reported.

4.2.8 Chicken Body Segmentation

Accurately isolating the chicken from the background was a crucial step in ensuring precise feature extraction for gait analysis. Variations in lighting, shadows, and environmental clutter could interfere with pose estimation and kinematic calculations, necessitating a robust segmentation method. To achieve this, the SAM was employed to generate high-quality segmentation masks that effectively separated the chicken from its surroundings.

SAM was chosen due to its zero-shot segmentation capabilities, meaning it did not require retraining for poultry-specific applications. The model used a prompt-based segmentation approach, where bounding box information from the YOLOv11 pose estimation model was used as prompt input to generate

precise segmentation masks. The SAM generated three mask samples with different confidence scores, and the mask with the highest confidence score was retained.

Once the segmentation masks were generated, they were applied to the corresponding RGB frames, removing background pixels and retaining only the chicken's body region. This process significantly improved the accuracy of subsequent pose estimation, keypoint tracking, and kinematic feature extraction by eliminating irrelevant regions that could introduce noise into the analysis. The architecture of the SAM used for segmentation is illustrated in Figure 4.9, showing the end-to-end segmentation pipeline.

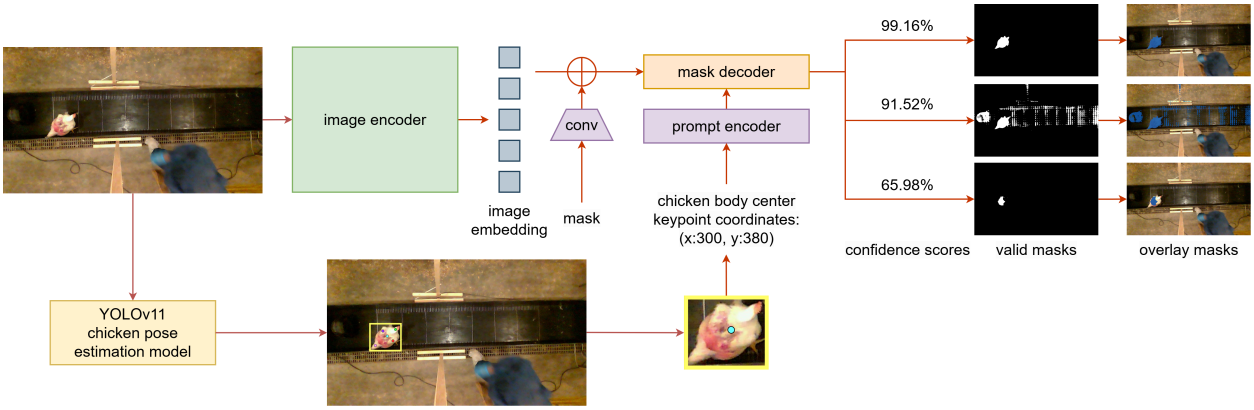

Figure 4.9: Architecture diagram of the Segment Anything Model used for chicken body segmentation, illustrating the prompt-based segmentation process and mask generation pipeline. The model generates three masks with different confidence scores, and the highest score is chosen as the best chicken body mask.

4.2.9 Feature Extraction for Gait Analysis

Once the chicken's body was segmented and the 3D pose data was reconstructed, key kinematic behavior metrics were extracted to quantify locomotion patterns. These features provided objective measures of movement efficiency, mobility impairments, and overall gait dynamics, forming the basis for automated gait scoring. The feature extraction process started with a preprocessing step and was then followed by calculating kinematic features, utilizing external contextual features, and visualizing chicken movement in each of the 2D planes as a heatmap.

Data Preprocessing: Each analyzed video contained a number of valid frames, from which movement heatmaps and kinematic features were computed. Before heatmap generation or kinematic feature extraction, several preprocessing steps were conducted. First, the chicken's initial valid location was defined as the movement origin, and the 3D coordinates for each subsequent frame n were translated by subtracting the coordinates of the first frame. Next, outliers along each axis were detected via the interquartile range (IQR) method (Q1=0.25, Q3=0.75, cutoff=1.5) (Kraaikamp & Meester, 2005) and replaced through linear interpolation. To ensure consistent scaling, X-axis coordinates were clipped to [0, 250], while Y-axis and Z-axis coordinates were clipped to [-40, 40] and [-25, 25], respectively. The ranges for each axis were

set according to the dimensions of the wooden gait scoring platform and the height of the camera. Once those steps were accomplished, the data was ready to be further processed for kinematic feature extraction.

Chicken Movement Heatmap: A chicken movement heatmap was generated to visualize the spatial distribution of movement in the XY, XZ, and YZ planes. Each heatmap was saved as an RGB image with a dimension of 1280×240 pixels. Once the individual heatmap for each plane was generated, they were then stacked vertically to form a combined RGB frame with a dimension of 1280×720 pixels. A sample of three individual heatmaps and a stacked heatmap are represented in Figure 4.10.

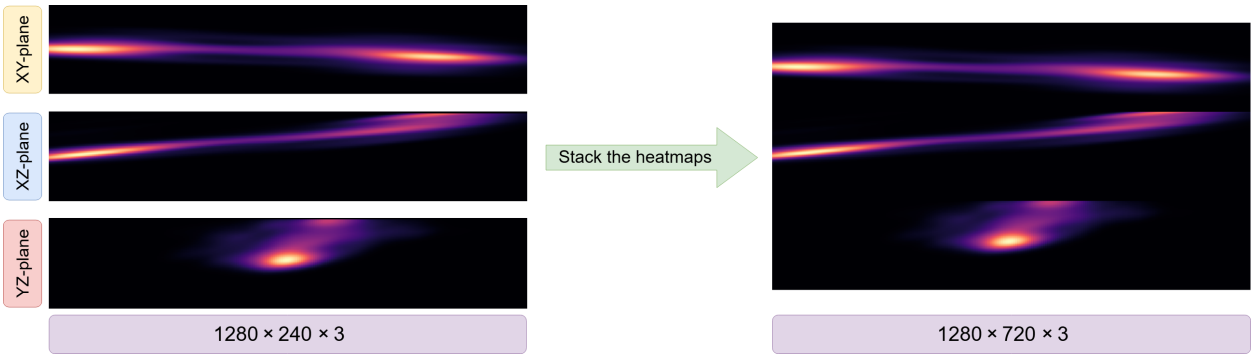

Figure 4.10: Heat maps of a broiler chicken walking across the measure platform. (Left) Three individual heatmaps representing chicken movement in the XY, XZ, and YZ planes; (Right) Stacked chicken movement heatmap.

The stacked chicken movement heatmap provided an intuitive representation of the chicken's most frequently occupied positions, highlighting movement asymmetries or irregular locomotion patterns. Heatmaps were particularly useful for identifying imbalanced gait cycles, where a chicken may favor one side due to limb weakness or pain-related avoidance behaviors.

Total Traveled Distance (*cm*): It represents the cumulative path length of each of the chicken's keypoint movements throughout the recorded sequence in each of the 2D planes and also in 3D space. The formula represented in Eq.4.9 represents how the total traveled distance is calculated in the XY-plane and a similar calculation was done for calculating the total traveled distance in the XZ-plane and YZ-plane.

$$TD_{XY} = \sum_{i=1}^{N-1} \sqrt{(X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2} \tag{4.9}$$

The total traveled distance in 3D space (Eq.4.10) is computed by summing the Euclidean distances between consecutive positions of a chicken's center keypoint across all frames:

$$TD_{3D} = \sum_{i=1}^{N-1} \sqrt{(X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2 + (Z_{i+1} - Z_i)^2} \tag{4.10}$$

where (X_i, Y_i, Z_i) represents the 3D coordinates of the chicken's body center at frame i, and N is the total number of frames. This metric was calculated and recorded for all of the five chicken body keypoints. This metric helps assess mobility levels, as birds with severe gait impairments typically exhibit shorter traveled distances.

End-to-End Displacement (*cm*): While *total traveled distance* summed all movement between consecutive frames, *end-to-end displacement* focused solely on the difference between the first and final positions. Specifically, for a chicken's center keypoint, the 3D displacement ΔD was calculated using Eq.4.11 as:

$$\Delta D = \sqrt{(X_N - X_1)^2 + (Y_N - Y_1)^2 + (Z_N - Z_1)^2} \tag{4.11}$$

where (X_1, Y_1, Z_1) and (X_N, Y_N, Z_N) represent the 3D coordinates of the chicken's center in the first and last frames, respectively. Analogous calculations can be performed for each individual axis to capture directional preferences in movement.

Average Speed (*cm/s*): It measures the overall movement rate of a chicken (Eq.4.12) and is defined as the total traveled distance divided by the total duration of the sequence.

$$S_{\text{avg}} = \frac{TD_K}{T} \tag{4.12}$$

where TD_K is the total traveled distance in the desired plane or in 3D space (K) and T represents the total time duration of the sequence in seconds. In our experiments, the data synchronized RGB-D frames were extracted in ~$15FPS$; therefore, T was set to $N/15$ where N is the total number of frames that the chicken was captured in the video. This feature provides insight into locomotion efficiency, distinguishing between birds with fluid motion and those with restricted mobility.

Average Velocity (*cm/s*): While speed accounts for total movement regardless of direction, average velocity considers only the displacement between the start and end positions of the movement (Eq.4.13).

$$V_{\text{avg}} = \frac{\Delta D_K}{T} \tag{4.13}$$

where ΔD_K is the straight-line distance between the initial and final positions of a chicken's center and T is the total time duration of the sequence in seconds. A lower velocity relative to speed suggests frequent direction changes or instability in movement, which may indicate gait impairments.

Average Acceleration (*cm/s²*): To quantify the changes in the rate of motion, *average acceleration* is derived by examining the *difference in speeds* between consecutive frames. For example, if $S_{k,i}$ and $S_{k,i-1}$ are the calculated speeds in frame i and $i-1$ in the desired plane or 3D space (k), and $\Delta S_{k,i} = S_{k,i} - S_{k,i-1}$ is the change in speed for the two consecutive frames in the same plane or 3D space, the acceleration $a_{k,i}$ at one time step Δt can be calculated using Eq.4.14:

$$a_{k,i} = \frac{\Delta S_{k,i}}{\Delta t} \tag{4.14}$$

Averaging $a_{k,i}$ across the entire sequence yields the *average acceleration*, which helps distinguish abrupt changes in motion, often indicative of lameness or instability.

Average Body Turn Angle (degree): This feature measures changes in the chicken's heading vector across consecutive frames to detect sudden turns or erratic rotation. Let $\mathbf{v}_i$ and $\mathbf{v}_{i-1}$ be the 3D vectors from the body center to the head in two consecutive frames. The turn angle θ is computed via the dot product formula presented in Eq.4.15:

$$\theta = \cos^{-1}\left(\frac{\mathbf{v}_i \cdot \mathbf{v}_{i-1}}{\|\mathbf{v}_i\|\|\mathbf{v}_{i-1}\|}\right) \tag{4.15}$$

Averaging θ across the entire set of frames yields the average body turn angle, which could be an indicator of chicken body health status.

Stop Time Ratio: It quantifies the fraction of frames where the chicken exhibits negligible movement, defined as frames where the displacement is below a 1-centimeter threshold and was formulated as Eq.4.16.

$$P_{\text{stop}} = \frac{N_{\text{stop}}}{N} \tag{4.16}$$

where N_{stop} is the number of frames with displacement below 1 centimeter compared to the previous frame, and N is the total number of frames. The rationale for this feature is that birds with lameness or mobility restrictions tend to have a higher stop time proportion, indicating increased stationary behavior.

Additional Contextual Features: Classification was primarily driven by movement dynamics, captured through heatmap-based spatial representations and essential kinematic metrics. Additionally, we incorporated a range of contextual features such as bird age, body weight, dietary copper level, and litter information to further strengthen the model's predictive accuracy.

Concatenated Kinematic and Contextual Feature Vector. In order to use the numerical features for the gait score prediction, these features should be collected and concatenated as a feature vector. As part of this feature vector, total traveled distance, total displacement, average speed, average velocity, average acceleration, and stop time ratio, computed in three 2D planes (XY, XZ, YZ) and in full 3D space (XYZ) for each of the 5 chicken body keypoints, yielded a total of 120 numerical values. Also, average body turn angle, calculated for 2D planes (XY, XZ, YZ) and in full 3D space (XYZ), added 4 other features to the vector. Lastly, the four additional contextual features (bird age, body weight, dietary copper level, and litter information) were added to the vector to make it a total of 128 features. The next section describes how these features were used to train a multi-layer perceptron (MLP) classifier for automatic gait score prediction.

4.2.10 Gait Score Prediction Model

To classify broiler locomotion into different gait scores, a multi-layer perceptron (MLP) classifier was trained using a combination of spatial and kinematic features extracted from the movement of each bird. The model assigned a gait score of 0, 1, or 2.

As explained in the previous section, multiple numeric and visual information were extracted from the 3D coordinates of the chicken in each video. The gait score prediction model uses two sources of data, including the stacked heatmaps represented in RGB images with the dimension of 1280×720 pixels (Figure 4.10) and the unified kinematic and contextual feature vector, including 128 unique features.

To predict the gait score, the stacked heatmap image for each video was resized to 224×224 pixels, passed through a ResNet-18 backbone, and 512 spatial features were extracted. The 512-dimensional feature vector was then subsampled to a 128-dimensional flattened feature vector using an average pooling operation to capture the summarized spatial characteristics of the chicken's movement patterns. Then, the 128 key kinematic and contextual features were concatenated with the 128-dimensional spatial feature vector, forming a final 256-dimensional input feature vector for classification.

The 256-dimensional feature vector was then passed through an MLP network, which consisted of two hidden layers with 128 and 64 neurons, respectively, each using ReLU activation functions to enhance non-linearity and feature separability. The output layer contained three neurons, each representing one of the valid gait score values. The full architecture of the gait score prediction model is illustrated in Figure 4.11.

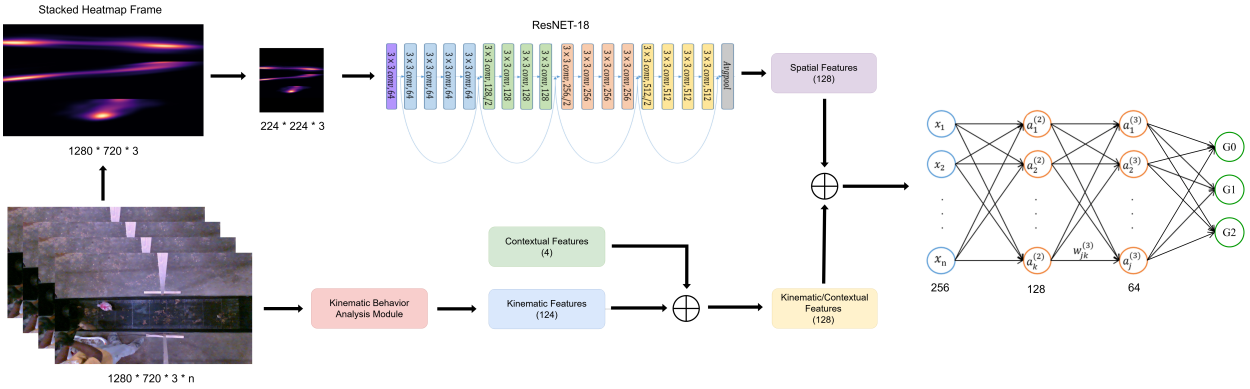

Figure 4.11: A visualization of the gait score prediction model representing how the kinematic and contextual features are being concatenated with the spatial features (extracted from the stacked heatmap frame) and passed to a multi-layer perceptron to predict the gait score classification for a given video. The probability scores of classes 0, 1, and 2 are represented as $G0$, $G1$, and $G2$ in the diagram.

The model was trained on a balanced dataset of 180 videos (with a total of 139, 743 frames), consisting of 60 videos for each gait score class (0, 1, and 2), gathered from broiler flocks at 6 and 7 weeks of age. Three humans were involved to evaluate the gait score of each chicken, and major voting was used to assign the final class label for each chicken. Also, the original dataset had almost 534 samples with a gait score of 0, 293 samples with a gait score of 1, and 61 samples with a gait score of 2 for weeks 6 and 7 of flock age.

Because of that, to keep the balance between classes, 60 samples from each class were selected randomly, shuffled, and split into 70% training, 15% validation, and 15% testing (i.e., 126 videos for training and 27 videos each for validation and testing). The number of samples for each class was equal to the number of samples for the other classes in the training, validation, and test sets. Model training proceeded for 100 epochs with a batch size of 8 and a learning rate of 0.001. We employed batch normalization (Ioffe & Szegedy, 2015) to normalize the kinematic features, which otherwise varied significantly across samples and could lead to exploding or vanishing gradients. Additionally, a dropout layer (Srivastava et al., 2014) with a 50% drop rate was applied after each hidden layer to reduce overfitting by randomly deactivating half of the neurons during training. The cross-entropy loss function (Mao et al., 2023) was used as the training criterion and was defined by Eq.4.17 as:

$$L = -\sum_{i=1}^{c} y_i \log(p_i) \tag{4.17}$$

where y_i is the ground-truth class label (in one-hot encoding) and p_i is the predicted probability for that class. We used Adam as the optimization function (Kingma & Ba, 2014) with the aforementioned hyperparameters to train the model effectively.

4.3 Results

4.3.1 Chicken and Human Pose Estimation Performance

As discussed earlier, a medium-sized YOLOv11-pose model (yolov11m-pose) was trained to detect five key anatomical points (head, tail, left wing, right wing, and center of body) on broiler chickens at weeks 6-7, from which the broilers presented more severe leg issues than they did in younger ages. The training set comprised $1,000$ labeled RGB frames, while separate validation and test sets each contained 100 frames. Over 100 epochs of training, the model converged quickly, as illustrated in Figure 4.12 (training curves for precision, recall, and *mAP* values). Final evaluations on the test set yielded near-perfect detection metrics, including a 100.00% precision, 99.98% recall, 99.50% *mAP*50, and 86.94% *mAP*50–95.

To accommodate human operators occasionally entering the platform area, a publicly available YOLOv11-pose model (pre-trained on large-scale human keypoint datasets) was employed. Empirical trials indicated that using a confidence threshold of 25% and an IoU threshold of 40% provided an optimal balance between accurate human detection and minimal false positives. With these thresholds, frames containing partial or obstructed human poses were still recognized, helping to later exclude occluded bird keypoints in the pipeline.

An example of simultaneous chicken–human pose estimation is shown in Figure 4.13, where each chicken keypoint (colored dots) and human keypoint (skeleton overlay) are identified at high confidence (> 0.92 for human keypoint detection and > 0.87 for chicken keypoint detection). These results confirm that the YOLOv11-pose framework was sufficiently robust to handle variable poses, lighting conditions, and minor occlusions in the broiler house environment.

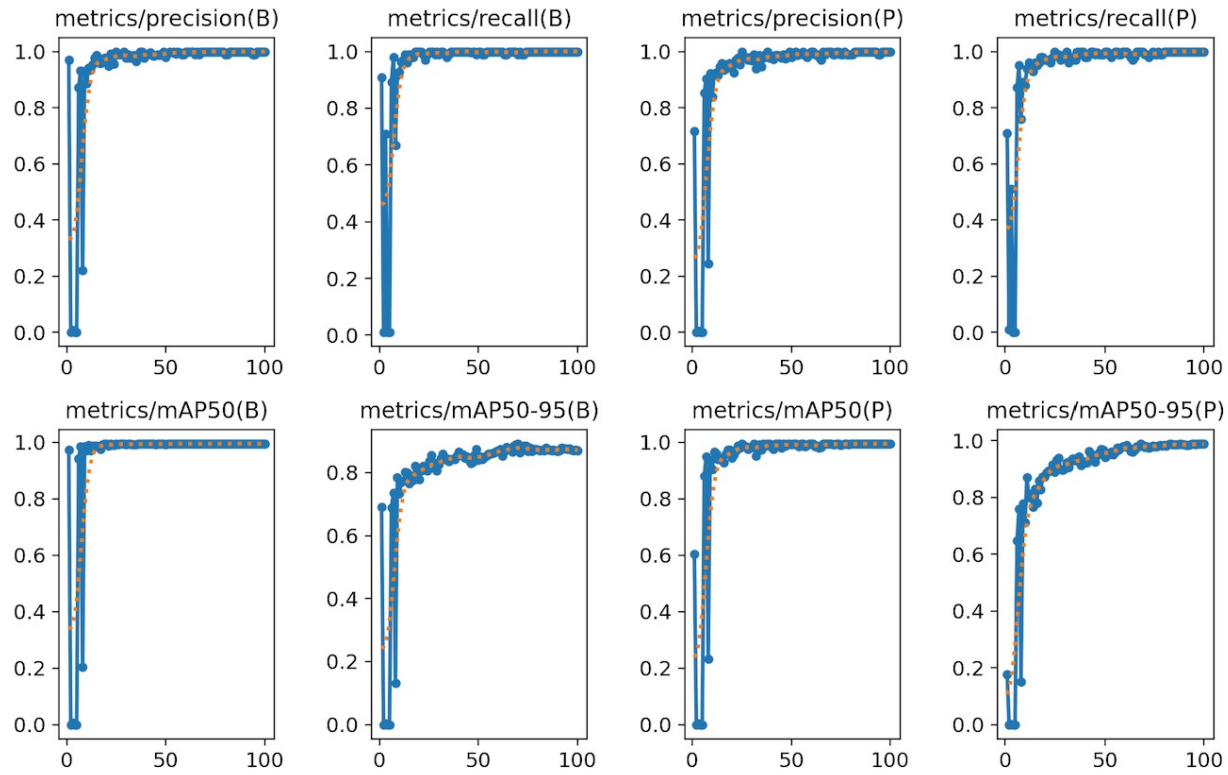

Figure 4.12: Training and validation curves over 100 epochs for the YOLOv11m-pose model. The top row shows precision and recall values for bounding-box (B) and keypoint (P) detection, while the bottom row presents $mAP50$ and $mAP50\text{-}95$ metrics for both bounding-box and keypoint.

4.3.2 Frame Validation Results

A separate classification framework that combined with ResNet-18, human/chicken keypoint detection results, and artificial neural network (Figure 4.6) was trained to distinguish between valid frames (clearly visible chicken with minimal occlusion) and invalid frames (e.g., blocked by humans, heavy motion blur). Figure 4.14 illustrates how the classification metrics (accuracy, precision, recall, and F1-score) evolved over 100 training epochs. The model achieved a 95.60% accuracy, 95.36% precision, 98.93% recall, and 97.11% F1-score on the test set, indicating highly reliable filtering of unusable frames. Occasional dips in the learning curves were observed around epochs 40–50 but quickly stabilized, reflecting transient overfitting episodes that were corrected by subsequent training updates. Overall, these results confirm the model's capability to robustly identify and discard frames plagued by human interference, partial chicken views, or poor visibility. This automated validation step substantially improved downstream processing, ensuring only high-quality frames contributed to kinematic feature extraction and subsequent gait analysis.

Figure 4.13: Examples of frames demonstrating simultaneous chicken and human keypoint detection. The YOLOv11m-pose model identifies five anatomical points on the chicken (colored dots for head, tail, wings, and body center), while a pre-trained human pose estimator overlays the human operator's body and head with up to 17 detected keypoints.

4.3.3 Platform Reorientation Results

Out of all valid frames identified, a subset appeared tilted relative to the camera's field of view, necessitating reorientation to maintain a consistent walking perspective. Overall, the reorientation algorithm successfully identified the platform edges in 97.14% of 20, 744 test frames, with an average detected slope of 1.36 ± 3.42. By removing platform tilt, the bounding box coordinates and keypoints for each chicken became more consistent across frames, facilitating downstream segmentation and kinematic feature extraction. Only a small fraction of frames (2.86%) could not be reliably reoriented (e.g., heavily occluded edges), and those were discarded. As a result, the final dataset was standardized in orientation, reducing variations that could otherwise distort velocity or step-length estimates.

4.3.4 Chicken Body Segmentation and 3D Back-Projection

After reorienting the walking platform (Section 4.3.3), the SAM was used to generate 2D segmentation masks of the chicken body. In over 97% of 20, 961 validated frames, SAM produced clean, continuous segmentations that accurately encompassed the bird's outline, even with moderate lighting variations or

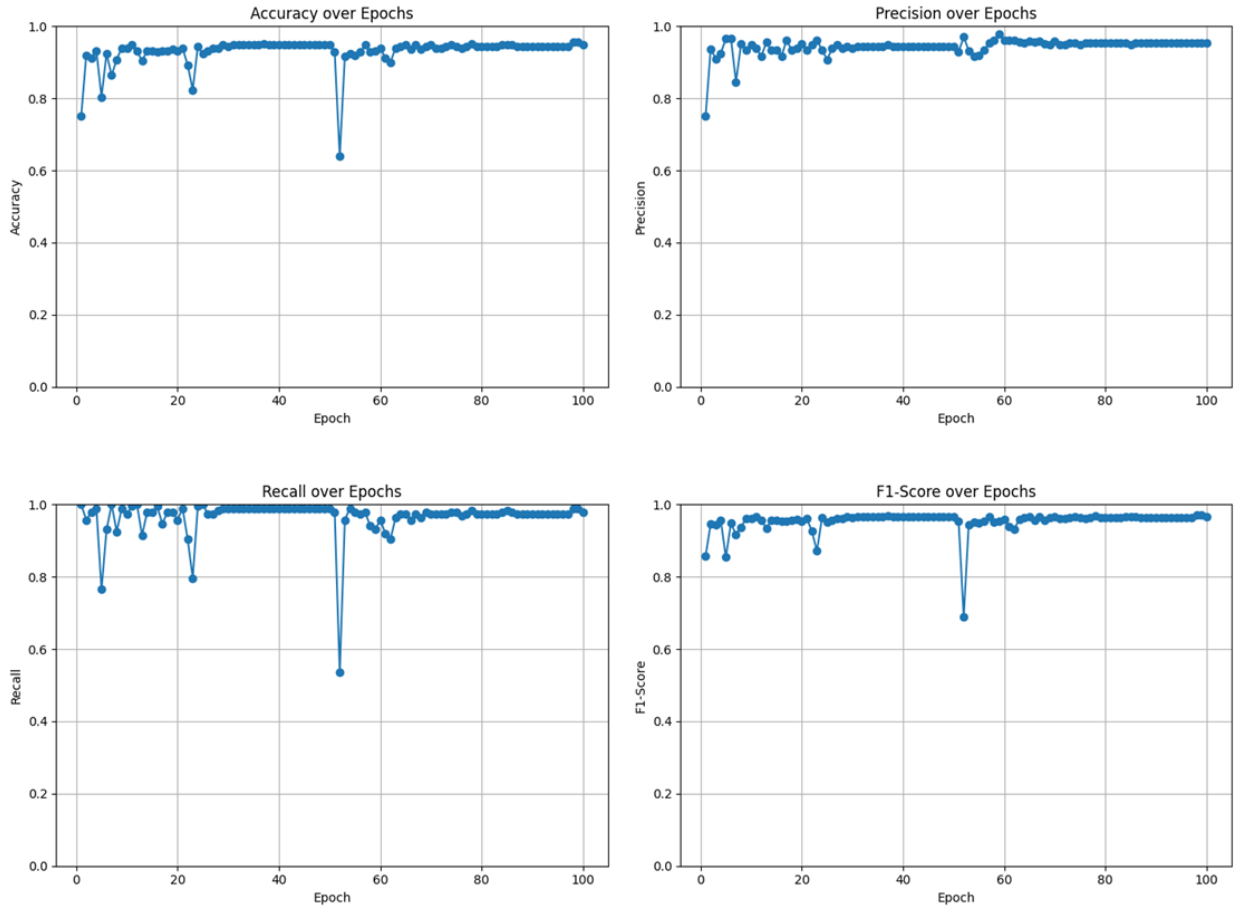

Figure 4.14: Learning curves of the frame-validation classification framework across 100 epochs, showing accuracy, precision, recall, and F1-score trends. Minor fluctuations are evident around the 40th to 60th epoch, but the metrics ultimately converge to high values, confirming the robustness of the frame filtering process.

minor occlusions. Only a small fraction of frames (~3%) displayed over-segmentation or disjoint masks, often in cases where the chicken had rapid moves or rapid wing openings. Leveraging the depth data and camera intrinsics, these 2D masks were then mapped into 3D space to form coherent chicken point clouds. Figure 4.15 illustrates this process by starting from a reoriented frame, followed by pose estimation, instance segmentation, and mesh generation. The resulting meshes preserved body shape with minimal background noise, enabling more reliable estimation of 3D kinematic features in subsequent analysis.

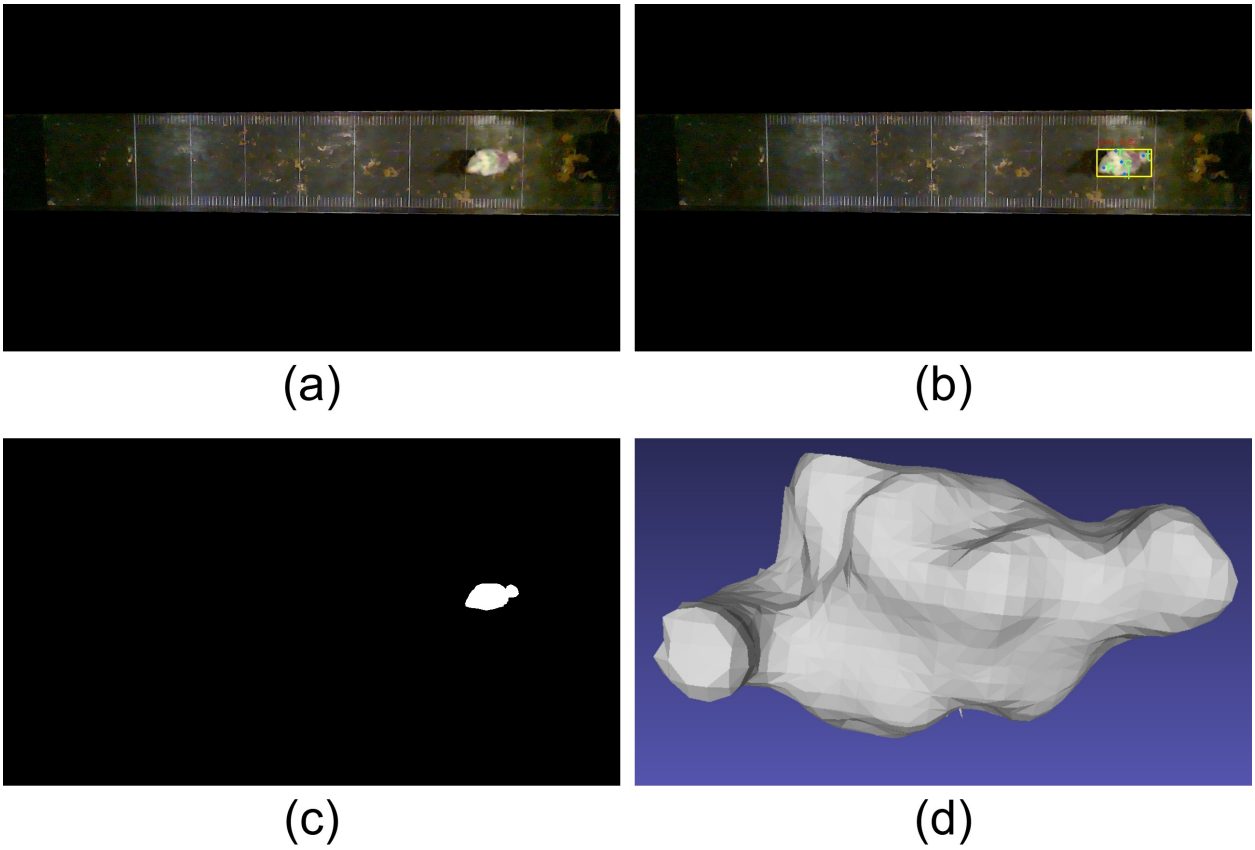

Figure 4.15: A depiction of (a) a reoriented RGB frame, (b) the result of YOLOv11 pose estimation model for the detected chicken, (c) the 2D segmentation mask generated by SAM having the center of the chicken body as input, and (d) the corresponding 3D point cloud after back-projection. The mesh closely matches the chicken's silhouette and excludes extraneous background objects.

4.3.5 Kinematic Feature Extraction and Distributions

Following the segmentation and 3D reconstruction (Section 4.3.4), a comprehensive set of kinematic metrics (Eqs.4.9 to 4.16) was derived from the processed coordinates of each chicken. Table 4.2 summarizes several key features averaged within each gait score category (0, 1, and 2), including total traveled distance, end-to-end displacement, average speed, average velocity, average acceleration, average body turn angle, and stop time ratio in 3D space. In line with expectations, higher gait impairment (score 2) correlated with lower traveled distances and speed, alongside more frequent stopping. Meanwhile, chickens scored as 0 tended to traverse the entire wooden platform with relatively smooth and continuous motion.

To visualize how these locomotion patterns manifest spatially, Figure 4.17 displays three sample stacked heatmaps (one per gait score). A chicken with a gait score of 0 covered most of the platform area, reflecting active walking and minimal stopping. By contrast, a bird with a gait score of 1 exhibited shorter overall

Table 4.2: Representative kinematic metrics (Mean $\pm$ SD) extracted from 3D coordinates for each gait score category ($n = 60$ birds per category). (Note: Values represent the 3D metrics computed using the center keypoint. For planes XY, XZ, and YZ, similar calculations were performed but are omitted here for brevity.)

Kinematic Feature	**Gait Score** 0	**Gait Score** 1	**Gait Score** 2
Total Traveled Distance (cm)	1643.48 ± 3282.13	1561.49 ± 3361.94	1347.36 ± 2359.84
End-to-End Displacement (cm)	182.25 ± 33.79	95.24 ± 42.51	49.59 ± 41.43
Average Speed (cm/s)	47.83 ± 72.48	33.58 ± 83.21	31.62 ± 35.0
Average Velocity (cm/s)	7.47 ± 5.56	2.20 ± 1.15	0.99 ± 0.83
Average Acceleration (cm/s^2)	-0.04 ± 0.05	-0.02 ± 0.02	-0.07 ± 0.26
Average Body Turn Angle (degree)	249.55 ± 92.78	212.53 ± 84.64	221.0 ± 115.18
Stop Time Ratio (%)	0.47 ± 0.17	0.63 ± 0.12	0.65 ± 0.16

displacement and a more fragmented path, while a bird with a gait score of 2 barely left its starting point. These differences reinforce how the underlying kinematic features capture relevant mobility impairments, ultimately informing automated gait score prediction in the next section.

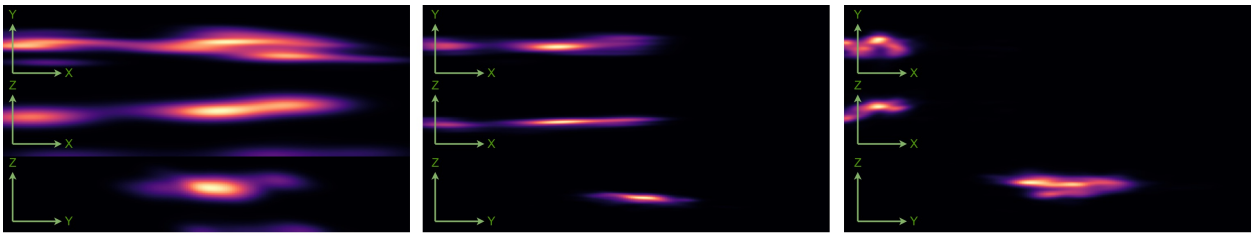

Figure 4.16: Sample stacked heatmaps illustrating the distribution of chicken movement in the XY, XZ, and YZ planes. Each column corresponds to one bird, each with a different gait score (0, 1, 2). The gait score 0 chicken (a) traversed nearly the entire platform area, while the gait score 1 bird (b) shows a partially covered pathway, and the gait score 2 chicken (c) remained mostly near its starting position. These heatmaps visually confirm the more limited mobility for higher gait scores.

4.3.6 Gait Score Classification Performance

An MLP classifier was trained to predict the final gait scores (0, 1, or 2) using the kinematic and contextual features described in Section 4.2.9 of this chapter. After 100 training epochs, the classifier achieved a 93.34% accuracy, 95.56% precision, 91.16% recall, and 93.31% F1-score when evaluated on the test set.

Convergence trends and final model outcomes are illustrated in Figure 4.17, where subfigures (a) and (b) show the evolution of training versus validation metrics over epochs, and subfigure (c) provides a confusion matrix of predicted vs. ground-truth classes.

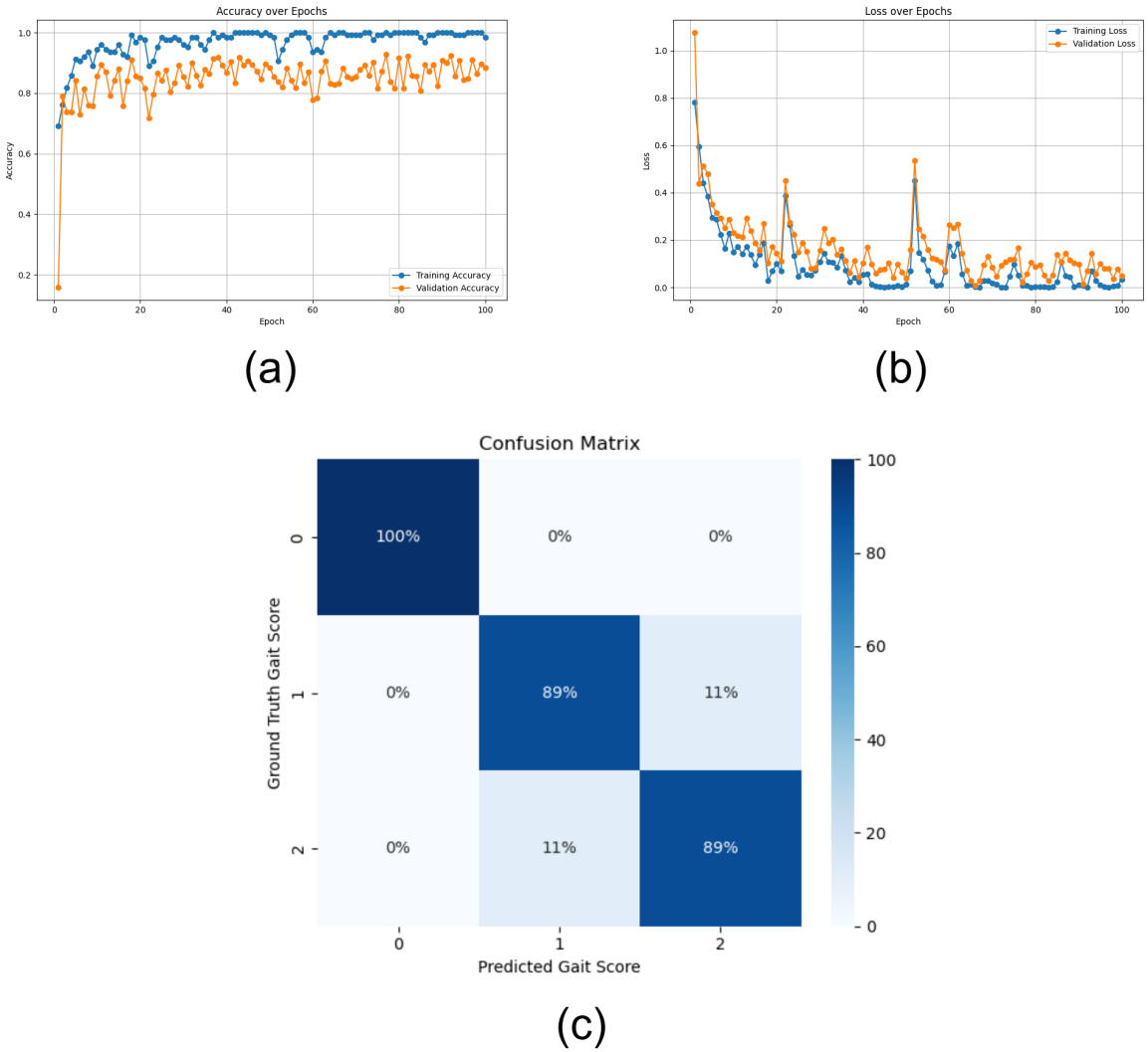

Figure 4.17: Gait score classification outcomes for the MLP classifier over 100 epochs: (a) Accuracy plot showing continuous improvement in both training (blue) and validation (orange); (b) cross entropy loss curves highlighting learning progress and minor overfitting events; and (c) confusion matrix chart indicating perfect separation of gait score 0 but moderate misclassification between gait scores 1 and 2.

As shown in Figure 4.17.(a), accuracy improved steadily during early training and stabilized above 90% after approximately 40–50 epochs. Figure 4.17.(b) displays the training and validation loss curves, revealing a consistent decrease over time, although occasional fluctuations indicate minor overfitting episodes that resolved with further training. Notably, in the confusion matrix of Figure 4.17.(c), the model perfectly classified all score 0 samples, whereas scores 1 and 2 classes exhibited a small overlap of 11%. These misclassifications likely stemmed from subtle similarities in partially impaired locomotion.

Overall, the robust classification performance demonstrates that the extracted features successfully capture meaningful movement differences, enabling reliable automated gait scoring in broiler chickens.

4.3.7 Processing Time Performance

After transferring the ROS bag files to an external hard drive, each video underwent a series of post-processing steps to determine gait scores, which were discussed in detail in previous sections. The average duration of the dataset videos was around 52 seconds, equal to 780 frames, and a sample video of this duration was used to calculate the processing time of each module in the proposed pipeline.

The first process performed on the bag files was to extract synchronized RGB and depth frames. Extracting the frames took about 43 seconds for the sample video (0.05 seconds per frame). Human and chicken pose estimation then required 39 seconds per video (also 0.05 seconds per frame), followed by a 2D-to-3D keypoint transformation that finished in 0.8 seconds (0.001 per frame). Frame validation added another 5.46 seconds (0.007 seconds per frame), whereas platform reorientation was more computationally intensive at 0.22 seconds per frame (171.6 seconds per video). Chicken body segmentation used 0.46 seconds per frame, totaling 358.8 seconds, and the kinematic feature extraction module took 39.48 seconds (0.05 seconds per frame). Finally, gait score classification consumed 6.24 seconds (0.008 seconds per frame). Summing these modules yielded an average processing speed of 0.845 seconds per frame, or roughly 664 seconds to fully process one 52-second video from start to finish.

4.4 Discussion

This study presented a novel three-dimensional deep learning pipeline that effectively bridged the gap between traditional manual gait scoring and automated, objective assessment in broiler chickens in 3D space. By integrating synchronized RGB-D data acquisition with advanced processing modules such as a custom-trained YOLOv11-based pose estimation network, precise 2D-to-3D back-projection using camera intrinsics, and robust segmentation through SAM, detailed kinematic features were successfully extracted to characterize chicken locomotion. The pipeline's comprehensive preprocessing and frame validation modules ensured that only high-quality, well-aligned frames contributed to the analysis, thereby mitigating issues related to occlusions and motion blurs. Consequently, the MLP classifier, which leveraged both spatial features from stacked movement heatmaps and a suite of kinematic descriptors, achieved a high predictive performance with 93.34% accuracy, 95.56% precision, 91.16% recall, and a 93.31% F1-score. These results underscore the system's ability to objectively capture subtle gait variations and demonstrate its potential as a scalable, cost-effective solution for precision poultry welfare monitoring.

Several methodologies had been explored for automated gait scoring in poultry, each employing distinct sensing and computational techniques. One approach (Aydin, 2017) used a 3D vision camera system to measure inactivity indicators such as the number of lying events and latency to lie down, achieving a strong correlation with manual gait scores and an accuracy of 93.34%. Another method (Nääs et al., 2018) utilized a paraconsistent logic-based model, which aimed to estimate gait scores based on walking

speed and acceleration. While this approach demonstrated strong performance in distinguishing severe cases of lameness, its classification accuracy for intermediate gait scores was lower (84%). Later, Nääs et al. (de Alencar Nääs et al., 2021) introduced a decision tree-based machine learning model that achieved higher accuracy (91%) and better recall (86%) for lame broilers, making it a notable improvement over prior logic-based estimation models. Another method introduced by Nasiri et al. (Nasiri et al., 2022) advanced this field by implementing a CNN with long short-term memory (LSTM) framework that leveraged pose estimation and action recognition for tracking sequential movement patterns, achieving 95% overall accuracy, demonstrating the benefits of deep learning in gait analysis. The performance of these automated gait scoring models is summarized in Table 4.3.

Table 4.3: Comparison of classification accuracies for automated gait scoring methods.

Reference	**Model/Methods**	**Classification Accuracy (%)**
(Aydin, 2017)	3D vision camera tracking number of lying events and latency to lie down	93.34
(Nääs et al., 2018)	Paraconsistent logic-based gait score estimation	84.00
(de Alencar Nääs et al., 2021)	Decision tree-based machine learning approach	91.00
(Nasiri et al., 2022)	Pose estimation and action recognition with CNN and LSTM	95.00
Current Study	3D deep learning pipeline (ResNet-18, YOLO, SAM, MLP classifier)	93.34

The comparison highlighted the progression of gait scoring methodologies from heuristic models to deep learning approaches. The 3D vision approach (Aydin, 2017) effectively measured inactivity indicators but required a structured test corridor, limiting its real-world applicability. The paraconsistent logic-based model (Nääs et al., 2018) exhibited limited accuracy for intermediate classifications, while the decision tree approach (de Alencar Nääs et al., 2021) improved classification consistency, particularly for severe lameness cases. The CNN-LSTM model (Nasiri et al., 2022) outperformed prior methods by combining pose estimation with temporal modeling, achieving the highest classification accuracy. In contrast, the proposed deep learning pipeline integrated multi-modal data processing with object detection (YOLO), advanced segmentation (SAM), and MLP classifiers to achieve precise gait characterization. The observed improvements in precision and recall indicated a stronger capability to capture subtle gait variations, mitigating classification errors seen in traditional methods. Additionally, while previous models often required structured environments, the proposed system was designed for broader adaptability and scalability in real-world poultry monitoring applications.

Notwithstanding the state-of-the-art performance and automation capabilities of the proposed system, certain limitations and challenges had been identified. First, the initial installation and calibration of the system, which included precise camera placement and depth sensor tuning, were recognized to require intermediate-level technical expertise, potentially posing challenges for widespread adoption in commercial poultry farms without appropriate training or support infrastructure. A dependency on

high-quality depth data had been observed, as variations in lighting conditions and reflective surfaces may adversely affect the accuracy of the Intel RealSense L515 depth measurements. Moreover, the computational complexity of the deep learning pipeline, particularly within the YOLO-based pose detection and SAM segmentation modules, necessitated the use of high-performance processing hardware. Consequently, real-time deployment may be constrained in farm environments that lack dedicated GPUs or edge computing solutions. It should also be acknowledged that further validation is required to ensure model generalization across different farm conditions, as variations in flooring and camera positioning could potentially influence performance. Although the MLP gait score classifier has demonstrated strong predictive capability, robustness could be enhanced by expanding the dataset to include more per-class samples and a broader range of poultry breeds and environmental conditions.

To enhance the proposed 3D deep learning-based gait scoring system, several avenues for future research and development are recommended. Integrating multi-camera fusion techniques can mitigate occlusion issues and provide comprehensive monitoring of broiler movements. For instance, a multi-view state-action recognition framework effectively combined data from multiple cameras to improve trajectory generation accuracy (Asali et al., 2023). Similarly, a super-resolution fusion optimization method for multi-object chicken detection was proposed to enhance detection accuracy in challenging environments (Z. Wu et al., 2023).

Incorporating multi-modal sensor data, such as thermal imaging, can provide additional insights into broiler health and behavior. A YOLO-based model utilizing both visual and thermal images was developed to detect pathological phenomena in broilers, demonstrating the potential of thermal data in monitoring poultry health (Elmessery et al., 2023). Additionally, zero-shot image segmentation was employed to monitor thermal conditions of cage-free laying hens, highlighting the utility of thermal imaging in avian welfare assessment (Saeidifar, Li, Chai, et al., 2024a).

Implementing automatic action and behavior recognition systems can further refine lameness detection by identifying deviations in normal activity patterns. A robust state-action recognition model capable of learning from observations has been presented, which could be adapted for poultry behavior analysis (**soans2020sa**). Moreover, a deep convolutional framework designed specifically for analyzing broiler behavior in poultry houses was introduced, facilitating early detection of anomalies (Ehsan & Mohtavipour, 2024). Furthermore, Recent developments in CNN architectures have further underscored the potential of deep learning in image classification tasks. For instance, evolutionary CNN-based architectures with attention mechanisms (Shams et al., 2024) have demonstrated enhanced performance by adaptively focusing on salient features in complex image data. Similarly, an evolving efficient CNN-based model (Shams et al., 2023) has showcased robust classification capabilities while maintaining computational efficiency. Integrating these advanced CNN models into the feature extraction or classification stages of the pipeline may further improve the system's accuracy and robustness under variable field conditions.

To enhance the system's adaptability and robustness, expanding the training dataset to encompass a diverse range of poultry breeds and environmental conditions is essential. This approach will improve the model's generalization capabilities across various farm settings. Moreover, optimizing computational

efficiency through lightweight deep learning models or edge computing solutions can enable real-time processing on low-power hardware, making the system more accessible for large-scale poultry operations.

Finally, integrating real-time anomaly detection algorithms can provide continuous health monitoring by dynamically flagging abnormal movement patterns. Exploring cross-species insights between bird lameness and human gait anomaly detection offers valuable perspectives for developing such algorithms (Zorriassatine et al., 2024). Additionally, a multi-object tracking algorithm aimed at detecting behavioral anomalies in poultry was proposed, underscoring the importance of real-time monitoring systems in maintaining flock health (Khairunissa et al., 2023).

By pursuing these enhancements, the proposed gait scoring system can evolve into a more robust, efficient, and comprehensive tool for monitoring and improving broiler welfare in commercial farms.

4.5 Conclusion

In this study, a novel three-dimensional (3D) deep learning pipeline was presented to automate gait scoring in broiler chickens by combining synchronized RGB-Depth data, YOLOv11-based pose estimation, SAM, two MLP classifiers (frame validation and gait score prediction), and complementary image processing steps. The system used ROS Noetic to record, save, and extract necessary data, ensuring efficient and reliable data gathering and synchronization. Key steps such as frame validation, platform reorientation, and 2D-to-3D back-projection were introduced to handle common challenges like occlusions and tilt variations. Segmentation masks generated by SAM accurately isolated the chickens, while kinematic features such as velocity, acceleration, and stop time ratio were extracted in XY, XZ, and YZ planes as well as 3D space. These features, along with contextual data like bird age and copper supplementation levels, were then fed into an MLP classifier for gait score prediction.

The results demonstrated that the proposed method achieved a classification accuracy of 93.34%, with a precision of 95.56%, recall of 91.16%, and F1-score of 93.31%. These findings indicated that the approach captured subtle lameness-related differences and delivered consistent performance under various environmental conditions. By incorporating a top-down Intel RealSense LiDAR camera at an approximate system cost of $\$1,483 \pm \$127$, the pipeline offered a cost-effective and scalable solution for commercial poultry operations.

Although real-time deployment may still require hardware optimizations, the current setup showed promise for reducing labor costs and subjectivity in manual scoring. It was also noted that the model's performance could be further improved by expanding the dataset with additional breeds and by integrating other sensing modalities like thermal imaging. Overall, the study demonstrated that combining advanced 3D vision with deep learning can significantly enhance gait assessment for broiler chickens, contributing to better poultry welfare monitoring and the growing field of precision livestock farming .

Chapter 5

Zero-Shot Perception and Spatiotemporal Transformers for Automated Gait and Footpad Score Classification

Manual assessment of broiler chicken gait and footpad condition is subjective, labor-intensive, and inefficient for large-scale welfare monitoring. This study introduces an automated pipeline using RGB-D video data captured by an Intel RealSense L515 camera as chickens traverse a platform. It simultaneously predicts gait quality and footpad condition using quantized scores (0–2). The proposed pipeline uses multiple zero-shot models, including YOLOE for detection, SAM2 for segmentation/tracking, and RAFT for optical flow. Decoupled X-, Y-, and Z-axis motion streams feed a dual-scale Transformer-based classifier (CNN-TimeSformer) with adaptive gating for optimal feature fusion. The pipeline achieves promising accuracy (gait: 88.9%, footpad: 81.5%), demonstrating potential for objective and comprehensive poultry welfare assessment.

5.1 Introduction

Broiler chicken production represents one of the largest and most rapidly expanding sectors within the global livestock industry (Canton, 2021) because of affordable, nutritious animal proteins for the growing human population. However, the intensive conditions associated with modern commercial poultry farming, including high stocking densities and rapid growth rates, have raised significant concerns regarding animal welfare, health, and well-being (K. Liu et al., 2023; Shynkaruk et al., 2023). Among the most prevalent issues are impaired locomotion, indicating lameness status, and poor footpad health, suggesting paw quality, which can cause pain, reduce mobility, and negatively impact productivity (Knowles et al., 2008). Monitoring these conditions is crucial not only for ethical considerations and maintaining public trust in high-quality poultry products but also for minimizing economic losses. Gait scoring and footpad

assessment are standard methods used to evaluate these aspects of broiler welfare, providing indicators of the birds' overall health and living conditions (Michel et al., 2012; Webster et al., 2008). Traditionally, gait and footpad assessments rely heavily on manual scoring by trained observers, often using standardized scales like the three-point gait scoring system widely adopted in the United States (Webster et al., 2008), with lower scores indicating better conditions. While providing a baseline, manual methods suffer from significant drawbacks that limit their effectiveness in contemporary farming contexts (D. Pereira, Nääs, & Lima, 2021). These methods are inherently labor-intensive and time-consuming, making frequent assessments impractical, especially considering the vast number of birds (often tens of thousands) housed in commercial facilities (van der Sluis et al., 2021b). Furthermore, manual scoring is subjective and prone to observer bias, leading to inconsistent results across different evaluators and even for the same evaluator at different times (Garner et al., 2002; Wurtz & Riber, 2024). The challenges associated with manual assessment are compounded by predicted labor shortages in agricultural jobs and logistical difficulties, such as restricted farm access during events like pandemics, highlighting the urgent need for more efficient and reliable alternatives (Malik et al., 2024).

The integration of automated monitoring technologies, particularly computer vision systems, offers a promising path forward (Okinda et al., 2020). Automated systems provide numerous benefits, including objectivity, consistency, the potential for continuous monitoring, enhanced scalability, reduced reliance on manual labor, and the possibility of earlier detection of welfare issues (Rohan et al., 2024). However, many existing automated systems face limitations. Early or simpler 2D vision-based systems often fail to capture critical three-dimensional spatial details, such as subtle changes in step height or lateral body oscillation, which are vital for accurate lameness assessment (Kang et al., 2021). Implementing robust 3D systems can be challenging due to difficulties in scaling systems for large, dynamic farm environments and potential high costs associated with specialized hardware or complex setups. Moreover, welfare assessment is often multifaceted; systems are needed that can evaluate multiple indicators simultaneously, such as both gait and footpad health, which are frequently interconnected. Critically, understanding animal movement requires analyzing not just static poses, but also dynamics over time. This necessitates advanced spatiotemporal analysis methods capable of interpreting complex motion patterns from sequences of images or sensor data. Recent advancements in deep learning, including sophisticated optical flow techniques like RAFT (Teed & Deng, 2020) and the emergence of powerful, adaptable zero-shot or foundation models, present new opportunities to overcome these previous limitations and develop more robust, data-efficient, and comprehensive automated monitoring solutions.

This paper introduces a novel fully automated pipeline designed specifically for the simultaneous multi-task classification of both gait quality and footpad condition in broiler chickens, utilizing RGB-D video data captured by an Intel RealSense L515 sensor. Our approach leverages several state-of-the-art zero-shot models for key processing steps, including YOLOE for multimodality-based bird detection and SAM2 for zero-shot segmentation and tracking, thereby enhancing robustness and reducing the dependency on task-specific annotated data. A core innovation lies in the generation of a novel 3D motion representation based on decoupled optical flow fields, where planar (XY) and depth (Z) motions are computed separately using the RAFT algorithm. These rich motion features, along with relevant metadata, are fed into a unique

dual-scale Transformer-based spatiotemporal feature extractor (CNN-TimeSformer). This architecture employs an adaptive gating mechanism to dynamically weigh contributions of the XY and Z motion streams before final classification. The pipeline also includes automated platform re-orientation for a consistent perspective and an outlier detection and repair module to improve data integrity. Figure 5.1 demonstrates an overview of the proposed method. The primary contributions of this work can be summarized as follows:

1. Developing a novel pipeline for simultaneous multi-task classification of broiler chicken gait and footpad scores using RGB-D video.
2. Introducing a novel 3D motion representation utilizing decoupled XY (planar) and Z (depth) RAFT optical flow streams derived from RGB-D data.
3. Designing a Transformer-based architecture (CNN-TimeSformer) with an adaptive gating mechanism for effective spatiotemporal feature extraction and adaptive fusion directly from optical flow fields.
4. Integrating state-of-the-art zero-shot models (YOLOE, SAM2, RAFT) for robust and data-efficient poultry analysis within an end-to-end system.

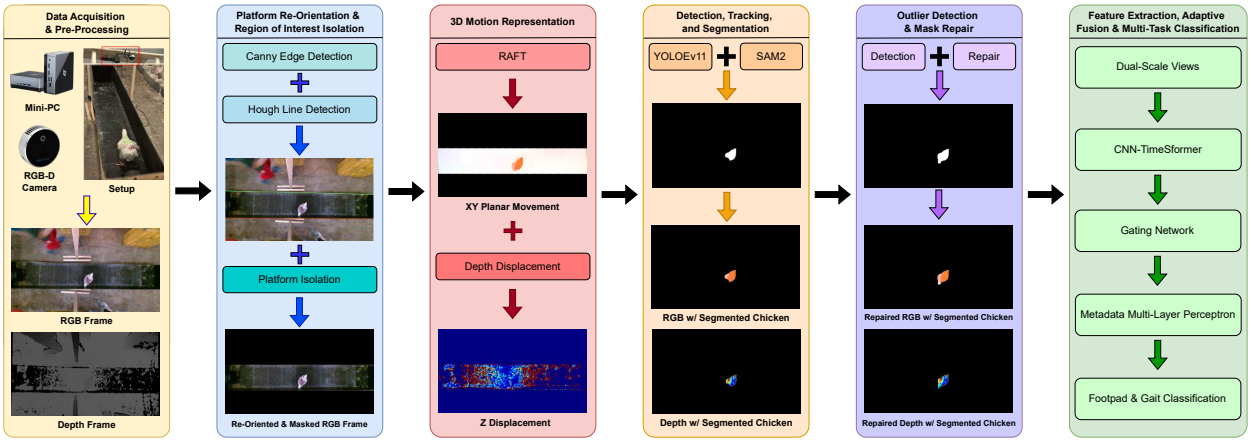

Figure 5.1: An overview of the proposed method demonstrating the data acquisition, data processing, feature extraction, adaptive fusion, and the multi-task classification processes.

5.2 Related Work

Automated computer vision systems enhance poultry welfare assessment by providing objectivity and efficiency over manual methods, particularly in large-scale operations (van Erp-van der Kooij & Rutter, 2020). Unlike Single-Task Learning (STL), which trains separate models for indicators like gait scoring (Aydin et al., 2010), Multi-Task Learning (MTL) trains a single model for concurrent tasks, leveraging

shared representations for improved data efficiency and generalization (Alirezaei et al., 2023; Pramanik et al., 2020; Ruder, 2017). Despite risks of negative transfer, MTL is increasingly applied in animal attribute classification (Kim et al., 2019; Liao et al., 2021). Our pipeline advances this trend by simultaneously classifying gait and footpad dermatitis scores from RGB-D data.

Gait analysis is vital for assessing locomotor health. While 2D vision struggles with viewpoint and occlusion, 3D methods using RGB-D sensors provide robust, marker-less data capture. Effective 3D motion representation is critical, with options including kinematic parameters or dense motion fields like optical flow. High-quality optical flow, estimated via RAFT (Teed & Deng, 2020), enables direct learning from motion patterns in 2D space. Our work introduces decoupled XY (planar) and Z (depth) optical flow streams using RAFT, offering a rich yet manageable representation of 3D motion dynamics compared to complex alternatives (Z. Y. Wang et al., 2025).

Spatiotemporal modeling has evolved from CNN-based methods (Carreira & Zisserman, 2017; Simonyan & Zisserman, 2014a; Tran et al., 2015; Zheng & Blasch, 2023) to Transformer architectures like TimeSformer, which capture long-range temporal dependencies essential for complex behaviors (Arnab et al., 2021; Bertasius et al., 2021; Gritsenko et al., 2024; Zhong et al., 2025). These models increasingly utilize optical flow as input (Ferede & Balasubramanian, 2023; Huang et al., 2022; Lin et al., 2022), with adaptive fusion techniques optimizing multi-stream integration (Asali et al., 2021). Our CNN-TimeSformer architecture processes decoupled flow inputs with an adaptive gating mechanism (Asali et al., 2023) for effective fusion and welfare score prediction, extending beyond general behavior analysis (Bodempudi et al., 2025; G. Li, Huang, et al., 2021; Merenda et al., 2024; Oso et al., 2025).

Deep learning often requires extensive labeled data (Asali & Doshi, 2024; Asali et al., 2023; Carraro et al., 2023), but Zero-Shot Learning (ZSL) (Xian et al., 2018) and foundation models like SAM/SAM2 offer solutions with robust segmentation capabilities for agriculture (Kirillov et al., 2023; Ravi et al., 2024; Saeidifar, Li, Chai, et al., 2024b). Combining detectors like YOLOE with SAM2 enhances perception in complex scenes (Kim et al., 2019; Liao et al., 2021; Saeidifar, Li, Lu, et al., 2024). Our pipeline integrates YOLOE for detection, SAM2 for segmentation, and RAFT for flow estimation, creating a data-efficient perception front-end that complements specific assessment tasks like footpad scoring.

In summary, trends in MTL (van Erp-van der Kooij & Rutter, 2020), optical flow representations (Teed & Deng, 2020), Transformer-based modeling (Bertasius et al., 2021), and foundation models (Kirillov et al., 2023; Ravi et al., 2024) highlight progress in poultry welfare assessment. Our pipeline addresses gaps in simultaneous gait and footpad scoring by combining MTL, decoupled 3D flow representations, an adapted CNN-TimeSformer with adaptive fusion, and integrated foundation models, advancing comprehensive automated welfare assessment.

5.3 Materials and Methods

This section details the experimental setup, data acquisition procedures, and the sequential processing steps involved in the proposed automated pipeline for multi-task footpad and gait score classification. The methodology encompasses initial data collection and pre-processing, followed by motion feature extrac-

tion, object detection and segmentation, outlier handling, and finally, spatiotemporal feature learning and classification.

5.3.1 Data Acquisition and Pre-Processing

The foundation of this study relies on acquiring synchronized color (RGB) and depth data from broiler chickens during locomotion. This section dives into the details of the data collection and pre-processing procedures.

System Setup: Data was collected using a system centered around an Intel RealSense L515 camera, which utilizes LiDAR technology for depth sensing. This camera offers depth resolutions up to 1024×768 and RGB resolutions up to 1920×1080, operating at up to 30 FPS. The camera was mounted on a wooden tower structure, positioned approximately 2.5 meters directly above a wooden walking platform (50 cm wide × 310 cm long × 40 cm high) to provide a consistent top-down view of the birds. Data recording and system control were managed using a lightweight mini-PC housed in a protective enclosure, connected to the camera via a high-speed USB-3 cable.

Experimental Protocol: Broiler chickens (Cobb 500 breed) were assessed for mobility during the later growth stages (weeks 6-7, when mobility issues are often more pronounced). Individual birds were placed at one end of the wooden platform and allowed to walk towards the other end while being recorded. All procedures involving animals were conducted in accordance with protocols approved by the Institutional Animal Care and Use Committee (IACUC). While manual scores for footpad and gait conditions were assigned by trained observers, the primary data for the automated system consisted of the recorded videos.

Data Recording: The Robot Operating System (ROS) Noetic framework was utilized to manage the data streams from the RealSense camera. For each chicken's walk, ROS recorded the RGB video feed, the depth map sequence, and the camera's intrinsic parameters (essential for 3D reconstruction) into a .bag file for offline processing.

Frame Extraction and Synchronization: The first pre-processing step involved extracting the raw data from the bag files. Since the RGB and depth sensors of the L515 camera operated at different frame rates (RGB at ~30 FPS, Depth at ~15 FPS), a temporal synchronization procedure was applied. Each depth frame was aligned with its nearest corresponding RGB frame in time (typically the nearest preceding one to avoid using future information). Unsynchronized RGB frames were discarded, resulting in perfectly aligned pairs of RGB and depth frames at the lower frame rate (~15 FPS). The synchronized RGB frames were saved in a standard image format (JPG), while the corresponding depth frames, containing crucial spatial information, were saved as NumPy arrays (.npy file) to preserve precision for subsequent calculations. This synchronized dataset formed the input for the next stages of the processing pipeline.

5.3.2 Zero-Shot Gait Platform Re-Orientation and Isolation

To standardize spatial analysis across recording sessions, variations in the walking platform's orientation relative to the camera are corrected using classical computer vision techniques without task-specific training. Implemented in Python with OpenCV, NumPy, and SciPy, this pre-processing step automatically detects platform edges and rotates frames for consistent perspective.

The re-orientation algorithm processes synchronized RGB frames. Each frame is converted to grayscale, and Canny edge detection (Canny, 1986) (thresholds: low=15, high=220) identifies edges. The Probabilistic Hough Line Transform (Kiryati et al., 1991) detects line segments with optimized parameters ($\rho \approx 1$ pixel, $\theta \approx \frac{\pi}{180}$ radians, accumulator threshold ≈ 100, minimum line length ≈ 50 pixels, maximum gap ≈ 20 pixels). Platform edges are filtered by length, orientation, and position. The mode slope across all frames is computed to estimate platform tilt, determining the correction angle (θ). A 2D affine rotation matrix is applied to align RGB and depth frames horizontally.

After rotation, the walkway is isolated by masking. The region of interest, defined by average platform boundaries, is used to create a binary mask via OpenCV's polygon filling. This mask is applied to rotated frames, blacking out visual data outside the platform, resulting in a sequence of consistently oriented, isolated frames ready for analysis.

5.3.3 Zero-Shot RAFT-3D

To effectively capture the complex three-dimensional locomotion patterns of the chickens, this pipeline generates separate representations for movement within the horizontal plane (XY) and movement along the vertical axis (Z). This decoupled approach forms the core of our novel 3D motion representation, providing distinct inputs for the subsequent deep learning model. The motion estimation relies on the zero-shot capabilities of pre-trained optical flow models and direct processing of depth data, implemented using Python with libraries like PyTorch, NumPy, and OpenCV.

Planar Motion (XY Optical Flow): The estimation of motion parallel to the ground (XY plane) utilizes the RAFT (Recurrent All-Pairs Field Transforms) architecture. RAFT is a deep learning model that estimates a dense optical flow field $f = (u, v)$ mapping pixels from an input image I_t to their corresponding locations in the subsequent image I_{t+1}. RAFT operates by:

1. Extracting per-pixel features from both input images, I_t and I_{t+1}.
2. Building a 4D correlation volume containing the visual similarities for all pairs of pixels.
3. Iteratively updating an initial flow-field estimate (starting from $f_0 = (0, 0)$) using a recurrent GRU-based network (J. Chung et al., 2014) that looks up values in the correlation volume based on the current flow estimate. The update process can be represented conceptually as:

$$f_{k+1} = f_k + \Delta f_k \tag{5.1}$$

where f_k is the flow estimate at iteration k, and Δf_k is the update predicted by the recurrent network based on correlation features and context.

A pre-trained RAFT model (e.g., `raft-things`) is applied to consecutive pairs of reoriented RGB frames $(I_{\mathrm{RGB},t},\ I_{\mathrm{RGB},t+1})$ without fine-tuning. The final output after K iterations,

$$f_{XY} = f_K = (u_{XY},\ v_{XY}),$$

represents the estimated pixel displacement in the XY plane. The implementation uses GPU acceleration via `PyTorch` for efficient inference.

Depth Motion (Z Displacement): Motion along the vertical axis (Z) is derived directly from the sequence of synchronized, reoriented, and isolated depth maps, $D_t\,(x, y)$, where (x, y) denotes pixel coordinates. The process involves:

1. Calculating the pixel-wise depth difference between consecutive frames:

$$\Delta D(x, y) = D_{t+1}(x, y)\ -\ D_t(x, y) \tag{2}$$

2. Computing the absolute difference to get the magnitude of depth change:

$$\Delta D_{\mathrm{abs}}(x, y) = \big|\Delta D(x, y)\big| \tag{3}$$

3. Optionally, clipping the absolute difference at a maximum threshold $\tau_{\max}$ to handle potential sensor noise or extreme values:

$$\Delta D_{\mathrm{abs,clipped}}(x, y) = \min\big(\Delta D_{\mathrm{abs}}(x, y),\ \tau_{\max}\big) \tag{4}$$

   If no threshold is applied $\big(\tau_{\max} = 0\big)$, then

$$\Delta D_{\mathrm{abs,clipped}} = \Delta D_{\mathrm{abs}}.$$

4. Normalizing the result to a $[0, 1]$ range to represent Z-motion intensity. The normalization factor N is typically the maximum value within the clipped map across the frame or the threshold $\tau_{\max}$ (if $\tau_{\max} > 0$):

$$M_z(x, y) = \frac{\Delta D_{\mathrm{abs,clipped}}(x, y)}{N} \tag{5}$$

   where

$$N = \max\big(\max_{(x,y)} \Delta D_{\mathrm{abs,clipped}}(x, y),\ \epsilon\big) \quad \text{or} \quad N = \max\big(\tau_{\max},\ \epsilon\big),$$

   and ϵ is a small constant to avoid division by zero. The resulting map M_z represents the normalized intensity of motion along the Z-axis.

Figure 5.2 depicts some examples of planar and depth motion maps for low, medium, and high intensity chicken movements masked with chickens. These two distinct streams, the dense XY optical flow fields f_XY from RAFT and the Z displacement intensity maps M_z from depth differences, constitute the decoupled 3D motion representation fed into the subsequent spatiotemporal feature extraction stage.

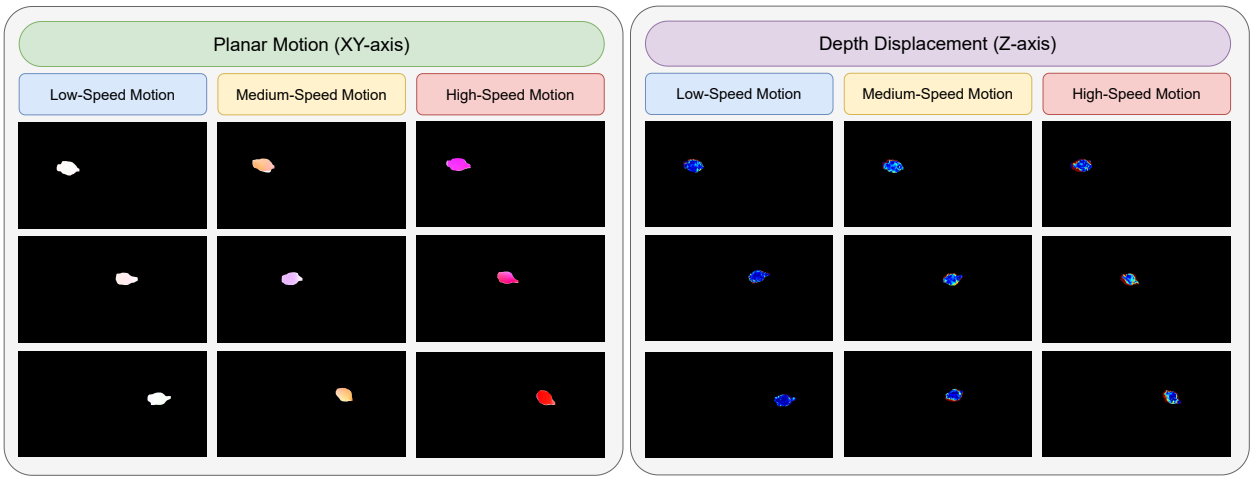

Figure 5.2: Instances of planar motion and depth displacement optical flow frames generated by our zero-shot RAFT 3D module and categorized in three different motion speeds with the chicken masked in each frame.

5.3.4 Zero-Shot Detection, Tracking, and Segmentation

Accurate localization and segmentation of the target chicken within each frame are crucial for isolating relevant motion features and preventing background clutter from interfering with the analysis. This stage employs state-of-the-art, pre-trained zero-shot models for detection, tracking, and segmentation, minimizing the need for extensive manual annotation specific to this task. The process leverages the YOLOE object detection model and the Segment Anything Model v2 (SAM2), implemented in Python using the Ultralytics library framework alongside OpenCV and NumPy.

The process begins by identifying the chicken in the initial frames of the reoriented and masked RGB video sequence. The YOLOE object detection model uses pre-trained YOLOEv11l (**l** stands for the **large** model) weights, provided with the processed frame and the text prompt *"white chicken or white bird"*, to find the first frame where a chicken is detected with sufficient confidence (e.g., confidence threshold ≈ 0.5). This initial detection provides the starting bounding box and centroid for tracking. The chicken detection and segmentation processes for the RGB, planar motion, and depth motion frames are depicted in Figure 5.3. For subsequent frames, a combined detection-tracking-segmentation approach is used.

Re-detection Attempt: YOLOE is first applied to the current frame, attempting to re-detect the bird near its previously known location, possibly using adjusted confidence and Intersection over Union (IoU)

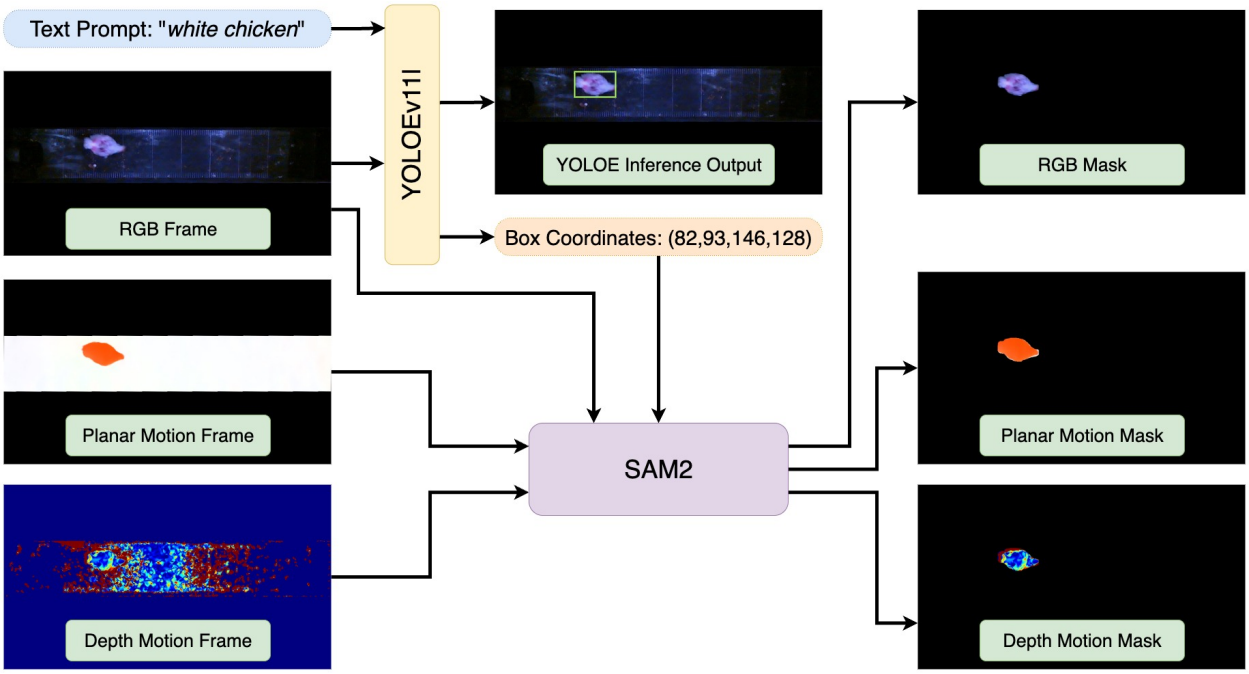

Figure 5.3: Depiction of the chicken detection and segmentation processes using YOLOEv11l and SAM2 models resulting in overlay mask frames for the RGB, planar motion, and depth motion frames.

thresholds (e.g., conf ≈ 0.4, IoU ≈ 0.4) for tracking continuity.

Segmentation with SAM2: The Segment Anything Model v2 with small architecture size is then utilized to generate a precise segmentation mask. The prompting strategy for SAM2 adapts based on the YOLOE outcome:

1. If YOLOE provides a reliable detection in the current frame (i.e., confidence score above the threshold), the center of the YOLOE bounding box is used as a point prompt for SAM2 operating in its standard prediction mode.

2. If YOLOE fails to detect the bird confidently, SAM2 is prompted using the centroid propagated from the previous frame's successful segmentation, leveraging SAM2's inherent tracking capabilities. Key parameters influencing SAM2's output include confidence thresholds for initial prediction (0.5) and tracking (0.4), and the processing image size (1024×1024).

Mask Refinement: The raw segmentation mask produced by SAM2 undergoes several post-processing steps to improve its quality and consistency. Typically, only the largest connected component (contour) within the mask is retained to eliminate small, potentially noisy detections. Morphological operations,

such as morphological opening (using a moderately large 17×17 kernel), are applied to remove small protrusions and smooth the mask boundaries. This is followed by Gaussian blurring (5×5 kernel) and re-thresholding to further refine the shape. A minimum pixel count threshold (1200 pixels in our experiments) is also enforced to discard frames where the segmentation result is deemed too small or unreliable.

The primary output of this stage is a sequence of refined, binary segmentation masks corresponding to the chicken in each valid frame. These masks are crucial for the subsequent steps, as they are applied to all relevant data streams, including the oriented RGB frames, XY optical flow fields, Z displacement maps, and depth maps, to effectively isolate the chicken's motion and appearance by setting background pixels to zero. This ensures that downstream analysis focuses solely on the target subject.

5.3.5 Zero-Shot Outlier Detection and Repair

To enhance the robustness of data fed to the classifier, an automated module identifies and repairs outlier segmentation masks, primarily by analyzing sequences of depth-derived masks and then propagating corrections to all related data modalities. This step, implemented in Python using NumPy and OpenCV, improves temporal consistency by addressing frames with noisy or incomplete segmentations. Outlier frames are identified using two main criteria:

1. **Temporal Inconsistency:** A frame is flagged if the mean absolute difference between its masked depth data and that of the preceding frame exceeds a dynamic local threshold (derived from the mean and standard deviation of changes in a temporal window, e.g., window ≈ 15 frames, using a sensitivity parameter $\alpha \approx 0.7$).

2. **Pixel Count Abnormality:** Frames with mask pixel counts falling substantially below the sequence average (e.g., 0.7 times the average) are also marked as outliers, indicating probable segmentation failure.

Identified outlier runs are then repaired. For temporal outliers where the mask is present but inconsistent, the centroid is interpolated from adjacent valid frames, and the last valid mask is translated to this position and refined. For pixel count outliers signifying a largely failed segmentation, particularly for the primary binary mask, the last valid mask is translated to the centroid of the current frame's (small) detected region. This repaired binary mask is then applied to the corresponding original rotated frames of other modalities (RGB, RAFT flows, raw depth) to ensure consistency. Post-repair, masks undergo morphological cleaning. The output includes refined data sequences with improved temporal coherence for the classification model.

5.3.6 Transformer-Based Spatiotemporal Feature Extraction

The core of the feature learning process involves extracting rich spatiotemporal representations from the decoupled XY planar motion and Z depth displacement streams. This is achieved using a Transformer-based architecture, designed to capture complex temporal dynamics and spatial patterns from the RAFT-

generated motion sequences. The implementation leverages Python and the PyTorch library, with backbone components from the timm library.

For each motion modality (XY and Z), a dual-scale approach is employed to process the input video clips. Each RAFT frame sequence is used to create:

1. A global view by resizing the full frame to a consistent dimension.
2. A local view by adaptively cropping the region around the chicken (based on its segmentation mask) and resizing the cropped area to another standard size.

This strategy yields four distinct input streams: XY-global, XY-local, Z-global, and Z-local. Each of these streams is processed by a dedicated Modality Encoder. The architecture of each Modality Encoder comprises a CNN patch encoder and a spatiotemporal attention mechanism.

CNN Patch Encoder: A convolutional neural network (CNN), such as one based on ResNet-18, serves as a feature extractor. It processes the input motion frames (2 channels, representing optical flow and depth displacement visualization) and its final feature map is projected by a 1×1 convolution to a suitable embedding dimension. This transforms image regions into a sequence of patch embeddings.

Spatiotemporal Attention Mechanism: A learnable class (CLS) token is prepended to the sequence of patch embeddings. Positional embeddings are added to these combined tokens to incorporate spatial information. This sequence is then passed through a series of TimeSformer blocks. Each TimeSformer block applies multi-head self-attention followed by a multi-layer perceptron (MLP), with layer normalization and dropout applied for regularization. This structure enables the model to learn relationships both within individual frames (effectively spatial attention on patches) and across the sequence of frames in the clip (temporal attention).

The output CLS token from the final TimeSformer block of each Modality Encoder is taken as the representative feature vector for that specific scale and modality of the input video clip. These operations are performed for each frame in the input clip, and the resulting CLS token features are then pooled across the temporal dimen-sion (by masked mean pooling, which accounts for any padding applied to shorter video clips to match a fixed processing length) to produce a fixed-size vector for each of the four streams.

5.3.7 Gating Network and Multi-Task Classifier

Following the spatiotemporal feature extraction, the fixed-size feature vectors derived from the four modality encoders (XY-global, XY-local, Z-global, Z-local) are concatenated. This aggregated feature vector is then processed by an adaptive gating mechanism. This mechanism is designed to learn the relative importance of the planar (XY) motion information versus the depth (Z) motion information for the downstream classification tasks. The gating mechanism consists of:

- A linear layer that takes the concatenated spatiotemporal features as input and outputs two logits.

- A softmax function applied to these logits produces two normalized weights, α_{XY} and α_Z (such that $\alpha_{XY} + \alpha_Z = 1$). These weights represent the dynamically learned importance of the combined XY motion features and the combined Z motion features, respectively. The design includes considerations such as bias initialization for the gate layer to set an initial preference (e.g., potentially favoring XY motion) and training aids like the injection of annealed Gaussian noise to encourage exploration in learning these weights. The behavior of the Z-modality weight (α_Z) is further guided during training by clamping it to a predefined conceptual range (e.g., ensuring it contributes meaningfully but not exclusively) and by associated regularization objectives like an alpha variance loss and a clamp penalty loss, which are part of the training loss function.

The final fused motion representation, F_{fused}, is computed as a weighted average of the features from the XY streams and the Z streams, using these learned alpha weights:

$$F_{\text{fused}} = \alpha_{XY} \operatorname{Pool}\big(F_{XY,\text{global}},\ F_{XY,\text{local}}\big) + \alpha_Z \operatorname{Pool}\big(F_{Z,\text{global}},\ F_{Z,\text{local}}\big) \tag{6}$$

where Pool represents an operation that combines the global and local features for a given modality (e.g., averaging).

Separately, tabular metadata associated with each video (including *bird age*, *copper supplementation level*, *initial and current body weights*, and geometric properties of the local crop region like normalized center coordinates and dimensions) are processed. These scalar values are passed through a dedicated MLP—typically composed of linear layers, non-linear activations (e.g., ReLU), and dropout—to generate a metadata embedding. This metadata embedding is then concatenated with the adaptively fused motion representation F_{fused}. The resulting combined feature vector serves as the input to the final classification stage, which performs simultaneous multi-task classification (Contribution 1). Two separate classification heads, each implemented as a small feed-forward network, independently predict the Footpad Score (3 classes: 0, 1, 2) and Gait Score (3 classes: 0, 1, 2). Each head outputs logits for its respective 3-class problem. Figure **??** shows the architecture of the spatiotemporal feature extraction, gating network, and the multi-task classifier in detail.

5.4 Experimental Results

This section details the quantitative evaluation of the proposed multi-task classification pipeline. We first describe the experimental setup and key training configurations, then present the classification results for both footpad and gait scoring tasks, incorporating the evaluation metrics used.

5.4.1 Training Configuration

System and Data: Experiments were conducted on a workstation equipped with an NVIDIA RTX A4500 GPU, using Python, PyTorch, timm, OpenCV, and Scikit-learn. The dataset comprised RGB-D

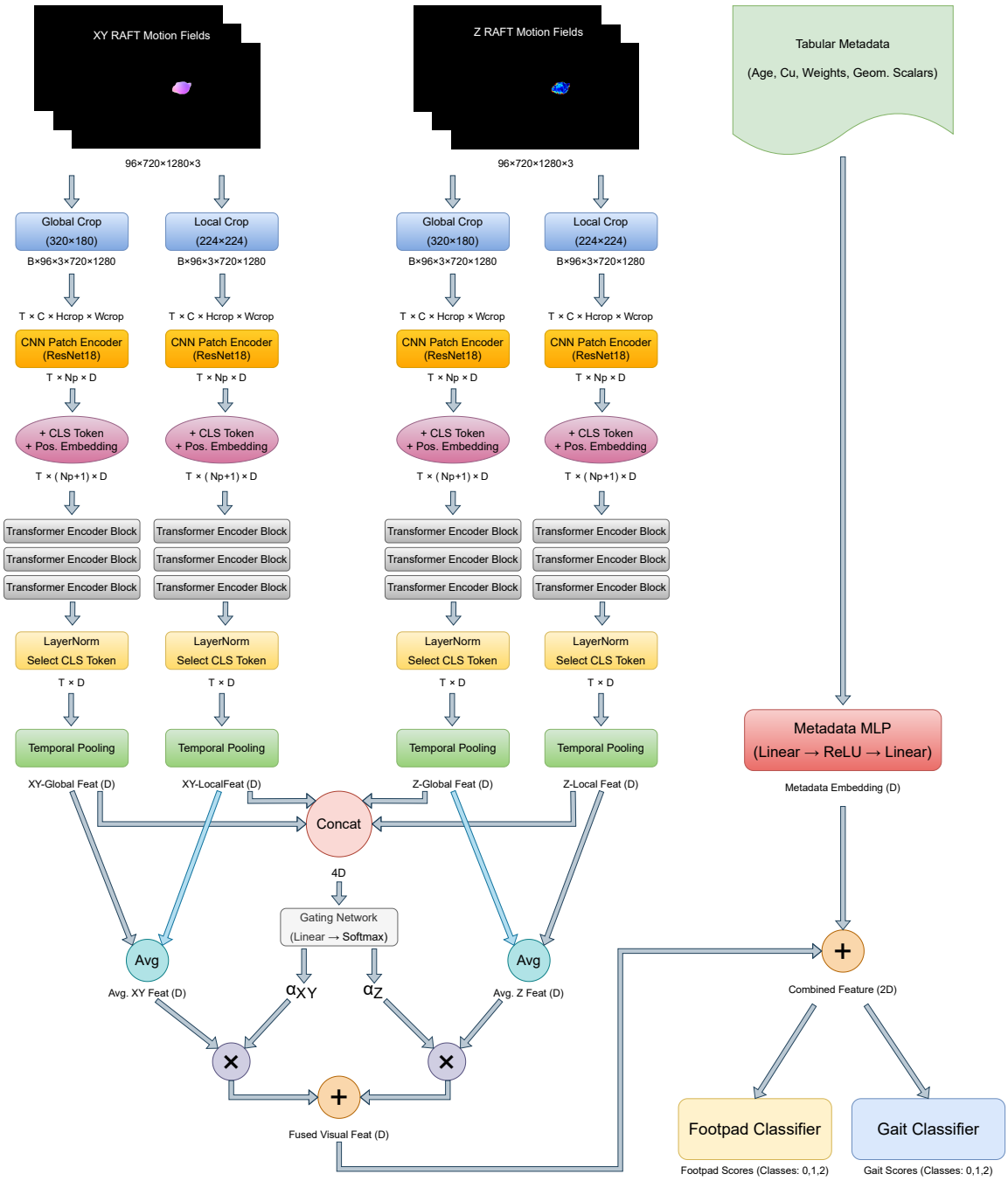

Figure 5.4: Architecture of the spatiotemporal feature extractor, gating network, and the multi-task classifier in the proposed method. Colors are used for the ease of reading and have no specific meaning. The dimensions or descriptions of the inputs and outputs of blocks are mentioned above and below them, respectively. Also, the function performed by each block is mentioned inside it.

video recordings of broiler chickens, each associated with foot-pad/gait scores and metadata. This dataset (180 videos) was divided into 70% train-ing (126 videos), 15% validation (27 videos), and 15% test (27 videos) sets using a balanced stratified splitting approach designed to ensure representation across classes, ensuring equal number of samples across different classes in each set.

Key Hyperparameters and Training: The multi-task Transformer-based model was trained end-to-end using the AdamW optimizer (Kingma & Ba, 2014; Loshchilov & Hutter, 2017) with an initial learning rate of 1×10^{-4} for most parameters and a 10x higher rate for the gating mechanism's parameters, along with a weight decay of 5×10^{-2}. A scheduler was set to monitor the average validation F1-score and adjust the learning rates during training. Training utilized a batch size of 4 with 2 gradient accumulation steps. Video clips of 96 frames were processed, with model features having an embedding dimension of 192. The composite loss function included weighted cross-entropy for each task (using inverse class frequency weights) and regularization terms for the gating mechanism, specifically an alpha variance loss (weight=1.0), a clamp penalty (weight=0.1), and temporal regularization for alpha weights (weight=0.1). Early stopping with a patience of 10 epochs based on validation F1-scores for both tasks was employed.

5.4.2 Training Results and Evaluation

The performance of the final model, selected based on the best combined validation F1-score, was assessed on the held-out test set. For evaluation, standard classification metrics were employed for both footpad and gait tasks: Accuracy, macro-averaged precision, recall, and F1-score, along with per-class F1-scores and confusion matrices. An overall average F1-score across both tasks was also used to gauge combined performance.

Test Set Performance: The classification performance on the test set is summarized in Table 5.1. The model achieved a macro F1-score of 82.0% for the footpad scoring task and 88.2% for the gait scoring task. The overall average F1-score across both tasks was 85.1%. Per-task F1-scores indicated that estimating the gait score value has more direct correspondence to the motion pattern and movement characteristics of the broiler chickens compared to the footpad score; this fact is represented by higher gait score prediction performance compared to the footpad prediction. Examination of the confusion matrices for both tasks on the test set revealed that most errors occurred between adjacent classes (score 1 vs. 2), with fewer misclassifications between extreme scores (score 0 vs. 2), suggesting the model captures ordinal relationships reasonably well.

Analysis of Modality Fusion (Gating Mechanism): The adaptive gating mechanism learned to assign an average importance of 83.5% to the XY-planar motion features and 16.5% to the Z-depth motion features on the test set. The standard deviation of these alpha weights was 11%. This indicates a clear preference for XY-planar modality, suggesting its higher relevance for the tasks.

Table 5.1: The performance of the multi-task classification on the test set.)

Task	**Accuracy**	**Precision**	**Recall**	**F1-score**
Gait Score	88.9	90.6	88.9	88.2
Footpad Score	81.5	84.7	84.7	82.0
Average	85.2	87.7	86.8	85.1

Ablation Study: An ablation study was conducted to evaluate the contributions of key components in the proposed pipeline: the XY-planar motion, the Z-depth motion, the adaptive gating mechanism, and the dual-scale feature extraction. Table 5.2 presents the performance of the full model compared to four ablated configurations on the test set, using accuracy and macro-averaged F1-score for gait and footpad scoring tasks. The full model, integrating all components, achieved the highest performance for both tasks. Removing the Z-depth motion input led to a notable performance drop, with greater degradation in footpad scoring than gait, demonstrating that depth information is critical for capturing vertical movements associated with uneven walking patterns in chickens with footpad issues. The highest degradation occurred when removing the XY-planar motion, as it captures the primary locomotion patterns essential to both gait and footpad assessments, making its absence more detrimental than the loss of depth information.

Table 5.2: Ablation study results on the test set.

Ablation Condition	**Gait Accuracy (%)**	**Gait F1-score (%)**	**Footpad Accuracy (%)**	**Footpad F1-score (%)**
Full Model	88.9	88.2	81.5	82.0
No Z-Depth Motion	77.8	77.3	70.4	70.8
No XY-Planar Motion	69.3	68.8	65.6	66.0
No Adaptive Gating	81.5	81.0	77.8	78.3
No Dual-Scale	81.5	81.0	74.1	74.5

Disabling the adaptive gating mechanism, which fixes equal weights of 50% XY and 50% Z instead of dynamically learned weights, resulted in a larger performance drop for gait than for footpad scoring. It could be due to the fact that the optimal model utilizes 83.5% contribution from XY-planar motion data and 16.5% contribution from the Z-depth motion, and when the gating mechanism is disabled, the influence of Z-depth motion becomes higher than the optimal model. Since the Z-depth motion contributes more to footpad assessment by capturing vertical motion irregularities, while gait relies more heavily on planar motion, the equal weighting dilutes the dominance of XY-planar motion. Omitting

dual-scale feature extraction by using only global features caused a moderate decline, as localized motion analysis enhances the model's ability to focus on fine-grained spatiotemporal details critical for precise classification. These results highlight the synergistic importance of decoupled 3D motion representation, adaptive gating for task-specific feature fusion, and dual-scale feature extraction in achieving robust and accurate broiler chicken welfare assessment.

The overall results underscore the capability of the integrated pipeline, combining zero-shot perception with the Transformer-based multi-task classifier to provide a robust framework for assessing multiple broiler chicken welfare indicators.

5.5 Challenges and Future Work

Despite its capabilities, the proposed pipeline presents certain challenges. The computational demands of components like RAFT optical flow and the Transformer classifier necessitate significant processing power; based on development using an NVIDIA A4500 GPU, dedicated hardware is likely required for practical execution times, limiting immediate edge deployment. The system's accuracy is also sensitive to environ-mental variations; factors such as inconsistent lighting, reflective surfaces, or back-ground clutter can potentially degrade the performance of multiple stages, including depth sensing (Intel L515), object detection and tracking (YOLOE/SAM2), segmentation (SAM2), and optical flow generation (RAFT). While zero-shot models offer adaptability, their generalization performance across substantially different breeds, farm environments, or unforeseen conditions needs further validation. Finally, ensuring optimal balance in the multi-task learning objectives and the initial technical expertise needed for system setup and calibration represents practical deployment considerations.

Future work should focus on addressing these limitations and extending the system's utility. Key directions include enhancing computational efficiency through model optimization techniques (e.g., lightweight architectures, pruning, quantization) and exploring edge computing solutions to facilitate real-time, on-farm applications. Improving robustness to environmental variability and occlusions could be achieved by expanding the training dataset with more diverse conditions and investigating sensor fusion approaches, such as utilizing multi-camera setups or integrating complementary modalities like thermal imaging. Further refinement of the outlier detection and repair mechanism could also bolster data integrity. Exploring other spatiotemporal architectures or integrating real-time anomaly detection capabilities could lead to more nuanced behavioral insights and proactive welfare monitoring beyond discrete scoring categories.

5.6 Conclusion

This study presents a robust automated pipeline that significantly enhances the efficiency, objectivity, and scalability of poultry welfare assessment by simultaneously classifying gait and footpad conditions using RGB-D video data. Leveraging advanced zero-shot foundation models and a novel decoupled 3D optical

flow representation, the system effectively captures critical spatiotemporal dynamics of chicken locomotion. The CNN-TimeSformer architecture, incorporating an adaptive gating mechanism, demonstrates effective feature fusion between planar (XY) and depth (Z) motion streams, achieving notable accuracy (88.9% for gait scoring and 81.5% for footpad scoring). The ablation study confirms the critical roles of the decoupled 3D motion representation, adaptive gating mechanism, and dual-scale feature extraction, with the XY planar motion being the most influential for achieving high performance in both gait and footpad scoring tasks. While computational demands and environmental sensitivity pose practical deployment challenges, the proposed framework establishes a strong foundation for future developments, including model optimization, enhanced robustness, and broader adaptability across varying operational environments. Ultimately, this pipeline represents a significant advancement toward automated, comprehensive, and reliable poultry welfare monitoring.

Chapter 6

Comparative Evaluation of Mask-Based Multimodal Action Recognition Pipelines for Broiler Chickens Using Early and Late Fusion Strategies

This chapter presents a comprehensive comparative evaluation of multiple proposed mask-based multimodal action recognition pipelines for broiler chickens, integrating high-resolution RGB-D video, advanced object detection, zero-shot segmentation, and state-of-the-art deep learning models. Leveraging a robust data acquisition system deployed in commercial broiler pens, the study systematically benchmarks five object detection architectures and a suite of segmentation models, identifying YOLOv11 and Mobile-SAM as optimal for downstream tasks based on their balance of accuracy and inference speed. Four distinct action recognition architectures, including Video Swin Transformer, TimeSformer, X3D, and CNN+LSTM, were assessed using six modality configurations, with early and late fusion strategies directly compared for X3D. Results demonstrate that late fusion of RGB and depth modalities with X3D yields the highest action recognition performance, achieving 88.13% accuracy and 87.95% F1-score, while maintaining rapid inference suitable for large-scale analysis. The pipeline's end-to-end automation enables detailed behavioral analyses, revealing natural activity patterns, temporal and spatial trends, and their correlations with gait, footpad health, and body weight. The chapter discusses key technical challenges, practical limitations, and deployment considerations, as well as broader applications for welfare monitoring and data-driven management in precision livestock farming. Overall, the findings highlight the transformative potential of scalable, multimodal, and mask-based deep learning frameworks for automated animal behavior recognition in complex, real-world environments.

6.1 Introduction

Automated behavior monitoring has become increasingly vital in modern poultry production, offering the potential to improve animal welfare, productivity, and management efficiency through objective, scalable, and real-time analytics. Recognizing and quantifying specific actions of broiler chickens in commercial environments presents a unique set of challenges, including high animal density, dynamic interactions, and visually complex settings with numerous confounding factors such as feeders and drinkers. Recent advances in computer vision, particularly in action recognition, object detection, and multimodal data fusion, provide promising tools to address these challenges (Oliveira et al., 2021). This chapter introduces a comprehensive, context-aware action recognition pipeline tailored to the needs and complexities of large-scale broiler production, establishing a new benchmark for evaluating multimodal methods and behavioral analysis in precision livestock farming.

6.1.1 Motivation & Background

The rapid advancement of precision livestock farming has brought automated animal behavior analysis to the forefront of modern poultry production (Norton et al., 2019). Reliable recognition of chicken actions, such as feeding, drinking, moving, and resting, not only enables real-time monitoring of flock welfare but also facilitates data-driven management strategies that can enhance productivity, sustainability, and animal well-being. Action recognition provides producers and researchers with objective metrics to evaluate the impact of dietary interventions, housing modifications, or environmental changes, enabling the timely detection of health issues such as lameness, dehydration, or abnormal inactivity.

Broiler chickens, which constitute the world's most widely produced meat animal, present both a highly relevant and uniquely challenging subject for automated action recognition. Their fast growth rates and high stocking densities, combined with rapid, context-dependent behavioral shifts, create complex and crowded scenes that challenge both classical and contemporary vision algorithms. Unlike many computer vision benchmarks that assume relatively sparse and isolated subjects, real-world broiler pens are characterized by dense groupings of similarly appearing individuals, frequent occlusions, and dynamic interactions with contextual elements such as feeders and drinker lines (N. Li et al., 2019). These factors make accurate behavior analysis in commercial settings far more demanding than in laboratory or single-animal studies.

Moreover, context dependency is a defining feature of broiler chicken behavior. Actions such as feeding and drinking are only meaningfully interpreted in relation to fixed resources, feeder bowls, and water lines, whose presence, position, and appearance may vary across pens or over time. Differentiating between, for instance, a chicken resting near a drinker and one actively drinking requires models to integrate not only the visual appearance and motion of the animal but also the spatial and semantic context of its surroundings. This interdependence of object detection, segmentation, and context-aware recognition necessitates sophisticated pipelines capable of multi-modal fusion, fine-grained spatial reasoning, and robust generalization to the variability inherent in commercial poultry production.

6.1.2 Related Work

This subsection reviews foundational and recent advancements in computer vision techniques relevant to action recognition, multi-modal fusion, object detection, and behavior analysis, with a particular focus on their applications in dense and dynamic environments such as poultry farming. The discussion spans spatiotemporal architectures for video analysis, fusion strategies for integrating diverse data modalities, detection and segmentation methods for crowded scenes, and vision-based approaches for animal behavior monitoring. Emphasis is placed on how these methods inform the development of robust systems for welfare-driven applications, highlighting gaps that this dissertation aims to address.

Vision-Based Spatiotemporal Architectures for Action Recognition: Over the past decade, video action recognition has evolved significantly, driven by advances in spatiotemporal architectures. Early work introduced two-stream convolutional networks—one processing RGB frames and another processing optical flow—to capture both appearance and motion cues (Simonyan & Zisserman, 2014a). Subsequent research combined convolutional spatial feature extractors with recurrent temporal models, enhancing frame semantics via LSTM units to model temporal dependencies in Long-term Recurrent Convolutional Networks (LRCNs) (Donahue et al., 2015).

Later advancements introduced 3D convolutional neural networks (3D CNNs), such as C3D, to jointly learn spatial and temporal features directly from raw video volumes (Tran et al., 2015). Further progress involved inflating high-performing 2D image encoders into 3D spatiotemporal models, as seen in I3D, which achieved state-of-the-art performance on the Kinetics dataset (Carreira & Zisserman, 2017). Research continued to explore deeper 3D CNN variants (Hara et al., 2018) and architectures like SlowFast networks that process video at different temporal speeds (Feichtenhofer et al., 2019).

Efficiency-focused approaches emerged subsequently, with models like X3D reducing computational cost through progressive model expansion (Feichtenhofer, 2020). Transformer-based architectures, including TimeSformer (Bertasius et al., 2021), Video Swin Transformer (Z. Liu, Ning, et al., 2022), and ViViT (Arnab et al., 2021), leveraged self-attention mechanisms to model long-range dependencies in both space and time. More recent studies explored spatio-action graph representations for robotic and interactive learning tasks, as demonstrated in SA-Net and MVSA-Net (Asali et al., 2023; Soans et al., 2020).

Multi-Modal Fusion in Computer Vision: Multi-modal fusion techniques have been pivotal in tasks where diverse data sources complement each other, such as combining RGB, depth, and spatial context for action recognition. Broadly, fusion approaches are categorized into early fusion, late fusion, and intermediate fusion. Early fusion, where modalities are concatenated at input or feature levels, enables joint feature learning but may struggle with heterogeneous data scales and modalities. Late fusion, combining modality-specific predictions at decision time, offers flexibility for differing data structures but may miss cross-modal feature interactions (Barnum et al., 2020). Intermediate or hybrid fusion methods integrate the benefits of both by enabling selective feature fusion at various network depths (Vaezi Joze et al., 2019).

A comprehensive survey provides a taxonomy of deep multimodal fusion, detailing architectures like multimodal autoencoders and attention-based modules, emphasizing early fusion's ability to capture

joint representations and late fusion's modular, scalable structure (J. Gao et al., 2020). Controlled experiments on RGB, optical flow, and skeleton modalities using I3D networks demonstrated that early fusion consistently outperformed late fusion in action recognition tasks, supporting the notion of richer joint representations (Gadzicki et al., 2020). This trend aligns with findings across various vision tasks.

Domain-specific research evaluated fusion approaches for tasks combining RGB and depth information in object recognition and semantic segmentation. A survey of RGB-D CNNs showed that the optimal fusion strategy varies across applications, with late fusion sometimes preferred for noisy or heterogeneous depth data (M. Gao et al., 2019). Another study on RGB-D semantic segmentation found that attention-based fusion improved consistency between visual and depth cues (C. Wang et al., 2021). These findings suggest tailoring fusion strategies based on modality reliability and nature.

In summary, the fusion literature indicates that early fusion often yields superior representations when modalities are well-aligned, while late fusion remains valuable for modularity. Intermediate methods, such as squeeze-and-excitation fusion modules, offer promising trade-offs (Vaezi Joze et al., 2019). However, these approaches have rarely been tested in dense, multi-object scenes like broiler chicken pens with feeder and drinker context, highlighting a gap this work aims to address.

Object Detection and Segmentation in Dense Scenes: Object detection in crowded poultry environments presents challenges due to overlapping subjects and occlusions. The YOLO family advanced real-time detection, with YOLOv3 described as "an incremental improvement" (Redmon & Farhadi, 2018), YOLOv4 optimizing speed and accuracy (Bochkovskiy et al., 2020), and YOLOv7 demonstrating high performance (C.-Y. Wang et al., 2022). The latest iteration, YOLOv8, introduced robustness improvements for complex scenes like dense animal housing (Varghese & Sambath, 2024). Transformer-based models, such as DETR (Carion et al., 2020) and RT-DETRv2 (Lv et al., 2024), enhanced end-to-end detection in occluded and cluttered frames. For segmentation, foundation models like Segment Anything (SAM) (Kirillov et al., 2023) and Mask2Former (Cheng et al., 2022) excel at zero-shot and prompt-based segmentation. CLIPSeg enables text-conditioned object delineation, useful for contextual labeling in poultry monitoring (Lüddecke & Steger, 2022). These segmentation models ensure reliable mask extraction under partial visibility.

Animal Action Recognition and Behavior Analysis: Computer vision has increasingly been applied to monitor animal behavior in farm settings, focusing on pose estimation and action recognition. Research introduced LEAP, a deep neural network for efficient animal pose estimation with minimal training data, demonstrating effectiveness on flies and mice (T. D. Pereira et al., 2019). Similarly, DeepLabCut, built upon the DeeperCut framework, became a widely used toolbox for markerless pose estimation across species (Mathis & Mathis, 2018). In livestock, studies achieved high accuracy (99%) in detecting pig behaviors like feeding, lying, walking, scratching, and mounting using a two-stream (RGB + optical flow) network based on I3D and TSN architectures (K. Zhang et al., 2020). Another study proposed a SlowFast-based spatiotemporal convolutional network, achieving 97.6% accuracy in multi-behavior pig

recognition (D. Li et al., 2020). These advancements highlight the potential of combining spatial and temporal information for robust animal behavior recognition, including welfare-driven applications.

The importance of behavior–health correlations in poultry prompted progress in vision-based methods. Research quantified nesting behaviors and correlated them with health metrics in breeder hens (Mandiga et al., 2024). Tools like KineWheel adapted DeepLabCut for rodent gait analysis, illustrating cross-species methodological transfer (Panconi et al., 2024). A review emphasized the growing use of vision systems in poultry welfare, noting challenges like occlusion and environmental variability (Okinda et al., 2020). Transformer-based models, such as pig-focused Time-Series Neural Networks (TNN), demonstrated strong performance on edge devices by modeling long-range dependencies in animal behavior (Y. Zhang et al., 2024). These efforts establish a foundation for behavior analysis in broiler systems, yet none integrate multiple models, fusion strategies, and depth–contextual fusion in a unified comparative framework as this study does.

Behavioral Correlations with Health and Environment: The relationship between animal behavior and health or environmental conditions has been extensively studied using computer vision. Research demonstrated that computationally efficient image processing methods can quantify nesting behavior metrics, such as time spent and frequency of visits in nest slots, for broiler breeder hens, offering insights for pen design and resource management (Mandiga et al., 2024). A review explored how housing factors like airflow, ammonia, lighting, and space impact chicken immune systems, linking stress-related behaviors to health outcomes (Hofmann et al., 2020). A systematic review highlighted computer vision's role in monitoring chicken behavior to assess health conditions like lameness, footpad dermatitis, and activity levels (Bhuiyan & Wree, 2024). Another survey examined coarse-to-fine-grained animal action recognition approaches, emphasizing refined behavioral detection for welfare assessment (Zia et al., 2025). Studies showed that environmental noise affects animal stress and health, suggesting automated behavioral tracking as an early warning system for stressors (Anderson, 2011). These findings underscore the importance of behavioral analysis linked to health and environment, reinforcing the value of automated, vision-based monitoring systems in poultry welfare research.

6.1.3 Objectives and Contributions

This study addresses the need for robust, scalable solutions for automated behavior monitoring in commercial poultry production by targeting action recognition in complex, multi-object pen environments (Nasiri et al., 2022). The **objectives** of this study are as follows:

- Systematically compare four action recognition models (Video Swin Transformer, TimeSformer, X3D, and CNN+LSTM) across six multi-modal data configurations for broiler chicken action recognition (Z. Liu, Mao, et al., 2022).
- Evaluate the impact of early versus late fusion strategies (for X3D) in integrating mask and depth information (Feichtenhofer, 2020).

- Benchmark state-of-the-art object detection and segmentation models for accurate localization and mask generation in dense, commercial pen environments (Zhuang et al., 2018).
- Enable large-scale, automated analysis of chicken behaviors in relation to spatial, temporal, and health-related variables.

To address these objectives, this work makes the following **key contributions**:

- Proposes multiple mask-based multimodal action recognition pipelines specifically designed for commercial broiler chicken environments.
- Provides the first systematic evaluation of four architectures and six modalities, quantifying the role of fusion strategy (in X3D) and context-aware masking (Z. Liu, Mao, et al., 2022).
- Establishes a robust, context-integrated pipeline for object detection, zero-shot segmentation, and multi-modal fusion.
- Delivers novel behavioral insights into broiler chicken activity, linking recognized actions to welfare and health indicators at commercial scale (Fang et al., 2021).

These objectives and contributions set the stage for a comprehensive exploration of automated action recognition in commercial poultry production. In the following sections, we detail the system design, experimental setup, data collection, and model development processes. We then present a thorough evaluation of detection, segmentation, and action recognition performance across all configurations, followed by downstream behavioral analyses and a critical discussion of the broader implications, challenges, and future opportunities emerging from this work.

6.2 Materials and Methods

This section describes the complete experimental framework and methodology underlying our comparative evaluation of mask-based multimodal action recognition pipelines for broiler chickens. We outline the system setup and hardware configuration, detail the dataset acquisition and annotation processes, and present the key components of our data processing pipeline, including object detection, segmentation, modality fusion strategies, and the action recognition models employed. By providing a transparent account of the technical design and experimental procedures, we ensure the reproducibility and rigor of our study and set the foundation for the results and analyses that follow.

6.2.1 System Setup and Experimental Configuration

The experimental system configuration used in this study is identical to the design previously detailed in Chapter 3 (under Section 3.2.1) of the dissertation. We provide a brief overview of the system specifications here.

An intelligent and portable data acquisition system was developed to support real-time video and depth data collection, robust data transfer, and reliable system operation in a commercial broiler house environment (Asali et al., 2025). The core of the system consisted of a mini-PC and all essential electronics, compactly integrated within a plastic junction box of dimensions 13 cm × 21 cm × 29 cm. The system was equipped with an Intel RealSense L515 LiDAR camera for three-dimensional (3D) image and depth sensing, offering a depth resolution of up to 1024×768 pixels and an RGB resolution of 1920×1080 pixels at a maximum sampling rate of 30 frames per second. The L515 camera was installed overhead using a wall mount arm to capture a comprehensive view of the pen area. A mini-PC (CyberGeek Nano J1) provided 8 GB of LPDDR4 RAM, a 256 GB SSD, and a 2 GHz Intel Celeron processor, running Ubuntu 20.04 LTS and ROS Noetic for data acquisition, device control, and on-site monitoring. Auxiliary components included an external hard drive for data backup, a touchable screen for system interaction, and various cables and adapters to support stable operation and flexible mounting. Figure 3.1 (see Chapter 3) presents an illustration of the integrated system and its deployment in the broiler house, and Table 3.1 (in chapter 3) provides further details and specifications about the monitoring system.

Following each data collection cycle, all raw data were securely stored on external hard drives. For model training and testing, a high-performance desktop computer running Windows 11 Enterprise was utilized. This system was equipped with a 13^{th} Generation Intel Core i7-13700 CPU, 64 GB of RAM, and an NVIDIA RTX A4500 GPU with 20GB of GDDR6 memory, ensuring computational efficiency for training deep learning models and processing large-scale datasets.

All experimental work was performed at the Poultry Research Center, University of Georgia, located in southern Athens, Georgia. The experiments were conducted from February to June 2024, encompassing a total of 1, 776 day-old Cobb 500 off-sex male broilers. Birds were raised in 48 pens, each pen housing an average of 37 birds and subjected to one of four copper supplementation levels (5, 125, 250, or 500ppm) and either fresh or used litter (Arias & Koutsos, 2006). Welfare monitoring involved randomly selecting 10 chickens per pen, uniquely color-marked and tracked from weeks 4 to 7. Each colored chicken was assigned a unique color code (ranging from 1 to 10) based on its marking, and individual measurements of body weight, gait score, and footpad score were performed weekly by trained personnel. All animal procedures followed the guidelines and approvals of the Institutional Animal Care and Use Committee (IACUC) at the University of Georgia (protocol number: A2023-07-016-Y1-A0).

6.2.2 Dataset

The dataset utilized in this study was collected using an Intel RealSense L515 LiDAR camera, providing both RGB and depth information for action recognition tasks. Data acquisition was conducted in three separate pens, each containing an average of 37 broiler chickens. For each pen, recordings were performed on two days: one in week 6 and one in week 7 of chicken ages. On each recording day, a 5-minute video was sampled every hour, resulting in 24 videos per day per pen, and a total of 48 five-minute videos per pen. This corresponds to approximately 4 hours of video data per pen.

During each recording, only 10 chickens per pen were targeted for action recognition, as these individuals were color-marked and had corresponding body weight, gait, and footpad score data available. In total, data from 30 uniquely marked chickens were included in the analysis.

The RGB frames were captured at an average rate of approximately 30 frames per second (FPS), while the depth frames were recorded at approximately 15 FPS. All RGB and depth frames were spatially and temporally aligned using the camera's intrinsic parameters and frame timestamps, respectively. The aligned and synchronized RGB-D frames were saved at 15 FPS to ensure consistency. For action recognition, the dataset was temporally downsampled by sampling every 5 frames, resulting in an effective frame rate of 3 FPS. The Depth Anything V2 (L. Yang et al., 2024) model was used to generate estimated depth frames given the downsampled set of cropped RGB frames as input.

Then, Non-overlapping sets of 3 consecutive frames (corresponding to 1 second of video) were grouped to form individual action samples. All videos, metadata, and annotation files were organized by pen, day, and chicken identity. Eventually, frames were cropped to only include the pen region and discard the excessive areas that the camera was able to cover.

For object detection model development, a total of 576 frames were labeled for training and evaluation. Specifically, 4 random frames were sampled from each of the 144 videos collected across the 3 pens. Using the X-Anylabeling tool, a total of 5,622 colored chickens were manually annotated within these frames (W. Wang, 2023). The object detection dataset was split into training, validation, and test sets with an 80%, 10%, and 10% ratio, respectively. Also, for evaluating the segmentation models, 50 randomly sampled frames were chosen and the 10 colored chickens were annotated by a human expert, again, using the X-Anylabeling tool. The data were later used to evaluate and compare the performance of different segmentation models.

For the action recognition task, a total of 1,265 action samples were used for training, 155 samples for validation, and 159 samples for testing. Each sample consisted of 3 consecutive frames representing approximately 1 second of activity for a single colored chicken. The action labels for all training, validation, and test samples were manually annotated by a human expert, ensuring reliable ground truth for supervised model development. After model evaluation, the trained action recognition models were used to infer the actions in the whole dataset, including 479,520 action samples, enabling a comprehensive behavior pattern analysis.

6.2.3 Pipeline Overview

The experimental pipeline for multimodal action recognition in broiler chickens was designed to process raw RGB-D video recordings and produce detailed behavior analysis for each individual subject. The complete workflow consisted of several key stages, each integrating state-of-the-art computer vision models and systematic data processing procedures. Figure 6.1 visualizes the chronological main steps of the pipeline and their processes, followed by behavioral pattern analysis of chicken actions.

‘

The pipeline began with the recording of synchronized RGB and depth data using the Intel RealSense L515 LiDAR camera. After data extraction, the RGB and depth frames were spatially and temporally

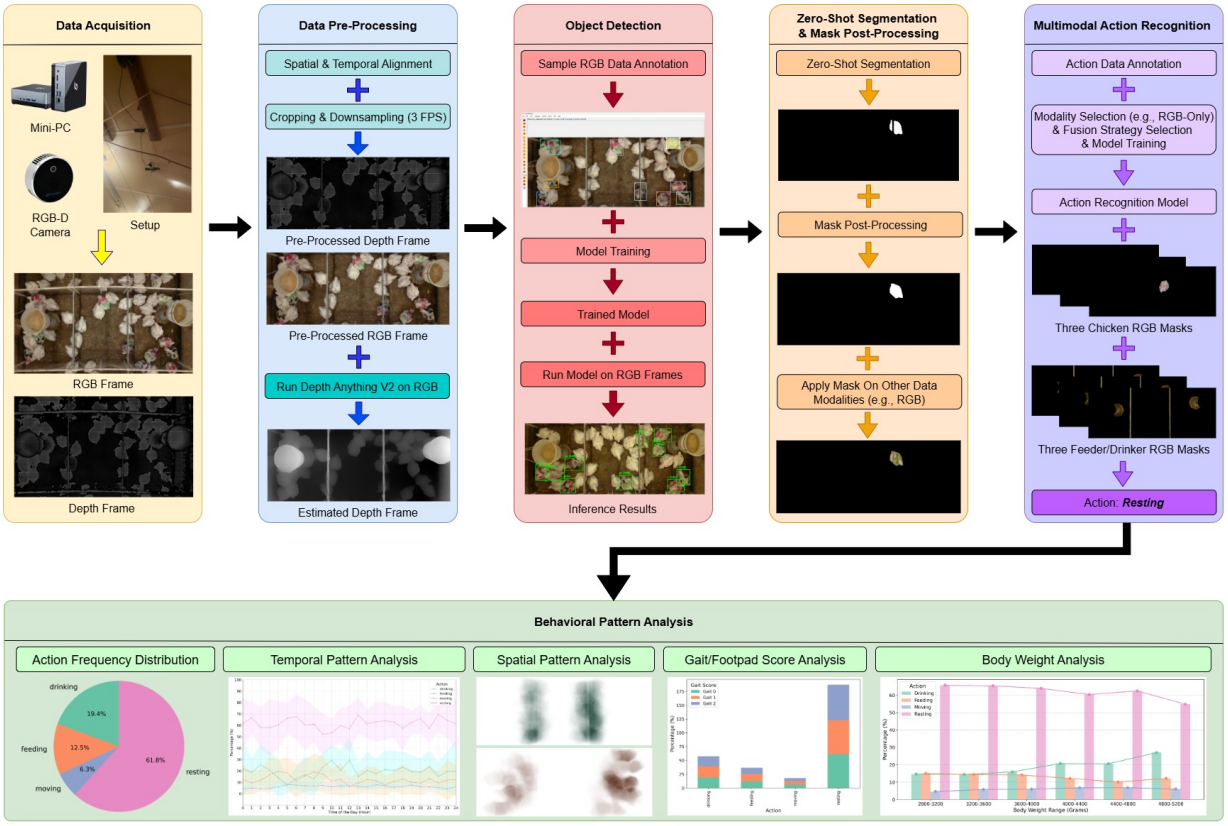

Figure 6.1: An end-to-end overview of the proposed multimodal action recognition pipeline, followed by behavioral pattern analysis of chicken actions.

aligned using the camera's intrinsic parameters and frame timestamps, resulting in well-registered RGB-D image pairs at an effective rate of 15 frames per second. To reduce redundancy and computational overhead, frames were sampled every 5 frames, yielding an effective frame rate of 3 frames per second for downstream processing. A sample of spatially and temporally aligned RGB and depth frames is presented in Figure 6.2.

Each sampled RGB frame was first passed through a custom-trained object detection model to localize all colored chickens present in the scene. Following detection, each frame and its corresponding detection results were processed by a zero-shot segmentation model to generate precise masks for each chicken, as well as the feeder and drinker lines. To further enhance mask quality, post-processing operations such as gap filling and contour smoothing were applied, minimizing the effects of noise and segmentation artifacts. With the segmentation masks available, the raw RGB, real or estimated depth, and context channels (feeder/drinker) were masked according to the selected data modality configuration. We will discuss the data modalities later in this chapter.

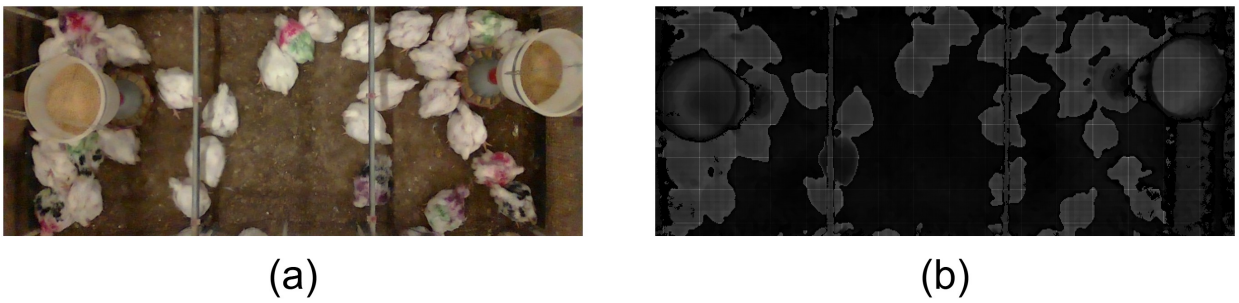

Figure 6.2: Demonstration of a sample synchronized and spatially aligned RGB and depth frame.

For action recognition, the processed dataset was segmented into non-overlapping samples, each comprising 3 consecutive frames and representing approximately 1 second of activity for a single chicken. These triplets, together with their associated modality inputs, were fed to one of four action recognition models: Video Swin Transformer, TimeSformer, X3D, or CNN+LSTM. Depending on the experiment, either early fusion (modality channels concatenated before input to the model) or late fusion (modalities processed independently before feature-level integration) was used.

The output of the action recognition models provided an action label for each sample. Using these predictions, further behavioral analyses were conducted to explore the temporal, spatial, and physiological correlates of individual and group chicken activity. These analyses included assessments of action frequencies over time, correlations with body weight, gait, and footpad scores, and visualization of action spatial distributions within the pen.

All model development, data extraction, and processing steps were implemented in Python 3.12.3, leveraging packages such as Ultralytics for detection and segmentation, and `torch`, `torchvision`, and `sklearn` for model training and evaluation (Jocher & Qiu, 2024). Additional data handling and visualization were performed using `os`, `PIL`, `cv2`, `tqdm`, `numpy`, `pandas`, `logging`, `glob`, `collections`, `matplotlib`, and `argparse`. All deep learning experiments were accelerated using CUDA version 12.8.

6.2.4 Object Detection and Segmentation

Accurate localization and segmentation of individual chickens were critical initial steps in the multimodal action recognition pipeline. To accomplish this, a combination of supervised object detection and zero-shot segmentation models was employed, followed by mask post-processing and precise object-to-mask association procedures.

Object Detection: All object detection experiments were conducted using the same labeled dataset, which comprised 576 randomly selected frames sampled from the full video collection (4 frames per video across 144 videos in 3 pens). Within these frames, a total of $5,622$ colored chickens were manually annotated using the X-Anylabeling tool. Five state-of-the-art detection architectures were evaluated:

YOLOv8-l (Jocher et al., 2023), YOLOv10-l (A. Wang et al., 2024), YOLOv11-l (Jocher & Qiu, 2024), YOLOv12-l (Tian et al., 2025), and RT-DETR (Lv et al., 2023). Each model was trained using the same hyperparameters: 100 epochs, a batch size of 16, an input image size of 640 pixels, an initial learning rate of 0.01, and a patience value of 50 for early stopping. All models were initialized with pre-trained weights from the COCO dataset and trained using the large architecture variant. The training procedure was conducted on a CUDA-enabled device.

For model evaluation, accuracy, precision, recall, F1-score, mAP, and inference time per sample were computed for each architecture. The object detection dataset was split into training, validation, and test subsets with an 80%, 10%, and 10% ratio, respectively. To ensure reproducibility, the random seed was set to 42 for all randomized processes.

Segmentation: Zero-shot segmentation models were applied to generate pixel-level masks for each detected chicken, as well as for background elements such as feeders and drinker lines. The segmentation models evaluated included all variants of SAM (Kirillov et al., 2023), SAM2 (Ravi et al., 2024), MobileSAM (C. Zhang et al., 2023), and FastSAM (Zhao et al., 2023). No fine-tuning was performed on these models. For each frame, the bounding boxes obtained from the detection step, along with the corresponding RGB frame, were fed to the segmentation model to extract the target object masks. The same procedure was used for all segmentation models.

Mask Post-processing: To enhance the quality and completeness of the segmentation masks, especially for challenging scenes with occlusions, a dedicated post-processing step was incorporated into the pipeline. The primary goal was to address common segmentation artifacts, such as noise, small disconnected regions, and gaps in chicken masks that arose from visual obstructions in the pen environment.

Mask cleaning began with binarization and denoising, removing all connected components with an area smaller than 100 pixels. In cases where the chicken body was separated into two major blobs, typically when the drinking line, which is positioned above the birds, occluded part of the body in a top-down view, the algorithm retained only the two largest blobs. A gap-filling procedure was then applied: connection lines were drawn between corresponding top and bottom points of the two blobs, and the enclosed region was filled, resulting in a continuous, unified mask for the chicken.

This gap-filling approach was specifically applied to chicken masks when the body was split by the drinking line, and was not used for other objects or for chicken instances where only a single major region was present. The post-processing pipeline proved effective for most cases; however, some edge cases persisted, such as when one of the blobs was smaller than 100 pixels or when a significant portion of the chicken body was occluded by either the drinking line or the feeder, leaving no contiguous body part across the obstruction. In such scenarios, the model was unable to fully reconstruct the chicken mask.

Examples of raw segmentation masks and their post-processed counterparts are illustrated in Figure 6.5 in the Results section of this chapter, highlighting the improvements achieved through this procedure. All mask post-processing operations were performed using Python, OpenCV (Bradski, 2000), and NumPy (Harris et al., 2020).

Feeder and Drinker Masks: The two feeders in each pen were circular bowls suspended from the ceiling, while the drinker lines consisted of two parallel pipes with multiple nipples for water access. Because the physical positions of feeders and drinkers relative to the camera remained fixed, custom binary masks for these objects were manually annotated for each pen. These static masks were later applied to the corresponding frames throughout the entire dataset whenever the feeder and drinker context was required for the input modality.

Model Selection for Downstream Tasks: Selection of the detection and segmentation models for use in the subsequent action recognition pipeline was based primarily on accuracy. In cases where models exhibited similar accuracy, preference was given to the model with substantially lower inference or training time. This approach ensured both high-quality object localization and practical processing efficiency for large-scale analysis.

6.2.5 Data Modality Configurations and Fusion Strategies

To systematically investigate the impact of multi-modal information on action recognition performance, the pipeline was evaluated with six distinct input modality configurations. Each configuration was constructed to capture varying combinations of spatial, contextual, and depth information for both the chicken and the feeder/drinker regions. The six configurations were:

1. RGB masks (chicken and feeder/drinker) only
2. Black/white masks (chicken and feeder/drinker) only
3. RGB masks (chicken and feeder/drinker) with real depth (chicken and feeder/drinker)
4. RGB masks (chicken and feeder/drinker) with estimated depth (chicken and feeder/drinker)
5. Black/white masks (chicken and feeder/drinker) with real depth (chicken and feeder/drinker)
6. Black/white masks (chicken and feeder/drinker) with estimated depth (chicken and feeder/drinker)

For each configuration, the chicken and feeder/drinker masks were provided either as single-channel black/white images or three-channel RGB images. When depth was included, both real and estimated depth channels were provided as single-channel arrays, and always paired between the chicken and feeder/drinker. All channels within a configuration were strictly matched in type, and no configuration mixed RGB and black/white masks, or real and estimated depth within the same sample. A sample of different chicken and feeder/drinker masks are demonstrated in Figure 6.3 and Figure 6.4, respectively.

Each action recognition sample comprised three consecutive frames, corresponding to approximately one second of observed activity. Prior to input to the models, all images and channels were resized to 256×256 pixels and normalized. black/white mask channels were normalized to a mean and standard deviation of 0.5, RGB mask channels used a mean of 0.45 and standard deviation of 0.225, and all depth channels were normalized to a mean and standard deviation of 0.5.

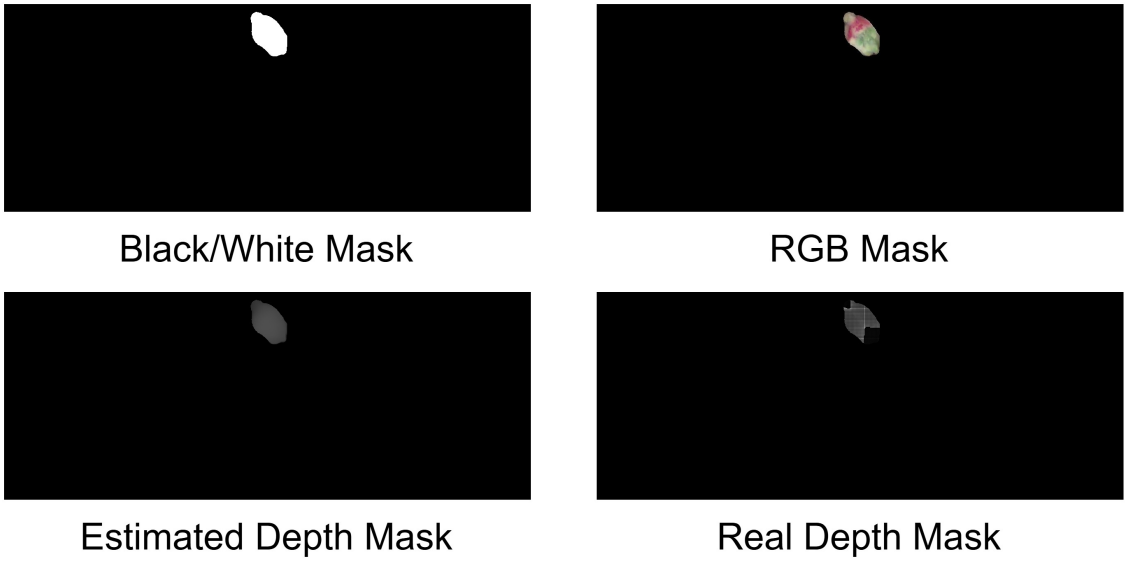

Figure 6.3: Illustration of samples for four different data modalities used in this study for chicken masks.

To further examine the impact of data integration, both early and late fusion strategies were systematically compared for all action recognition models and data modalities. In the early fusion strategy, all available channels (mask and optional depth channels for both chicken and feeder/drinker) were concatenated along the channel dimension and presented as a single input tensor to the model. In the late fusion strategy, the input channels were separated into modality-specific streams (e.g., mask stream and depth stream), each processed by a separate subnetwork. The outputs from each stream were then fused at the feature level, typically by concatenation, prior to the final classification layer. The classifier head was trained from scratch for each fusion setup to ensure a fair comparison.

The pipeline maintained consistent preprocessing across all fusion strategies and data modalities, with no additional adjustments or edge case handling required for specific configurations. The combination of six data modality setups and two fusion strategies provided a comprehensive evaluation framework for understanding the influence of input information and integration method on the accuracy and robustness of chicken action recognition.

6.2.6 Action Recognition Models

To comprehensively evaluate chicken behavior recognition, we compared four action recognition architectures: Video Swin Transformer (Z. Liu, Ning, et al., 2022), TimeSformer (Bertasius et al., 2021), X3D (Feichtenhofer, 2020), and a hybrid CNN+LSTM (He et al., 2016) model. These models span a range from advanced transformer-based methods to more classical sequence modeling approaches, enabling a thorough benchmarking of spatial-temporal feature extraction techniques. Each model was implemented

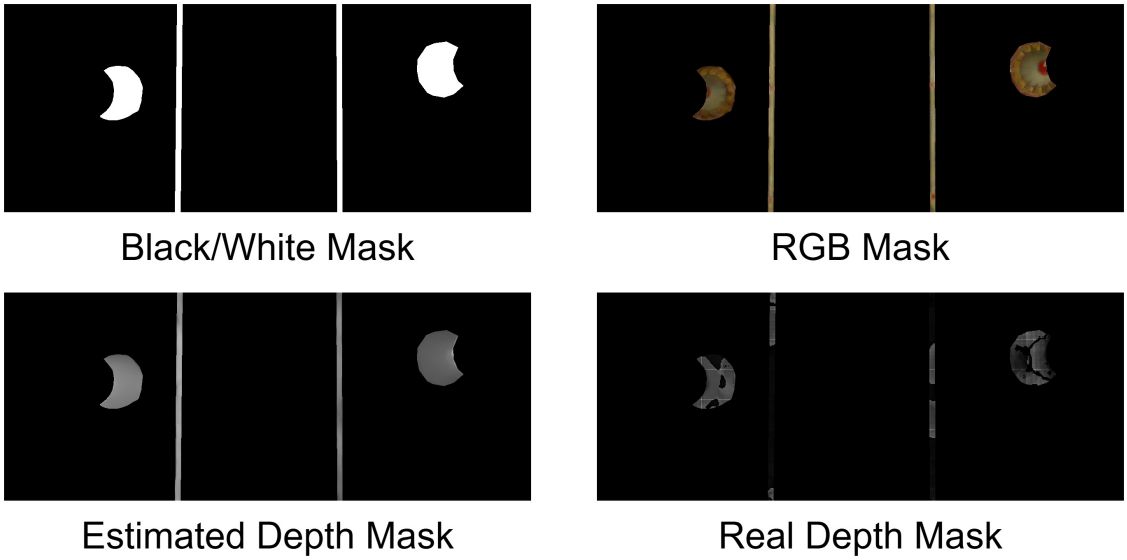

Figure 6.4: Demonstration of samples for four different data modalities used in this study for feeder/drinker masks.

with identical training hyperparameters, using a consistent input data structure across all experiments, and all training was performed from scratch to ensure fairness. The implementations were based on the PyTorch deep learning library, leveraging official modules from torchvision or custom layers where required.

Video Swin Transformer: The Video Swin Transformer (Z. Liu, Ning, et al., 2022) leverages a hierarchical vision transformer backbone that has shown state-of-the-art results for various video understanding tasks. In this work, we utilized a Swin-B architecture adapted to process short video clips of chicken actions. The model first processes each frame using a modified patch embedding to accommodate the concatenated foreground and background channels of our input data, allowing the network to jointly capture fine-grained spatial features and contextual cues. Frames are then passed through several Swin Transformer blocks, where shifted window-based self-attention models both local and global dependencies. The outputs are pooled temporally and classified with a linear head. This architecture offers strong representational power for spatial-temporal patterns and is particularly effective for distinguishing actions with subtle movement cues or context differences. However, the model's complexity and large parameter count can make it slower to train and more memory-intensive than CNN-based alternatives. Implementation was based on the `torchvision.models.swin_transformer` module, with custom adaptation for multi-channel input.

TimeSformer: The TimeSformer model (Bertasius et al., 2021) is a pure transformer architecture designed specifically for video recognition, using divided attention mechanisms across space and time. In our implementation, the model first embeds spatial patches from each frame, combines them with learnable positional and temporal embeddings, and then applies multiple transformer encoder layers to jointly reason about both temporal progression and spatial structure. The advantage of TimeSformer lies in its ability to model long-range dependencies both within and across frames, making it well-suited to capture extended or complex behaviors. It also allows flexible sequence lengths and patch sizes, and its self-attention-based reasoning can better differentiate between similar postures with differing temporal dynamics. On the downside, transformer models generally require large datasets to fully leverage their capacity and may underperform when only limited video data is available. For this work, a custom TimeSformer was implemented in PyTorch, inspired by open-source reference implementations but tailored for our specific input format and scale.

X3D: An efficient 3D convolutional neural network designed for video classification with a focus on high accuracy and low computational cost is called X3D (Feichtenhofer, 2020). The model processes the entire input sequence using lightweight 3D convolutions, progressively increasing spatial-temporal resolution in early layers and channel capacity in later layers. This design enables X3D to capture both appearance and motion cues from short action snippets with a small model size and reduced training time compared to transformer-based models. In our pipeline, the X3D backbone was adapted to accept the custom number of channels present in our fused input, and the final classification layer was adjusted for four action classes. X3D was found to be particularly robust to noisy frames and efficient for training on limited hardware, but it may be less effective at capturing very subtle temporal patterns that transformers can leverage. The model was implemented using the official PyTorchVideo repository, with in-script modifications for our modality fusion and input structure.

CNN+LSTM: As a classical baseline, we included a hybrid CNN+LSTM model (He et al., 2016) to evaluate the benefits of decoupled spatial and temporal modeling. Each frame in the video sequence is first passed through a ResNet-18 backbone (with the first convolutional layer modified for the input channel size), extracting frame-wise feature vectors. These features are then fed sequentially to a multi-layer LSTM, which is responsible for modeling temporal evolution and aggregating information across frames. The output of the last LSTM time step is classified using a fully connected layer. This approach has the advantage of interpretability and lower computational cost, making it suitable for lightweight or real-time applications. However, its sequential LSTM architecture may struggle with very complex temporal dependencies or with differentiating between visually similar behaviors that require longer context windows. Our implementation leverages the `torchvision.models.resnet18` backbone and PyTorch's built-in LSTM modules (Paszke et al., 2019), and follows standard practices for sequence-based action recognition.

All models in this study were built and trained using PyTorch and standard deep learning libraries, including `torchvision`, `PyTorchVideo` (Facebook AI Research (FAIR), 2021), and associated utilities for data processing, logging, and evaluation.

6.2.7 Evaluation Metrics and Training Procedure

To comprehensively assess the performance of the action recognition and object detection models, a consistent set of evaluation metrics was used across all experiments. For the action recognition models, the evaluation included accuracy, precision, recall, F1-score (Bishop, 2006), and inference time. Accuracy measures the overall proportion of correctly classified samples, while precision and recall respectively quantify the rate of correct positive predictions and the proportion of actual positives identified by the model. The F1-score, which is the harmonic mean of precision and recall, provides a balanced measure that accounts for both false positives and false negatives. Inference time was also recorded to assess the computational efficiency of each model, allowing for a practical comparison in terms of real-time deployment capabilities. For the object detection models, in addition to the aforementioned metrics, the mean Average Precision (mAP) was used, with $mAP@0.5$ ($mAP50$) serving as the primary indicator for localization and classification performance. This metric aggregates precision-recall values across different confidence thresholds and is a standard measure in object detection tasks.

All models were trained and validated using the same standardized data splits and protocols. The dataset for each task was divided into training, validation, and test sets in an 80-10-10 ratio, ensuring that the splits were stratified and mutually exclusive. During training, we used the AdamW (Kingma & Ba, 2014) optimizer with a fixed learning rate of 5×10^{-4} and a weight decay of 0.05 for regularization, unless otherwise specified. Early stopping was employed based on validation F1-score, with a patience parameter of 20 epochs, to prevent overfitting and conserve computational resources. The learning rate scheduler reduced the learning rate by a factor of 0.5 if no improvement was observed for five consecutive epochs. Training was performed from scratch for all models, and class balancing was addressed by using a weighted random sampler (Loshchilov & Hutter, 2017) based on the class distribution in the training set. Each model was trained for up to 100 epochs, and the best-performing checkpoint was selected based on the highest validation F1-score. All hyper-parameters, training protocols, and evaluation criteria were kept consistent across models to enable a fair and direct comparison of results.

The implementation of training, evaluation, and metric computation was carried out using the PyTorch deep learning framework, along with Scikit-learn (Pedregosa et al., 2011) for metric calculations, and the `tqdm` (Irshad & other contributors, n.d.) package for progress visualization. Model training and evaluation pipelines, including data augmentation, transformation, and metric logging, were standardized for reproducibility and transparency across all experiments.

6.3 Results

The following section presents the experimental results for all major components of the proposed multimodal action recognition pipeline. We first compare the performance of object detection and segmentation models using both quantitative metrics and qualitative visualizations. Next, we demonstrate the impact of post-processing on segmentation mask quality. Then, we comprehensively evaluate the action recognition models under various modality configurations and fusion strategies. All quantitative values are reported on the test sets, and key qualitative findings are supported by representative figures and tables. Finally, detailed behavioral analyses and downstream insights are presented in the last part of this section.

6.3.1 Object Detection Models Results

Object detection performance was evaluated using five state-of-the-art architectures: **RT-DETR**, **YOLOv**8, **YOLOv**10, **YOLOv**11, and **YOLOv**12 and for all models, we chose the **large** architecture to have a fair comparison. Each model was assessed on the same labeled test set using a suite of metrics: accuracy, precision, recall, F1-score, $mAP50$, and average inference time per frame. Table 6.1 summarizes the results for all models.

Table 6.1: Performance of object detection models on the test set. All values except inference time are reported as percentages (%).

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**mAP**50	**Inference Time (ms)**
RT-DETR	99.71	99.60	99.88	99.77	99.50	102
YOLOv8	99.83	99.86	99.78	99.82	99.48	27.4
YOLOv10	99.60	99.64	99.54	99.59	99.49	27.5
YOLOv11	99.84	99.86	99.81	99.83	99.49	30
YOLOv12	99.76	99.81	99.67	99.77	99.50	44.5

The quantitative results in Table 6.1 demonstrate that all evaluated object detection models delivered excellent performance, with accuracy, precision, recall, F1-score, and mAP50 all above 99.5%. RT-DETR achieved the highest recall (99.88%), suggesting a strong ability to detect almost all annotated chickens. However, its average inference time was substantially higher at 102 ms per frame, making it less suitable for large-scale inference.

The YOLO family of models offered a more favorable balance between detection performance and computational efficiency. Notably, YOLOv11 achieved the highest accuracy (99.84%) and F1-score (99.83%) among all models, together with a precision of 99.86%, recall of 99.81%, and mAP50 of 99.49%. Importantly, YOLOv11 maintained a low inference time of just 30 ms per frame, very close to other YOLO models, enabling scalable deployment across large datasets.

Based on these results, YOLOv11 was selected for use in the downstream segmentation and action recognition pipeline, as it combined near-optimal accuracy with fast and reliable inference. This trade-off

was crucial for processing hundreds of thousands of frames efficiently without compromising detection quality.

6.3.2 Segmentation Models Results

Table 6.2 presents the comprehensive performance metrics and inference times for all evaluated segmentation models, including all major SAM variants, Mobile-SAM, and FastSAM. Each model was evaluated on the same dataset using accuracy, precision, recall, F1-score, and average inference time per frame. All results are reported as averages over the test set, with inference time computed on the same hardware as the action recognition experiments.

Table 6.2: Segmentation model performance and inference time. The numbers for accuracy, precision, $recall$, and F1-score are in percentage (%).

Model	Accuracy	Precision	Recall	F1-score	Inference Time (ms/frame)
SAM-b	98.8	99.0	92.7	95.7	220.4
SAM-l	99.0	98.9	93.8	96.3	443.4
SAM2-t	98.9	97.3	94.9	96.1	95.3
SAM2-s	98.9	97.6	94.7	96.1	93.7
SAM2-b	98.9	97.1	94.8	95.9	141.3
SAM2-l	98.9	96.9	95.2	96.0	281.0
SAM2.1-t	98.9	97.6	94.4	96.0	84.7
SAM2.1-s	98.9	97.7	94.4	96.0	94.3
SAM2.1-b	98.9	97.6	94.4	95.9	131.3
SAM2.1-l	98.9	97.0	95.3	96.1	280.7
Mobile-SAM	98.7	98.4	92.5	95.4	58.0
FastSAM-s	94.6	88.0	82.9	85.4	56.0
FastSAM-x	96.4	90.1	84.3	86.8	61.0

The results show that all SAM-based models, including SAM2, SAM2.1, and their variants, achieved high segmentation performance, with accuracy and F1-score values consistently above 0.95. However, these models also exhibited substantially higher inference times, with most requiring between 93.7 and 443.4 ms per frame. In contrast, Mobile-SAM provided a remarkable balance between speed and accuracy: it achieved 0.987 accuracy and 0.954 F1-score, nearly matching the larger SAM variants, but with an average inference time of only 58.0 ms per frame—making it nearly four to seven times faster than the standard SAM-b and SAM-l models. FastSAM-s and FastSAM-x were indeed the fastest models, with inference times as low as 56.0 ms per frame, but their segmentation performance lagged behind substantially, with F1-score below 0.87.

Given the need for both high segmentation quality and practical processing speed in large-scale video analysis, Mobile-SAM was selected as the preferred segmentation model for downstream action recognition and behavior analysis. Its nearly state-of-the-art accuracy, combined with the lowest inference time

among all accurate models, made it the most effective choice for robust and efficient mask generation. FastSAM, while extremely fast, was not adopted due to its significantly lower accuracy and reliability in complex pen scenes.

6.3.3 Mask Post-processing Outcomes

Mask post-processing substantially improved the visual quality and continuity of the segmentation masks, especially in challenging cases with partial occlusions or noise. The primary effects of post-processing—removal of small noisy regions, filling of minor gaps, and reconnecting disjoint chicken body regions—are illustrated in Figure 6.5. This figure presents side-by-side comparisons of raw segmentation masks and their corresponding post-processed versions, demonstrating the effectiveness of the cleaning and gap-filling procedures. In most typical scenarios, the post-processing routine restored mask completeness, improved mask boundaries, and suppressed erroneous artifacts, resulting in masks that more closely matched the true chicken body shapes.

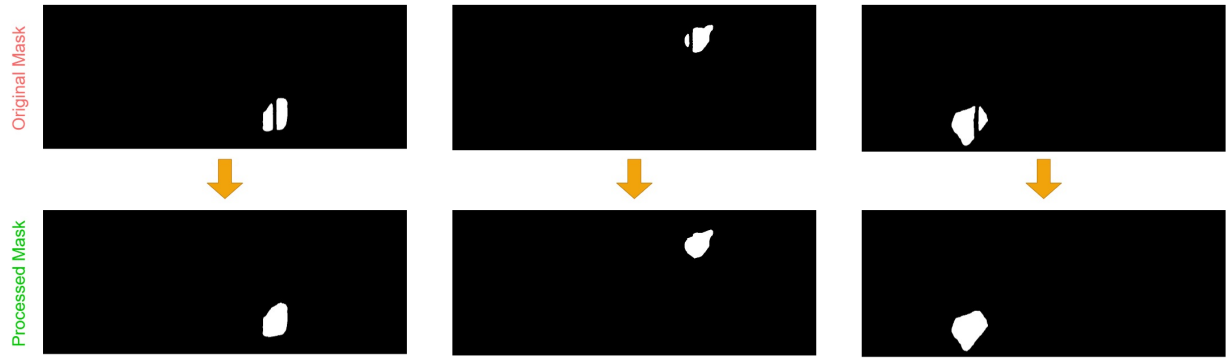

Figure 6.5: Demonstration of a sample synchronized and spatially aligned RGB and depth frame.

Despite these improvements, several persistent failure cases remained due to the inherent limitations of top-down imaging and occlusion scenarios. These are systematically illustrated in Figure 6.6, which provides visual examples for each major failure mode:

1. **Complete occlusion of one side by the drinking line:** When one side of the chicken body is fully obscured by the drinking line, the camera cannot capture or infer the body boundary under the pipe. The resulting mask omits the occluded region entirely (Figure 6.6.a).

2. **Small visible region on one side of the drinking line:** If most of the chicken body appears on one side of the drinking line and only a small region (less than 100 pixels) is visible on the other, this small component is discarded as noise. The occluded area remains unrecovered (Figure 6.6.b).

3. **Disconnected blobs with poor connection across the drinking line:** When the chicken is positioned under the drinker line and segmentation yields two main blobs on opposite sides that are not properly joined—leaving part of the body unconnected—the gap-filling step is not triggered. Only small noise is removed, and the main blobs remain separated (Figure 6.6.c).

4. **Occlusion by the feeder:** In situations where the chicken body is partly or largely hidden by the feeder, segmentation cannot reconstruct the missing portion of the body, leading to incomplete masks (Figure 6.6.d).

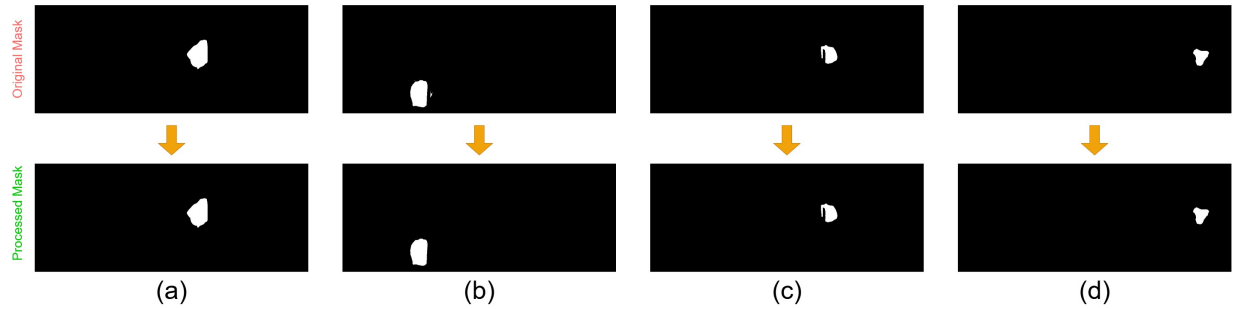

Figure 6.6: Depiction of samples of failure/edge cases for the mask post-processing method used in this study: (a) Complete occlusion of one side by the drinking line; (b) Small visible region on one side of the drinking line;(c) Disconnected blobs with poor connection across the drinking line;(d) Occlusion by the feeder.

Together, Figures 6.5 and 6.6 provide a comprehensive overview of the successes and limitations of the mask post-processing step in this study. While post-processing substantially improved segmentation in the majority of cases, these specific edge cases remain open challenges for future refinement.

6.3.4 Action Recognition Models Results

This part presents the test set performance of all four action recognition architectures—Video Swin Transformer, TimeSformer, X3D, and CNN+LSTM—across the six data modality configurations using the early fusion strategy. For the X3D model, we also present a comparison between early and late fusion strategies to assess the impact of fusion methods on recognition performance. All results are reported as accuracy, precision, recall, F1-score, and average $Inference\ Time$ (ms/sample), as shown in Tables 6.3 and 6.4.

The results in Table 6.3 highlight that, among all tested architectures, X3D consistently outperformed the other models across most modality configurations, achieving the highest accuracy (88.13%) and F1-score (87.95%) with the RGB+estimated depth modality using late fusion. Close behind were the RGB+real depth (87.02% F1-score) and BW+real depth (87.02% F1-score) modalities for X3D, confirming the benefit of leveraging depth information. Video Swin Transformer and TimeSformer showed their best performance with RGB+estimated depth and RGB+real depth modalities, but their accuracy and F1-score remained below 78% and 77.87%, respectively. The classical CNN+LSTM baseline achieved 73.27% F1-score with RGB+estimated depth but trailed the other deep models overall.

In terms of computational efficiency, TimeSformer offered the lowest inference time per sample (approximately 0.31 ms), whereas X3D's inference times were also very fast (0.64–3.11 ms/sample) given its higher accuracy. Notably, the best-performing X3D modality configuration required only 3.02 ms/sample, making it highly practical for large-scale and near real-time deployment.

Table 6.3: Test set accuracy, precision, recall, F1-score, and Inference Time (ms/sample) for all models and modalities using late fusion. The numbers for accuracy, precision, recall, and F1-score are in percentage (%).

Model	Modality	Accuracy	Precision	Recall	F1-score	Inference Time
Video Swin	BW	72.05	73.15	71.74	72.50	4.83
Video Swin	RGB	75.55	75.94	74.08	75.16	5.17
Video Swin	BW+est. depth	73.88	74.06	71.69	73.14	14.47
Video Swin	RGB+est. depth	77.35	78.00	77.65	77.87	14.63
Video Swin	BW+real depth	75.81	75.90	74.38	75.17	14.43
Video Swin	RGB+real depth	76.67	75.96	75.16	75.52	15.10
TimeSformer	BW	70.78	70.54	68.68	69.92	0.31
TimeSformer	RGB	72.33	74.05	71.59	73.07	0.31
TimeSformer	BW+est. depth	70.89	71.23	70.43	70.81	0.32
TimeSformer	RGB+est. depth	73.12	73.84	72.60	73.25	0.32
TimeSformer	BW+real depth	71.70	72.16	71.12	71.77	0.32
TimeSformer	RGB+real depth	73.36	72.96	71.99	72.57	0.32
X3D	BW	75.94	77.21	73.78	75.44	0.64
X3D	RGB	81.40	83.85	81.97	82.92	0.78
X3D	BW+est. depth	79.70	82.52	79.84	81.28	3.11
X3D	RGB+est. depth	88.13	88.59	87.56	87.95	3.02
X3D	BW+real depth	87.50	88.23	85.97	87.02	2.93
X3D	RGB+real depth	85.47	87.54	85.03	86.16	3.03
CNN+LSTM	BW	65.36	66.49	65.19	65.77	2.08
CNN+LSTM	RGB	71.16	72.52	70.80	71.13	2.27
CNN+LSTM	BW+est. depth	69.46	71.12	70.20	70.46	4.46
CNN+LSTM	RGB+est. depth	72.97	74.78	72.02	73.27	4.53
CNN+LSTM	BW+real depth	71.66	71.24	70.89	71.03	4.67
CNN+LSTM	RGB+real depth	71.78	71.30	70.84	71.05	4.76

Table 6.4 presents the results for the X3D model when early fusion is applied. Results indicate that, for some modalities (particularly those incorporating depth information), late fusion yields slightly improved F1-score and accuracy, suggesting that separate processing of mask and depth channels preserves more discriminative features. For example, the RGB+estimated depth modality achieved 81.66% F1-score with early fusion compared to 87.95% with late fusion. Overall, the results showed that late fusion performed better for the X3D model, and it is expected to be the case for the rest of the models, too.

In summary, these findings underscore the value of deep spatiotemporal models and the integration of multimodal data (especially depth) for robust broiler chicken action recognition. The X3D model, particularly with RGB and depth modalities, provides the optimal trade-off between high accuracy and

Table 6.4: Test set accuracy, precision, recall, F1-score, and inference time (ms/sample) for X3D model using early fusion across all modality configurations.

Modality	**Accuracy**	**Precision**	**Recall**	**F1-score**	**Inference Time**
BW	72.44	76.36	72.98	74.53	0.62
RGB	80.68	83.56	79.31	81.45	0.81
BW+est. depth	76.96	80.38	75.05	77.71	0.68
RGB+est. depth	81.73	84.64	79.25	81.66	0.60
BW+real depth	81.36	83.98	77.49	82.06	0.68
RGB+real depth	80.21	82.59	77.90	80.17	0.64

practical inference speed for real-world deployment. Transformer-based models such as Video Swin and TimeSformer may require further optimization or larger datasets to realize their full potential in this application.

6.4 Behavioral Pattern Analysis

The preceding sections established a robust multimodal action recognition pipeline, systematically evaluating object detection, segmentation, and action recognition models to achieve high-accuracy identification of broiler chicken behaviors in commercial pen environments. Building on these results, this section leverages the pipeline's large-scale predictions to conduct an in-depth analysis of behavioral patterns, focusing on action frequency distributions, temporal and spatial trends, and correlations with critical health indicators such as gait score, footpad score, and body weight. By quantifying these patterns, the analysis provides actionable insights into flock activity dynamics, resource utilization, and welfare implications, offering a foundation for data-driven management and precision livestock farming. These findings are crucial for understanding how automated behavioral monitoring can enhance animal welfare, optimize production practices, and inform future research in scalable, vision-based poultry monitoring systems.

6.4.1 Action Frequency Distribution

This subsection reports the overall distribution of $Drinking$, $Feeding$, $Moving$, and $Resting$ actions as identified by the trained action recognition models across the complete dataset. Figure 6.7 shows the percentage of time chickens spent in each action, with $Resting$ emerging as the most frequent behavior, accounting for approximately 61.8% of total observed time. $Drinking$, $Feeding$, and $Moving$ actions followed in descending order, at 19.4%, 12.5%, and 6.3%, respectively.

These results align with the natural behavioral repertoire of broiler chickens, where periods of rest and inactivity dominate the daily activity budget, interspersed with bouts of drinking, feeding, and movement. The observed frequencies provide an empirical baseline for broiler activity allocation under commercial rearing conditions, and set the stage for more detailed temporal, spatial, and welfare-related analyses in subsequent subsections.

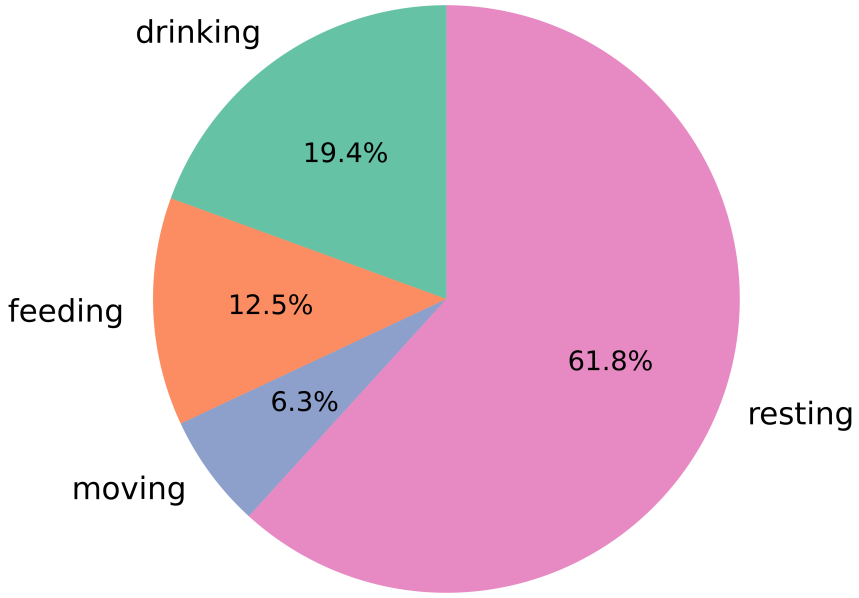

Figure 6.7: Distribution of time spent in each action ($Drinking$, $Feeding$, $Moving$, $Resting$) across the full dataset.

6.4.2 Temporal Patterns of Actions

The temporal distribution of actions across the 24-hour daily cycle reveals key insights into how broiler chickens modulate their behavior in response to time-of-day effects, management routines, and environmental cues. Figure 6.8 presents a stacked line chart where, for each hour (1–24), the percentages of time spent $Drinking$, $Feeding$, $Moving$, and $Resting$ are shown as colored points and the trend of changes are presented with lines connecting the consecutive of each action.

Analysis of this chart indicates that $Resting$ remains the dominant action throughout the day, but $Feeding$ and $Drinking$ activities show pronounced peaks during specific hours, typically coinciding with feed delivery and lighting schedule changes. $Moving$ behavior is the least performed action in most hours and is more evenly distributed. Notably, $Feeding$ activity is highest during the early hours of the morning, reflecting the expected circadian rhythm of broilers under commercial lighting schedules.

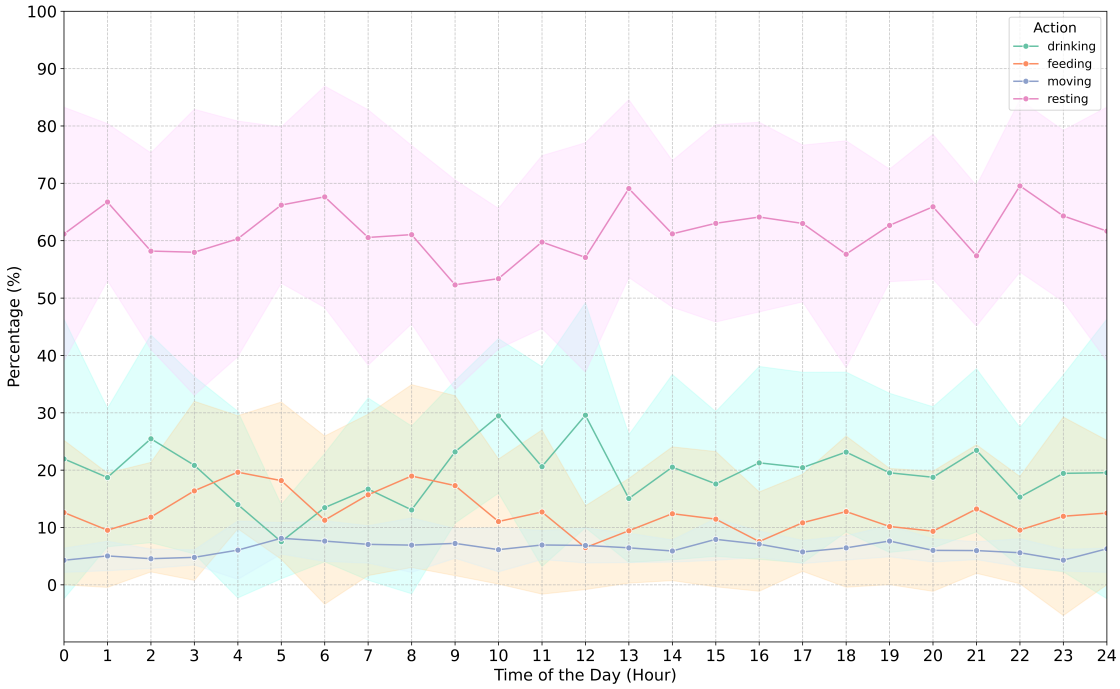

Figure 6.8: Hourly distribution of action percentages across the day. At each specific hour, the action percentages sum up to 100.

6.4.3 Spatial Patterns of Actions

The spatial distribution of chicken activity within the pen was analyzed to determine how different areas are utilized for specific behaviors, and whether any regions of the pen are under-utilized or poorly accessible. Figure 6.9 displays four heatmaps, each representing the spatial distribution of each action around a pen. More specifically, subfigure (a) corresponds to $Drinking$, (b) to $Feeding$, (c) to $Moving$, and (d) to $Resting$. The heatmaps were generated by aggregating the locations of all chickens throughout the entire dataset, grouped by actions.

The heatmaps reveal that $Feeding$ and $Drinking$ are highly localized near the feeder and drinker lines, respectively. $Moving$ and $Resting$, however, tend to occur all around the pen and are distributed more broadly. These findings not only validate the effectiveness of the behavioral recognition pipeline but also provide actionable insights for pen design and resource placement.

6.4.4 Correlation of Actions with Health Indicators

Relationship with Gait Score: To explore how behavioral patterns relate to locomotor health, the percentage of time spent in each action was computed for chickens with gait scores of 0, 1, and 2. Fig-

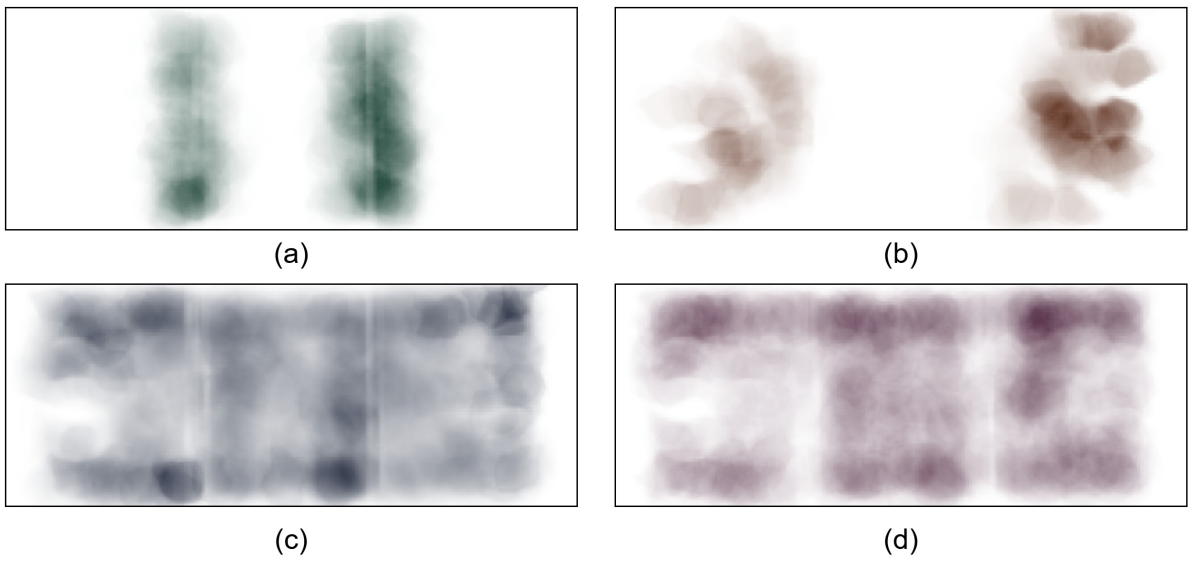

Figure 6.9: Spatial heatmaps showing density of (a) $Drinking$, (b) $Feeding$, (c) $Moving$, and (d) $Resting$ actions within the pen.

ure 6.10.a-c show pie charts for each gait score (0 to 2, respectively). Analyzing and comparing these pie charts leads to the following conclusions:

- The ratio of $Resting$ was higher for chickens with a gait score of 2 (highly impaired movement), which aligns with our expectations.
- The percentage of the $Moving$ action was highest for chickens with a gait score of 0 (optimal mobility), which also aligns with our expectations.
- The percentages of $Drinking$ and $Feeding$ were highest for chickens with a gait score of 1, which was unexpected.

Figure 6.10.d directly compares the percentage of each action for chickens with different gait score values. It can be inferred from the chart that for all chickens, regardless of gait score value, $Resting$ was the most frequently performed action, while $Moving$ was the least observed activity. Additionally, the ratio of each action for chickens with different gait score values is relatively similar, with no significant differences in the ratio of any action for chickens with a specific gait score.

Relationship with Footpad Score: A similar analysis was performed for footpad scores, and Figure 6.11 presents the corresponding pie charts and a stacked bar chart for easier comparison. The charts highlight the following findings:

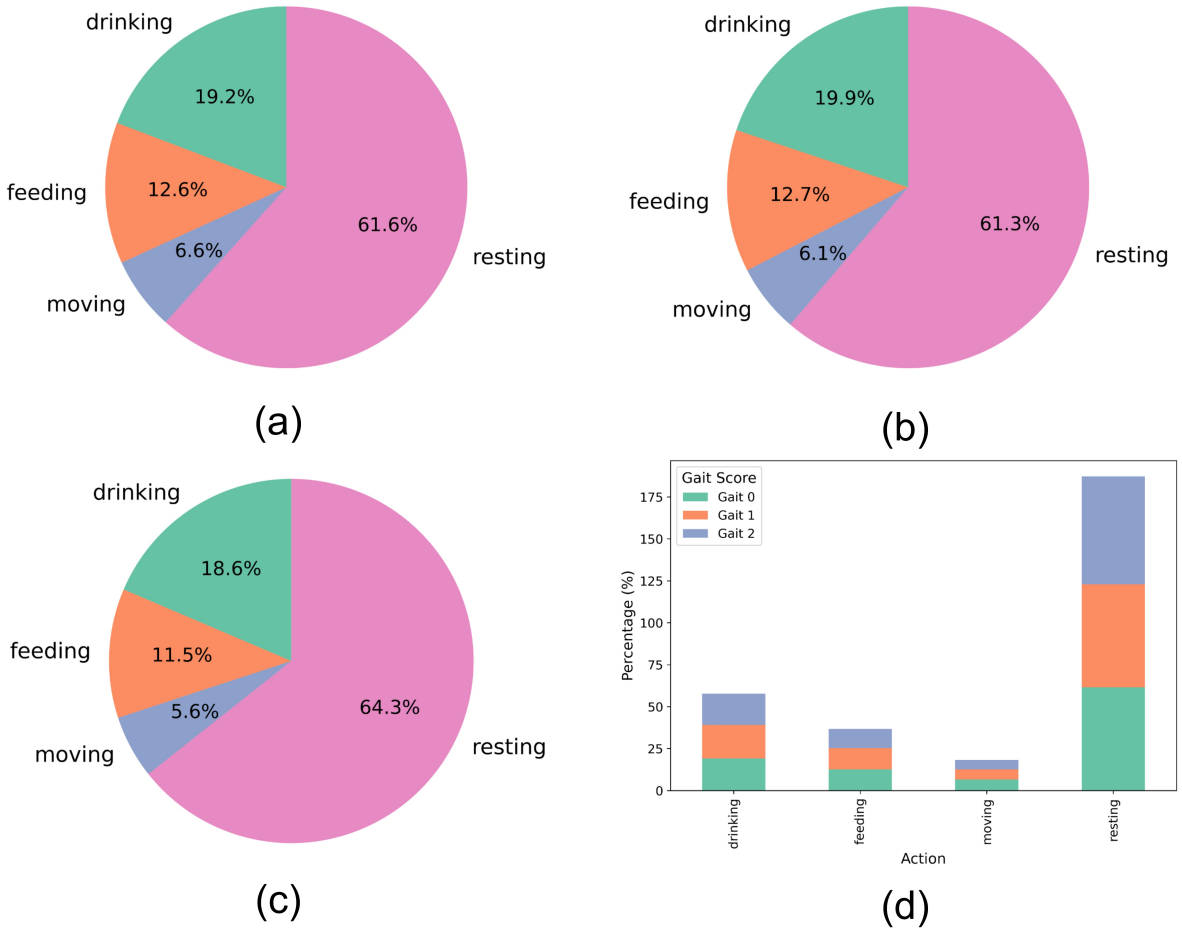

Figure 6.10: Percentage of each action for chickens with gait score (a) 0, (b) 1, and (c) 2. Sub-figure (d) depicts a stacked bar chart comparing the percentage of each action for chickens with varied gait scores.

- Surprisingly, chickens with higher footpad scores (indicative of worse footpad health) rested relatively less than chickens with lower footpad scores (indicative of better footpad health).
- The higher the footpad score, the more $Drinking$ and $Moving$ actions were performed.
- The $Feeding$ action was performed most frequently by chickens with a footpad score of 1, with a 13.8% occurrence.
- Similar to the gait score results, regardless of the footpad score, the order of action occurrences from most to least frequent was $Resting$, $Drinking$, $Feeding$, and $Moving$, respectively.

Relationship with Body Weight: The relationship between individual body weight and action time allocation is visualized in Figure 6.12, which illustrates the distribution of the four actions across six

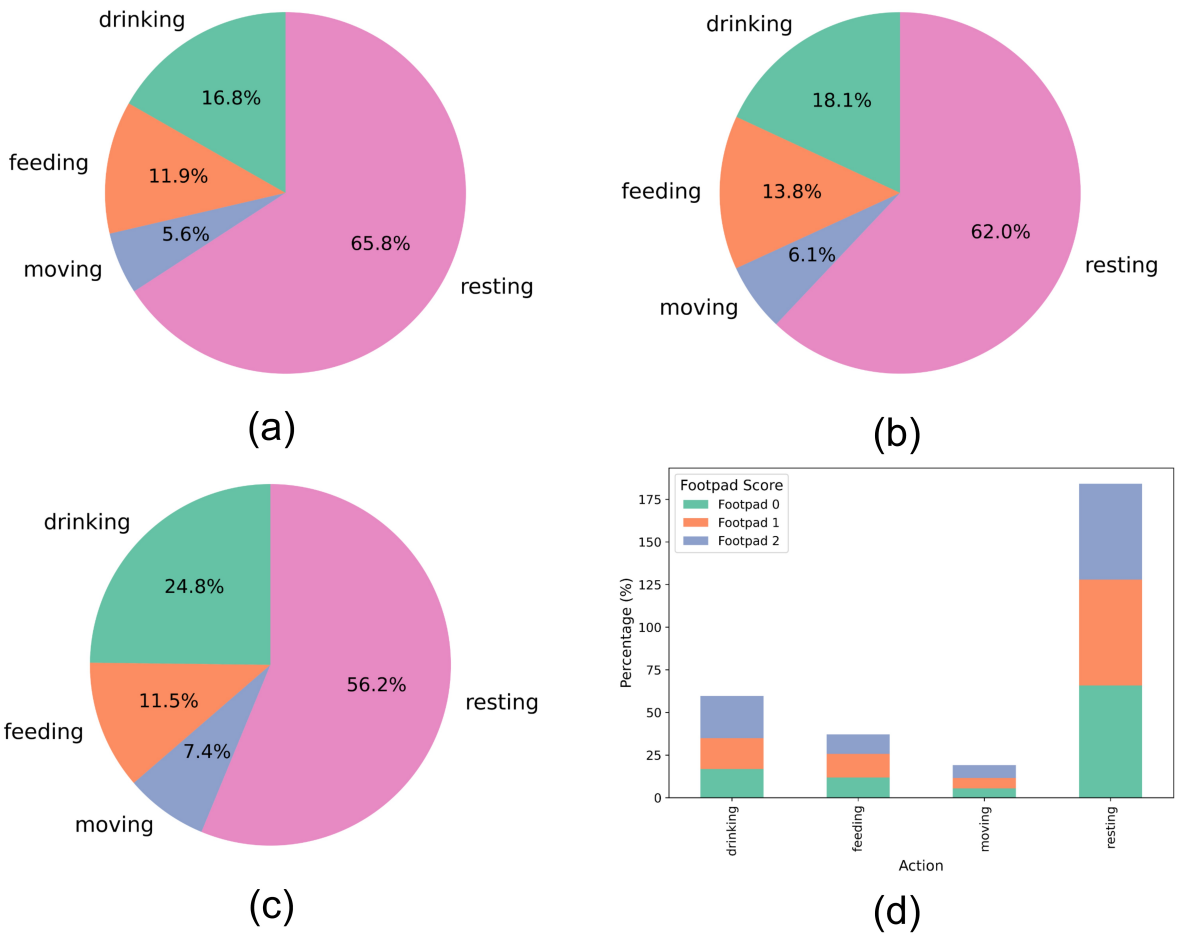

Figure 6.11: Percentage of each action for chickens with footpad score (a) 0, (b) 1, and (c) 2. Sub-figure (d) presents a stacked bar chart comparing the percentage of each action for chickens with different footpad scores.

body weight categories: 2800–3200 grams, 3200–3600 grams, 3600–4000 grams, 4000–4400 grams, 4400–4800 grams, and 4800–5200 grams. The chart combines bar and line elements to not only show the proportion of each action within each weight category, but also to depict the trend of action occurrences as body weight increases. The following observations can be made from the chart:

- Surprisingly, the *Resting* and *Feeding* actions were performed more frequently in the first three body weight categories (lighter chickens) compared to the heavier categories.
- The *Drinking* action was more prevalent among heavier chickens.

- The *Moving* action was performed for nearly the same amount of time across all weight categories. Typically, one might expect lighter chickens to move more than heavier ones, but this was not observed.

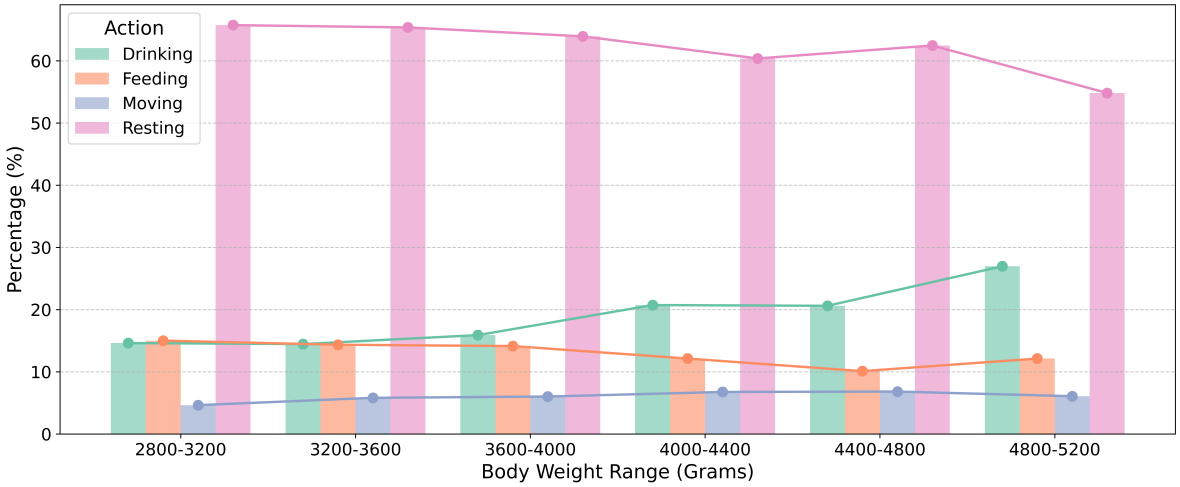

Figure 6.12: A combined bar and line chart depicting the correlation between body weight and the percentage of time spent on *Drinking*, *Feeding*, *Moving*, and *Resting* actions.

6.5 Discussion

In previous sections, we introduced our multimodal action recognition pipeline, explained the methodologies and models involved, presented the evaluation results, and extracted detailed behavior patterns of broiler chickens. In this section, we first discuss the key technical challenges encountered throughout the study and critically examine limitations and potential improvements. Then, we explore additional use cases and applications for the proposed methodology, and finally, we outline promising directions for future research.

6.5.1 Technical Challenges

Throughout the development and deployment of the multimodal action recognition pipeline, several significant technical challenges were encountered.

Data Management and Processing: The size of the collected depth data posed a major obstacle. Each 5-minute RGB-D video amounted to approximately 6.2 GB, totaling nearly 900 GB of raw data and about 1.8 TB post-extraction. Efficient data management, particularly the extraction and precise alignment of RGB and depth frames, required substantial time and computational resources. Storage limitations

necessitated processing data from external hard drives, further slowing the pipeline.

Annotation and Labeling: Manual annotation of object detection and action recognition tasks was labor-intensive. Meticulous labeling of chicken positions and behaviors demanded extensive human effort and careful quality control.

Model Training and Computational Load: Training action recognition models across multiple modality configurations and fusion strategies significantly increased computational complexity. On average, each model required approximately 2 hours per configuration, rapidly accumulating as experiments multiplied. Running training processes on external drives, due to limited local storage, further exacerbated this issue.

Hardware and Scalability Limitations: The post-processing dataset size ($\sim$1.8 TB) tested the limits of memory management and computational throughput even on high-performance hardware. Real-time inference speed also imposed constraints. RGB-D data loading and pre-processing takes 21 ms per RGB and depth frame, Object detection required 30 ms per frame, segmentation takes 58 ms, and action recognition (X3D) required 1 ms per RGB and depth frame. Consequently, total processing time, T, measured in milliseconds (ms), increases with the number of chickens, as defined in Equation 6.1:

$$T = 21\ (\text{loading}) + 30\ (\text{detection}) + 58\ (\text{segmentation}) + 3 \times 1 \times (\text{number of chickens}) \quad (6.1)$$

Based on this formula, real-time recognition is achievable for up to 13 chickens per frame ($\sim$946 ms). More chickens necessitate additional optimization or computational resources.

Image Quality and Environmental Variability: Image clarity occasionally degraded due to dust accumulation, especially near ventilation fans, resulting in noisier RGB data and potential segmentation inaccuracies.

Model-Specific Issues: While the X3D model achieved high accuracy, other models (Video Swin Transformer, TimeSformer, CNN+LSTM) did not surpass 77.87% F1-score, indicating a need for additional tuning or alternative architectures.

Mask Post-processing and Fusion Strategies: Persistent edge cases remained challenging, especially with occlusions caused by drinker lines or feeders. These occlusions sometimes resulted in incomplete mask reconstructions, limiting segmentation accuracy and affecting action recognition reliability.

Overall, these technical challenges are critical considerations for future improvements and large-scale pipeline deployment.

6.5.2 Limitations and Potential Improvements

Despite the high performance of the best-performing configuration (X3D with RGB and estimated depth modalities using late fusion, achieving 88.13% accuracy and 87.95% F1-score), several limitations remain. Misclassifications occurred primarily when chickens rested close to drinking lines (without drinking) or near feeders, as spatial cues overlapped with drinking or feeding actions. Occlusions by feeders prevented object detection, resulting in missed segmentation and action recognition instances. Inconsistent segmentation across consecutive frames due to occlusions sometimes created false movement perceptions.

The zero-shot segmentation model offers strong generalizability; however, the object detection model currently generalizes poorly to naturally colored (white) chickens. Extending the model to other environments or breeds would require additional annotation and retraining.

Manual annotation remains a potential source of bias and error. Efficiency could be improved using semi-supervised or active learning approaches, automated pipelines, or consensus-based annotation methods.

The fixed temporal context of three frames (one second) adequately captured most behaviors but could be enhanced by evaluating longer or variable-length windows, potentially improving recognition of complex behaviors or transitions.

Real-time deployment is currently feasible for up to 13 chickens per pen using an NVIDIA RTX A4500 GPU and 64 GB RAM. Handling more chickens simultaneously would require either more powerful hardware, optimized inference (e.g., ONNX model conversions), or batching/pruning strategies.

Class imbalance was mitigated using a WeightedRandomSampler, but increasing the dataset size for minority actions (drinking, feeding, moving) could further enhance performance.

Cost remains a significant barrier for commercial deployment, given the need for high-performance GPUs. Future work should aim for lightweight, edge-friendly models and user-friendly interfaces to simplify maintenance and reduce operational complexity.

Overall, continued development is required to enhance generalizability, efficiency, affordability, and practical integration into farm management systems.

6.5.3 Other Use Cases and Applications

The developed action recognition framework holds substantial potential beyond the primary application of automated behavioral monitoring in broiler pens.

Continuous, automated welfare assessment can detect early signs of lameness or abnormal behaviors, providing timely alerts for veterinary interventions and building comprehensive welfare records. Action frequency and movement patterns could measure the efficacy of dietary supplements, medication, or environmental changes.

From a management perspective, detailed behavioral analytics can inform optimal feeding schedules, environmental enrichment, and space utilization, improving productivity and bird welfare. Economic benefits include reduced labor costs, early disease detection, and optimized resource allocation.

The system can integrate with existing farm management platforms, scaling across different poultry breeds and livestock species, supporting broad adoption. Additionally, auditable behavioral data can satisfy regulatory compliance, welfare certification, and consumer transparency.

Innovative research opportunities include individual chicken tracking for personality studies, monitoring social interactions, and adaptation to breeder or layer hens. These applications illustrate the diverse impacts achievable with automated action recognition technology.

6.5.4 Future Work

Future research should expand datasets to include varied environmental conditions, breeds, and color variations, enhancing generalizability. Streamlining annotation using semi-supervised, active learning, or improved annotation methods will further reduce manual effort and bias.

Exploring longer or adaptive temporal windows could capture more complex behaviors. Converting models to lightweight, edge-device-friendly formats (ONNX, TensorRT) and developing automated model-update pipelines will facilitate scalable deployment. Improved user interfaces and seamless integration with farm management systems are essential for practical adoption.

Research should extend to other poultry types, livestock species, and new applications like proactive welfare alerts and long-term behavioral studies. These directions will further enhance the robustness and practical utility of automated welfare monitoring.

6.6 Concluding Remarks

This chapter presented a comprehensive comparative study of multiple proposed mask-based multimodal action recognition pipelines for broiler chickens and evaluated early and late fusion strategies across diverse data modalities for one of the deep learning pipelines (X3D model). By integrating high-resolution RGB-D video, robust object detection, zero-shot segmentation, and modern spatiotemporal architectures, the proposed pipeline enables accurate and efficient recognition of key chicken behaviors in real-world pen environments. The findings underscore the importance of leveraging multimodal information—especially depth data—in improving model robustness and action classification performance, while also demonstrating the practicality of scalable deep learning approaches for animal monitoring.

In the Results section, the pipeline's quantitative performance was rigorously benchmarked across multiple stages, including object detection, segmentation, and action recognition. YOLOv11 emerged as the optimal detection model, balancing near-perfect accuracy with fast inference. For segmentation, Mobile-SAM provided high-quality masks with the lowest computational cost among accurate models. Among action recognition models, the X3D architecture, particularly when using the late fusion of RGB and estimated depth modality, achieved the highest accuracy (88.13%) and F1-score (87.95%), alongside rapid inference times suitable for large-scale deployment. Notably, the early fusion strategy was systematically compared to late fusion only for X3D, revealing that late fusion generally resulted in improved performance, especially when incorporating depth information. These outcomes collectively

demonstrate that the fusion of complementary modalities and the selection of efficient architectures are pivotal to achieving high performance and practical scalability in behavioral recognition pipelines.

The Behavioral Pattern Analysis further leveraged the pipeline's large-scale predictions to explore broiler activity allocation, temporal and spatial trends, and correlations with health indicators such as gait, footpad score, and body weight. Results revealed biologically meaningful patterns, with resting as the predominant activity, and drinking, feeding, and moving distributed according to environmental factors, health status, and circadian rhythms. Spatial heatmaps and temporal distributions offered actionable insights into pen resource utilization and daily behavioral cycles. Moreover, the pipeline enabled the empirical assessment of welfare-related variables, highlighting its value for both research and commercial management.

Overall, this work not only establishes a rigorous baseline for automated chicken behavior recognition but also highlights key technical and practical challenges for real-world deployment, such as data management, annotation, and scalability. The demonstrated potential for continuous welfare monitoring, management optimization, and research applications paves the way for future developments aimed at greater generalizability, efficiency, and integration with farm management systems. Continued research and innovation in this area will be critical for advancing precision livestock farming and enhancing animal welfare at scale.

Chapter 7

Concluding Remarks

In this dissertation, we systematically developed and assessed novel spatiotemporal multimodal representation learning methodologies aimed at activity understanding and behavior monitoring, particularly within precision livestock monitoring contexts. The overarching contributions emphasized the effective integration of multimodal data, leveraging deep learning to achieve superior accuracy and robustness in real-world applications. Across these studies, we demonstrated how advanced feature extraction, adaptive fusion strategies, and scalable frameworks significantly enhanced monitoring capabilities, enabling precise and automated livestock assessments.

7.1 Summary of Contributions

The main shared contributions include robust multimodal data fusion, advancements in feature selection strategies, and the introduction of automated real-time assessment frameworks. Each work specifically contributed as follows:

1. **Multimodal Speaker Recognition (DeepMSRF):** We presented DeepMSRF, a multimodal speaker recognition framework that integrates audio and visual modalities using parallel VGGNet streams, significantly outperforming traditional unimodal techniques. Advanced feature selection methods were employed to enhance recognition accuracy.

2. **Low-cost 3D Monitoring System:** An affordable and robust system was developed to automatically monitor broiler chickens, employing depth sensors and YOLOv8 for object detection. Comprehensive experiments validated system configurations and performance under varied environmental conditions, providing practical insights for efficient data storage and management.

3. **Automated 3D Gait Scoring:** A comprehensive deep learning pipeline was proposed to automate gait scoring in broilers. Utilizing synchronized RGB-D data, pose estimation, and frame validation techniques, this work significantly improved precision in classifying gait abnormalities, supporting better animal welfare assessment.

4. **Zero-Shot and Spatiotemporal Transformers for Gait and Footpad Scoring:** A pioneering approach was developed that combined zero-shot segmentation (RAFT3D) with adaptive feature fusion and transformer-based classification. This model effectively classified both gait and footpad conditions, demonstrating significant resilience and accuracy in varied experimental setups.

5. **Mask-Based Multimodal Action Recognition Pipelines:** We proposed and evaluated multiple mask-based multimodal action recognition pipelines for broiler chickens, integrating early and late fusion strategies. These pipelines leveraged high-resolution RGB-D data, robust object detection, and spatiotemporal models like X3D to accurately recognize key behaviors (Drinking, Feeding, Moving, Resting), providing actionable insights for pen design and resource placement.

7.2 Common Challenges

One common challenge encountered was handling noisy, incomplete, or poor-quality data collected from real-world livestock environments. For instance, variable lighting conditions frequently caused issues such as shadows, overexposure, or insufficient illumination, complicating the accurate detection and segmentation of objects. Depth data collected in poultry houses often suffered from noise due to reflections from metal surfaces and interference from environmental factors such as dust and feathers, necessitating advanced preprocessing and noise reduction strategies to ensure accurate results. Additionally, inconsistent camera angles and fluctuating animal positions introduced substantial variability into the dataset, making it essential to develop robust algorithms capable of accommodating such dynamic and unpredictable conditions. In the action recognition pipelines, occlusions by feeders or other chickens further complicated segmentation and behavior classification, requiring sophisticated post-processing techniques to enhance mask quality.

Another prominent challenge was managing the efficient storage of extensive multimodal datasets. Continuous data generation from multiple monitoring devices significantly increased storage requirements, creating practical limitations in terms of data handling and retrieval. For example, a single poultry monitoring system generated several terabytes of data per month, making cloud storage solutions financially impractical. Consequently, external hard drives emerged as a cost-effective alternative, albeit with limitations related to data backup and accessibility. Addressing these storage concerns required careful planning and the exploration of hybrid storage strategies combining local and cloud solutions. For action recognition, the high volume of RGB-D frames necessitated efficient compression methods like Zlib and Blosc to balance storage needs with data integrity.

Ensuring the robustness and generalizability of the deep learning models also presented considerable challenges, primarily due to limited and highly specialized training data. The training datasets frequently suffered from imbalanced class distributions, limited samples per class, and an absence of diverse environmental conditions. This situation heightened the risk of models performing well on training data but poorly on unseen scenarios. To mitigate this, extensive experimentation with techniques such as zero-shot learning, domain adaptation, and strategic augmentation methods was required. These techniques helped improve the models' ability to generalize effectively, even when faced with significantly different

or novel scenarios. In the context of action recognition, distinguishing subtle behavioral differences (e.g., between Feeding and Drinking) required careful model design and robust feature extraction to capture context-specific cues.

Lastly, computational complexity posed a significant challenge as real-time performance was critical for practical deployment in livestock monitoring. Models involving deep architectures and extensive feature extraction methods typically demand substantial computational resources, potentially limiting their applicability in real-world settings with resource-constrained hardware. Balancing accuracy and computational efficiency required optimization of neural network architectures, model pruning, and deployment on specialized edge computing platforms such as Jetson devices. For the action recognition pipelines, achieving real-time performance for up to 13 chickens per frame, with a GPU-enabled device, necessitated careful optimization of detection, segmentation, and classification stages. Successfully addressing these challenges was crucial in enabling real-time, reliable performance without sacrificing accuracy.

7.3 Key Findings

One of the principal findings was that multimodal data fusion significantly enhances accuracy and robustness compared to unimodal methods. This was demonstrated across various experiments, notably in speaker recognition, chicken monitoring, and action recognition applications. Furthermore, advanced feature selection and adaptive fusion strategies were critical in achieving optimal performance.

Another important finding was that real-time monitoring and automated assessments can substantially improve animal welfare monitoring by enabling early detection of health issues. Our automated gait scoring system and action recognition pipelines exemplified this benefit by providing rapid, accurate assessments of individual broiler chickens' health status and behaviors, facilitating data-driven management strategies.

Zero-shot segmentation approaches also emerged as highly effective in addressing data scarcity challenges, successfully leveraging limited labeled datasets to achieve high classification accuracy. Additionally, transformer-based architectures effectively captured complex spatiotemporal relationships in livestock behavior, further enhancing the models' predictive capabilities. The action recognition pipelines further demonstrated that mask-based approaches, combined with early and late fusion strategies, significantly improved the differentiation of subtle behaviors, with the X3D model achieving efficient and accurate performance.

Summarizing the key findings:

- Multimodal fusion greatly improved model performance across speaker recognition, gait scoring, and action recognition.
- Automated real-time assessments significantly enhanced animal welfare monitoring and management.
- Zero-shot techniques efficiently handled limited labeled data.

- Transformer-based and mask-based models excelled at capturing complex spatiotemporal patterns and subtle behavioral differences.

7.4 Future Directions and Improvements

Future research in spatiotemporal multimodal representation learning holds immense potential to further refine and expand the applicability of the methodologies developed in this dissertation. To fully exploit these techniques, continued exploration of advanced deep learning architectures, more efficient model implementations, and innovative data management strategies will be essential. Investigating the integration of additional modalities and adopting robust generalization and adaptability techniques will further enhance performance.

For DeepMSRF, extending evaluations to include a broader variety of datasets is critical for validating model robustness and generalizability. Further exploration into more sophisticated feature extraction and selection methodologies, potentially leveraging state-of-the-art attention mechanisms or autoencoders, can significantly improve the system's accuracy across diverse speaker scenarios. Additionally, adopting continual learning strategies can ensure models remain accurate as new speakers and data scenarios emerge over time.

Regarding the low-cost 3D monitoring system, future efforts should focus on developing lightweight yet accurate deep learning models optimized for deployment on edge computing devices, which are typically resource-constrained. Further expansion and diversification of the datasets collected from various poultry farms under different conditions will help improve the robustness and applicability of the system. Additionally, exploring continual and incremental learning approaches would enable the system to dynamically adapt and continually enhance its performance as new data becomes available, thereby maintaining accuracy and reliability.

In the context of automated gait scoring, future research could significantly benefit from incorporating advanced unsupervised and semi-supervised learning approaches. Such techniques would substantially reduce reliance on large quantities of labeled data, making the system more flexible and applicable across various livestock management scenarios. Furthermore, exploring domain adaptation and transfer learning methods could facilitate model deployment across diverse environmental conditions and varied animal behaviors without extensive retraining.

For the zero-shot perception and transformer-based classification approach, investigating alternative deep learning frameworks such as graph neural networks or vision-language models could be highly beneficial. These models have the potential to capture richer inter-modal interactions and better represent complex behavioral dynamics. Moreover, integrating explainable AI techniques could enhance interpretability and trust in automated decision-making processes, thereby broadening the applicability of this technology in sensitive livestock monitoring contexts.

For the mask-based multimodal action recognition pipelines, future work should focus on optimizing computational efficiency to enable deployment on resource-constrained edge devices while maintaining high accuracy. Exploring lightweight models, such as MobileNet or EfficientNet, could reduce inference

times and make real-time processing feasible for larger flocks. Additionally, incorporating temporal context modeling through advanced recurrent or transformer-based architectures could improve the recognition of behaviors with long-term dependencies. Expanding the dataset to include more diverse environmental conditions and behavioral variations will further enhance model generalizability. Finally, developing user-friendly interfaces for farm managers to interact with the system and interpret behavioral insights will be critical for practical adoption in commercial settings.

Bibliography

Abd Aziz, N., Mohd Daud, S., Dziyauddin, R., Adam, M., & Azizan, A. (2021). A Review of Computer Vision for Real-Time Broiler Monitoring. *IEEE Access*, *9*, 12431–12445. https://doi.org/10.1109/ACCESS.2021.3051464

Afshar, A., Perros, I., Park, H., deFilippi, C., Yan, X., Stewart, W., Ho, J., & Sun, J. (2020). Taste: Temporal and static tensor factorization for phenotyping electronic health records. *Proceedings of the ACM Conference on Health, Inference, and Learning (CHIL)*, 193–203. https://doi.org/10.1145/3368555.3384464

Alirezaei, M., Nguyen, Q. C., Whitaker, R., & Tasdizen, T. (2023). Multi-task classification for improved health outcome prediction based on environmental indicators. *IEEE Access*, *11*, 73330–73339.

Amini, P. V., Shahabinia, A., Jafari, H., Karami, O., & Azizi, A. (2016). Estimating conservation value of lighvan chay river using contingent valuation method.

Amirian, S., Wang, Z., Taha, T. R., & Arabnia, H. R. (2018). Dissection of deep learning with applications in image recognition. *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 1142–1148.

Amraei, S., Abdanan Mehdizadeh, S., & Sallary, S. (2017). Application of computer vision and support vector regression for weight prediction of live broiler chicken. *Eng. Agricult. Environ. Food*, *10*, 266–271. https://doi.org/10.1016/j.eaef.2017.04.003

Anderson, P. A. (2011). Sound, stress, and seahorses: The consequences of a noisy environment to animal health. *Aquaculture*, *311*, 129–135. https://doi.org/10.1016/j.aquaculture.2010.11.005

Arandjelovic, R., & Zisserman, A. (2017). Look, listen and learn. *Proceedings of the IEEE International Conference on Computer Vision*, 609–617.

Arias, V., & Koutsos, E. (2006). Effects of copper source and level on intestinal physiology and growth of broiler chickens. *Poultry Science*, *85*(6), 999–1007.

Ariza-Sentís, M., Vélez, S., Martínez-Peña, R., Baja, H., & Valente, J. (2024). A computer vision system based on deep learning for automatic detection of lameness in broiler chickens. *Computers and Electronics in Agriculture*, *219*, 108757. https://doi.org/https://doi.org/10.1016/j.compag.2024.108757

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). Vivit: A video vision transformer. *Proceedings of the IEEE/CVF international conference on computer vision*, 6836–6846.

Asali, E., & Doshi, P. (2024). Visual irl for human-like robotic manipulation. *arXiv preprint arXiv:2412.11360*.

Asali, E., Doshi, P., & Sun, J. (2023). Mvsa-net: Multi-view state-action recognition for robust and deployable trajectory generation. *arXiv preprint arXiv:2311.08393*.

Asali, E., Shenavarmasouleh, F., Mohammadi, F. G., Suresh, P. S., & Arabnia, H. R. (2021). Deepmsrf: A novel deep multimodal speaker recognition framework with feature selection. In *Advances in computer vision and computational biology: Proceedings from ipcv'20, hims'20, biocomp'20, and bioeng'20* (pp. 39–56). Springer International Publishing.

Asali, E., Li, G., Chen, C., Olukosi, O. A., Oluseyifunmi, I., Abaunza, N. M., Liu, T., Saeidifar, M., Bodempudi, V. U. C., Mandiga, A., et al. (2025). Integration and evaluation of a low-cost intelligent system and its parameters for monitoring three-dimensional features of broiler chickens. *Computers and Electronics in Agriculture*, *237*, 110553.

Asali, E., Negahbani, F., Tafazzol, S., Maghareh, M. S., Bahmeje, S., Barazandeh, S., Mirian, S., & Moshloelgosha, M. (2018). Namira soccer 2d simulation team description paper 2018. *RoboCup 2018*.

Asali, E., Valipour, M., Afshar, A., Asali, O., Katebzadeh, M., Tafazol, S., Moravej, A., Salehi, S., Karami, H., & Mohammadi, M. (2016). Shiraz soccer 2d simulation team description paper 2016. *RoboCup 2016 Symposium and Competitions: Team Description Papers*.

Asali, E., Valipour, M., Zare, N., Afshar, A., Katebzadeh, M., & Dastghaibyfard, G. (2016). Using machine learning approaches to detect opponent formation. *2016 Artificial Intelligence and Robotics (IRANOPEN)*, 140–144.

Attia, Y., Rahman, M., Hossain, M., Basiouni, S., Khafaga, A., Shehata, A., & Hafez, H. (2022). Poultry production and sustainability in developing countries under the covid-19 crisis: Lessons learned. *Animals*, *12*. https://doi.org/10.3390/ani12050644

Aydin, A. (2017). Using 3d vision camera system to automatically assess the level of inactivity in broiler chickens. *Comput Electron Agric*, *135*, 4–10. https://doi.org/10.1016/j.compag.2017.01.024

Aydin, A., Cangar, O., Ozcan, S. E., Bahr, C., & Berckmans, D. (2010). Application of a fully automatic analysis tool to assess the activity of broiler chickens with different gait scores. *Computers and Electronics in Agriculture*, *73*(2), 194–199.

Barnum, G., Talukder, S., & Yue, Y. (2020). On the benefits of early fusion in multimodal representation learning. *arXiv preprint arXiv:2011.07191*.

Bertasius, G., Wang, H., & Torresani, L. (2021). Is space-time attention all you need for video understanding? *ICML*, *2*(3), 4.

Bhuiyan, M. R., & Wree, P. (2024). Animal behavior for chicken identification and monitoring the health condition using computer vision: A systematic review. *IEEE Access*, *12*, –. https://doi.org/10.1109/ACCESS.2024.xxxxx

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Bodempudi, V. U., Li, G., Mason, J. H., Wilson, J. L., Liu, T., & Rasheed, K. M. (2025). Identifying mating events of group-housed broiler breeders via bio-inspired deep learning models. *Poultry Science*, *104*(7), 105126.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Buffinton, K. W., Wheatley, B. B., Habibian, S., Shin, J., Cenci, B. H., & Christy, A. E. (2020). Investigating the mechanics of human-centered soft robotic actuators with finite element analysis. *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, 489–496.

Čakić, S., Popović, T., Krčo, S., Nedić, D., & Babić, D. (2022). Developing object detection models for camera applications in smart poultry farms. *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, 1–5.

Call, T., & Stuesse, A. (2024). Labor shortages and the unmaking of class in mississippi's poultry plants. *Dialect. Anthrop.* https://doi.org/10.1007/s10624-024-09727-x

Campbell, M., Miller, P., Díaz-Chito, K., Hong, X., McLaughlin, N., Parvinzamir, F., Martínez Del Rincón, J., & O'Connell, N. (2024). A computer vision approach to monitor activity in commercial broiler chickens using trajectory-based clustering analysis. *Comput Electron Agric*, *217*, 108591. https://doi.org/10.1016/j.compag.2023.108591

Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-8*(6), 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

Canton, H. (2021). Food and agriculture organization of the united nations—fao. In *The europa directory of international organizations 2021* (pp. 297–305). Routledge.

Caplen, G., Hothersall, B., Murrell, J., Nicol, C., Waterman-Pearson, A., Weeks, C., & Colborne, G. (2012). Kinematic analysis quantifies gait abnormalities associated with lameness in broiler chickens and identifies evolutionary gait differences. *PLoS One*, *7*, e40800. https://doi.org/10.1371/journal.pone.0040800

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. *European Conference on Computer Vision (ECCV)*.

Carraro, A., Sozzi, M., & Marinello, F. (2023). The segment anything model (sam) for accelerating the smart farming revolution. *Smart agricultural technology*, *6*, 100367.

Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6299–6308.

Certified, A. H. (2019). Broiler chickens: Animal welfare standards audit tool [Accessed July 2024]. https://www.americanhumane.org/app/uploads/2021/08/Broiler-Chickens-Audit-Tool.pdf

ChangWen, Y., Yousaf, K., Yang, Z., Rui, K., Bin, P., & KunJie, C. (2018). Effectiveness of computer vision system and back propagation neural network in poultry stunning prediction. *Int. J. Agric. Eng.*

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer vision – eccv 2018* (pp. 833–851). Springer International Publishing. https://doi.org/10.1007/978-3-030-01234-2_49

Cheng, K.-K., Schwing, A. G., Kirillov, A., & Rohrbach, M. (2022). Masked-attention mask transformer for universal image segmentation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chetty, G., & Wagner, M. (2008). Robust face-voice based speaker identity verification using multilevel fusion. *Image and Vision Computing*, *26*(9), 1249–1260.

Chibelushi, C. C., Deravi, F., & Mason, J. S. (1994). Voice and facial image integration for person recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Chung, J. S., Nagrani, A., & Zisserman, A. (2018). Voxceleb2: Deep speaker recognition.

Cramer, J., Wu, H.-H., Salamon, J., & Bello, J. P. (2019). Look, listen, and learn more: Design choices for deep audio embeddings. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3852–3856.

Dadvar, M., Moazami, S., Myler, H. R., & Zargarzadeh, H. (2020). Multiagent task allocation in complementary teams: A hunter-and-gatherer approach. *Complexity*, *2020*(1), 1752571.

de Alencar Nääs, I., da Silva Lima, N., Gonçalves, R., Antonio de Lima, L., Ungaro, H., & Minoro Abe, J. (2021). An analysis of broiler production data based on fuzzy clustering and principal component analysis. *Information Processing in Agriculture*, *8*(3), 409–418. https://doi.org/https://doi.org/10.1016/j.inpa.2020.09.003

De Montis, A., Pinna, A., Barra, M., & Vranken, E. (2013). Analysis of poultry eating and drinking behavior by software eyenamic. *J. Agric. Eng.*, *44*. https://doi.org/10.4081/jae.2013.275

Dhakal, P., Damacharla, P., Javaid, A. Y., & Devabhaktuni, V. (2019). A near real-time automatic speaker recognition architecture for voice-based user interface. *Machine Learning and Knowledge Extraction*, *1*(1), 504–520.

Donahue, J., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. *CVPR*.

Duda, R. O., & Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, *15*, 11–15. https://doi.org/10.1145/361237.361242

Ehsan, T. Z., & Mohtavipour, S. M. (2024). Broiler-Net: A Deep Convolutional Framework for Broiler Behavior Analysis in Poultry Houses.

Elmessery, W., Gutiérrez, J., Abd El-Wahhab, G., Elkhaiat, I., El-Soaly, I., Alhag, S., Al-Shuraym, L., Akela, M., Moghanm, F., & Abdelshafie, M. (2023). Yolo-based model for automatic detection of broiler pathological phenomena through visual and thermal images in intensive poultry houses. *Agriculture*, *13*. https://doi.org/10.3390/agriculture13081527

Etemad, M., Zare, N., Sarvmaili, M., Soares, A., Brandoli Machado, B., & Matwin, S. (2020). Using deep reinforcement learning methods for autonomous vessels in 2d environments. *Advances in Artificial Intelligence: 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13–15, 2020, Proceedings 33*, 220–231.

Facebook AI Research (FAIR). (2021). PyTorchVideo.

Fang, C., Wu, Z., Zheng, H., Yang, J., Ma, C., & Zhang, T. (2024). Mcp: Multi-chicken pose estimation based on transfer learning. *Animals*, *14*. https://doi.org/10.3390/ani14121774

Fang, C., Zhang, T., Zheng, H., Huang, J., & Cuan, K. (2021). Pose estimation and behavior classification of broiler chickens based on deep neural networks. *Comput Electron Agric*, *180*, 105863. https://doi.org/10.1016/j.compag.2020.105863

Fang, C., Zheng, H., Yang, J., Deng, H., & Zhang, T. (2022). Study on poultry pose estimation based on multi-parts detection. *Animals*, *12*. https://doi.org/10.3390/ani12101322

Faysal, M. A. H., Ahmed, M. R., Rahaman, M. M., & Ahmed, F. (2021). A review of groundbreaking changes in the poultry industry in bangladesh using the internet of things (iot) and computer vision technology. *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)*, 1–6. https://doi.org/10.1109/ACMI53878.2021.9528224

Feichtenhofer, C. (2020). X3d: Expanding architectures for efficient video recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 203–213. https://doi.org/10.1109/CVPR42600.2020.00028

Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). Slowfast networks for video recognition. *Proceedings of the IEEE/CVF international conference on computer vision*, 6202–6211.

Feichtenhofer, C., Pinz, A., & Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1933–1941.

Feichtenhofer, C., Pinz, A., & Zisserman, A. (2017). Detect to track and track to detect. *Proceedings of the IEEE International Conference on Computer Vision*, 3038–3046.

Ferede, F. A., & Balasubramanian, M. (2023). Sstm: Spatiotemporal recurrent transformers for multi-frame optical flow estimation. *Neurocomputing*, *558*, 126705.

Fodor, I., Taghavi, M., Ellen, E., & van der Sluis, M. (2024). A deep learning-based computer vision method for automated detection of severe feather pecking in laying hens. *Poultry Science*, *104*, 104724. https://doi.org/https://doi.org/10.1016/j.psj.2024.104724

Gadzicki, K., Khamsehashari, R., & Zetzsche, C. (2020). Early vs late fusion in multimodal convolutional neural networks. *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, 1–6.

Gao, J., Li, P., Chen, Z., & Zhang, J. (2020). A survey on deep learning for multimodal data fusion. *Neural Computation*, *32*(5), 829–864.

Gao, M., Jiang, J., Zou, G., John, V., & Liu, Z. (2019). Rgb-d-based object recognition using multimodal convolutional neural networks: A survey. *IEEE access*, *7*, 43110–43136.

Garner, J., Falcone, C., Wakenell, P., Martin, M., & Mench, J. (2002). Reliability and validity of a modified gait scoring system and its use in assessing tibial dyschondroplasia in broilers. *British poultry science*, *43*(3), 355–363.

George, D., & George, A. (2023). Development of an IoT-based Poultry Farm Monitoring System. *Partners Universal International Innovation Journal (PUIIJ)*, *1*(2), 77–97.

Goyal, V., Yadav, A., & Mukherjee, R. (2024). Deep Learning Applications in Smart Agriculture: A Comprehensive Review. *ACS Agricultural Science & Technology*, *4*(3), 368–388. https://doi.org/10.1021/acsagscitech.3c00329

Gritsenko, A., Xiong, X., Djolonga, J., Dehghani, M., Sun, C., Lucic, M., Schmid, C., & Arnab, A. (2024). End-to-end spatio-temporal action localisation with video transformers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18373–18383.

Haeri, H., Jerath, K., & Leachman, J. (2019). Thermodynamics-inspired modeling of macroscopic swarm states. *Dynamic Systems and Control Conference*, *59155*, V002T15A001.

Hara, K., Kataoka, H., & Satoh, Y. (2018). Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? *CVPR*.

Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., et al. (2020). Array programming with numpy. *Nature*, *585*(7825), 357–362.

Hartung, J., Lehr, H., Rosés, D., Mergeay, M., & Van Den Bossche, J. (2019). Chickenboy: A farmer assistance system for better animal welfare, health and farm productivity. *9th European Conference on Precision Livestock Farming. ECPLF*, 272–276.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2980–2988. https://doi.org/10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications*, *13*(4), 18–28.

Ho, T. K. (1995). Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, *1*, 278–282.

Hofmann, T., Schmucker, S. S., Bessei, W., Grashorn, M., & Stefanski, V. (2020). Impact of housing environment on the immune system in chickens: A review. *Animals*, *10*(7), 1138. https://doi.org/10.3390/ani10071138

Huang, Z., Shi, X., Zhang, C., Wang, Q., Cheung, K., Qin, H., Dai, J., & Li, H. (2022). Flowformer: A transformer architecture for optical flow. *European conference on computer vision*, 668–685.

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on Machine Learning*, *37*, 448–456.

Irshad, H., & other contributors. (n.d.). Tqdm: A fast, extensible progress bar for python and cli.

Ji, B., Li, G., Casey, D., Psota, E., Fitzgerald, R., Gates, R., Banhazi, T., & Liu, Z. (2024). *Automated swine phenotypic trait measurement for commercial breeding management via a three-dimensional scanning system* [under review].

Ji, B., Zheng, W., Gates, R., & Green, A. (2016). Design and performance evaluation of the upgraded portable monitoring unit for air quality in animal housing. *Comput Electron Agric*, *124*, 132–140. https://doi.org/10.1016/j.compag.2016.03.030

Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics yolo (version 8.0.0) [computer software] [Accessed on 2024 August]. https://github.com/ultralytics/ultralytics

Jocher, G., & Qiu, J. (2024). *Ultralytics yolo11* (Version 11.0.0). https://github.com/ultralytics/ultralytics

John, G. H., & Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 338–345.

Kang, X., Zhang, X., & Liu, G. (2021). A review: Development of computer vision-based lameness detection for dairy cows and discussion of the practical applications. *Sensors*, *21*(3), 753.

Karami, O., Yazdani, S., Saleh, I., Rafiee, H., & Riahi, A. (n.d.). A comparison of zayandehrood river water values for agriculture and the environment. *River Research and Applications*.

Karimi, M., & Ahmazadeh, M. (2014). Mining robocup log files to predict own and opponent action. *International Journal of Advanced Research in Computer Science*, *5*(6), 1–6.

Kashiha, M., Pluk, A., Bahr, C., Vranken, E., & Berckmans, D. (2013). Development of an early warning system for a broiler house using computer vision. *Biosys. Eng.*, *116*, 36–45. https://doi.org/10.1016/j.biosystemseng.2013.06.004

Kestin, S., Knowles, T., Tinch, A., & Gregory, N. (1992). Prevalence of leg weakness in broiler chickens and its relationship with genotype. *The Veterinary Record*, *131*(9), 190–194. https://doi.org/10.1136/vr.131.9.190

Khairunissa, J., Wahjuni, S., Soesanto, I., Wulandari, Akbar, A., & Rahmawan, H. (2023). Multi-Object Tracking Algorithm for Poultry Behavior Anomaly Detection. *International Journal of Advanced Smart Convergence*, *12*(1), 11–20. https://doi.org/10.15849/IJASCA.230320.11

Khairunissa, J., Wahjuni, S., Soesanto, I., & Wulandari, W. (2021). Detecting poultry movement for poultry behavioral analysis using the multi-object tracking (mot) algorithm. *2021 8th International Conference on Computer and Communication Engineering (ICCCE)*, 265–268.

Khayami, R., Zare, N., Karimi, M., Mahor, P., Afshar, A., Najafi, M. S., Asadi, M., Tekrar, F., Asali, E., & Keshavarzi, A. (2014). Cyrus 2d simulation team description paper 2014. *RoboCup 2014 Symposium and Competitions: Team Description Papers*.

Kim, D., Lee, Y., & Ko, H. (2019). Multi-task learning for animal species and group category classification. *Proceedings of the 2019 7th International Conference on Information Technology: IoT and Smart City*, 435–438.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment anything.

Kiryati, N., Eldar, Y., & Bruckstein, A. M. (1991). A probabilistic hough transform. *Pattern recognition*, *24*(4), 303–316.

Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., & Klein, M. (2002). *Logistic regression*. Springer.

Knowles, T., Kestin, S., Haslam, S., Brown, S., Green, L., Butterworth, A., Pope, S., Pfeiffer, D., & Nicol, C. (2008). Leg disorders in broiler chickens: Prevalence, risk factors and prevention. *PloS one*, *3*(2), e1545.

Koda, Y., Yoshitomi, Y., Nakano, M., & Tabuse, M. (2009). A facial expression recognition for a speaker of a phoneme of vowel using thermal image processing and a speech recognition system. *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interaction Communication*, 955–960.

Kraaikamp, F., & Meester, H. (2005). A modern introduction to probability and statistics. *Springer: Berlin/Heidelberg, Germany*.

Lamping, C., Derks, M., Groot Koerkamp, P., & Kootstra, G. (2022). Chickennet - an end-to-end approach for plumage condition assessment of laying hens in commercial farms using computer vision. *Comput Electron Agric*, *194*, 106695. https://doi.org/10.1016/j.compag.2022.106695

Lee, C., Adom, A., Markom, M., & Tan, E. (2019). Automated chicken weighing system using wireless sensor network for poultry farmers. *IOP Conference Series: Materials Science and Engineering*, *557*, 012017. https://doi.org/10.1088/1757-899X/557/1/012017

Lei, T., Li, G., Luo, C., Zhang, L., Liu, L., & Stephen Gates, R. (2022). An informative planning-based multi-layer robot navigation system as applied in a poultry barn. *Intelligence Robotics*, *2*, 313–332. https://doi.org/10.20517/ir.2022.18

Li, D., Zhang, K., Li, Z., & Chen, Y. (2020). A spatiotemporal convolutional network for multi-behavior recognition of pigs. *Sensors*, *20*(8), 2381. https://doi.org/10.3390/s20082381

Li, G., Chesser, G., Huang, Y., Zhao, Y., & Purswell, J. (2021). Development and optimization of a deep-learning-based egg-collecting robot. *Trans. ASABE*, *64*, 1659–1669. https://doi.org/10.13031/trans.14642

Li, G., Gates, R., Meyer, M., & Bobeck, E. (2023). Tracking and characterizing spatiotemporal and three-dimensional locomotive behaviors of individual broilers in the three-point gait-scoring system. *Animals*, *13*. https://doi.org/10.3390/ani13040717

Li, G., Huang, Y., Chen, Z., Chesser Jr, G. D., Purswell, J. L., Linhoss, J., & Zhao, Y. (2021). Practices and applications of convolutional neural network-based computer vision systems in animal farming: A review. *Sensors*, *21*(4), 1492.

Li, G., Hui, X., Lin, F., & Zhao, Y. (2020). Developing and evaluating poultry preening behavior detectors via mask region-based convolutional neural network. *Animals*, *10*. https://doi.org/10.3390/ani10101762

Li, G., Xu, Y., Zhao, Y., Du, Q., & Huang, Y. (2020). Evaluating convolutional neural networks for cage-free floor egg detection. *Sensors*, *20*. https://doi.org/10.3390/s20020332

Li, N., Ren, Z., Li, D., & Zeng, L. (2019). Automated techniques for monitoring the behaviour and welfare of broilers and laying hens: Towards the goal of precision livestock farming. *Animal*, *14*(3), 617–625. https://doi.org/10.1017/S1751731119002155

Li, P., Prieto, L., Mery, D., & Flynn, P. (2018). Face recognition in low quality images: A survey.

Liao, Y., Qiu, C., Zhang, Z., Chen, J., Zheng, J., Su, K., Li, H., & Wang, L. (2021). Animal attribute recognition via multi-task learning based on yolox: A multi-task learning network based on yolox to realize target detection and attribute recognition at the same time. *Proceedings of the 2021 5th International Conference on Video and Image Processing*, 7–12.

Lin, J., Cai, Y., Hu, X., Wang, H., Yan, Y., Zou, X., Ding, H., Zhang, Y., Timofte, R., & Van Gool, L. (2022). Flow-guided sparse transformer for video deblurring. *arXiv preprint arXiv:2201.01893*.

Liu, K., He, Y., Xu, B., Lin, L., Chen, P., Iqbal, M., Mehmood, K., & Huang, S. (2023). Leg disorders in broiler chickens: A review of current knowledge. *Animal biotechnology*, *34*(9), 5124–5138.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. (2016). Ssd: Single shot multibox detector. *Computer Vision – ECCV 2016*, 21–37.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11976–11986. https://doi.org/10.1109/CVPR52688.2022.01167

Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., & Hu, H. (2022). Video swin transformer. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3202–3211.

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lubich, J., Thomas, K., & Engels, D. (2019). Identification and classification of poultry eggs: A case study utilizing computer vision and machine learning. *SMU Data Sci. Rev.*, *2*, 20.

Lüddecke, M., & Steger, C. (2022). Image segmentation using text and image prompts. *arXiv preprint arXiv:2205.08453*.

Lv, W., Xu, S., Zhao, Y., Wang, G., Wei, J., Cui, C., Du, Y., Dang, Q., & Liu, Y. (2023). Detrs beat yolos on real-time object detection.

Lv, W., Zhao, Y., Chang, Q., Huang, K., Wang, G., & Liu, Y. (2024). Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer. *arXiv preprint arXiv:2407.17140*.

Maddah, E., & Beigzadeh, B. (2020). Use of a smartphone thermometer to monitor thermal conductivity changes in diabetic foot ulcers: A pilot study. *Journal of Wound Care*, *29*(1), 61–66.

Malik, Y., Ansari, M., Gharieb, R., Ghosh, S., Chaudhary, R., Hemida, M., Torabian, D., Rahmani, F., Ahmadi, H., Hajipour, P., et al. (2024). The impact of covid-19 pandemic on agricultural, livestock, poultry and fish sectors: Covid-19 impact on agriculture, livestock, poultry and fish sectors. *Veterinary Medicine International*, *2024*.

Mandiga, A., Li, G., Wilson, J. L., Liu, T., Bodempudi, V. U. C., & Mason, J. H. (2024). Quantifying nesting behavior metrics of broiler breeder hens with computationally efficient image processing algorithms and big data analytics. *AgriEngineering*, *6*(4).

Mathis, M. W., & Mathis, A. (2018). Deeplabcut: Markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*. https://doi.org/10.1038/s41592-018-0234-5

Merenda, V. R., Bodempudi, V. U., Pairis-Garcia, M. D., & Li, G. (2024). Development and validation of machine-learning models for monitoring individual behaviors in group-housed broiler chickens. *Poultry Science*, *103*(12), 104374.

Michel, V., Prampart, E., Mirabito, L., Allain, V., Arnould, C., Huonnic, D., Le Bouquin, S., & Albaric, O. (2012). Histologically-validated footpad dermatitis scoring system for use in chicken processing plants. *British Poultry Science*, *53*(3), 275–281.

Mohammadi, F. G., & Abadeh, M. S. (2014a). Image steganalysis using a bee colony based feature selection algorithm. *Engineering Applications of Artificial Intelligence*, *31*, 35–43.

Mohammadi, F. G., & Abadeh, M. S. (2014b). A new metaheuristic feature subset selection approach for image steganalysis. *Journal of Intelligent Fuzzy Systems*, *27*(3), 1445–1455.

Mohammadi, F. G., & Amini, M. H. (2019a). Applications of nature-inspired algorithms for dimension reduction: Enabling efficient data analytics. In *Optimization, learning and control for interdependent complex networks*. Springer.

Mohammadi, F. G., & Amini, M. H. (2019b). Evolutionary computation, optimization and learning algorithms for data science. In *Optimization, learning and control for interdependent complex networks*. Springer.

Mohammadi, F. G., Amini, M. H., & Arabnia, H. R. (2020). An introduction to advanced machine learning: Meta-learning algorithms, applications, and promises. *Optimization, Learning, and Control for Interdependent Complex Networks*, 129–144.

Mohammadi, F. G., Arabnia, H. R., & Amini, M. H. (2019). On parameter tuning in meta-learning for computer vision. *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 300–305.

Mohialdin, A., Elbarrany, A., & Atia, A. (2023). Chicken behavior analysis for surveillance in poultry farms. *Int. J. Adv. Comput. Sci. Appl.*

Mortensen, A., Lisouski, P., & Ahrendt, P. (2016). Weight prediction of broiler chickens using 3d computer vision. *Comput Electron Agric*, *123*, 319–326. https://doi.org/10.1016/j.compag.2016.03.011

Mudumuri, S. P., & Biswas, S. (2015). Low resolution face recognition across variations in pose and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(5), 1034–1040.

Murad, M., Yahya, K., & Hassan, G. (2009). Web based poultry farm monitoring system using wireless sensor network. *Proceedings of the 7th International Conference on Frontiers of Information Technology*. https://doi.org/10.1145/1838002.1838010

Nääs, I. d. A., Lozano, L., Abdanan Mehdizadeh, S., Garcia, R., & Abe, J. (2018). A fuzzy logic approach to assess broiler chicken welfare. *Biosystems Engineering*, *173*, 115–123. https://doi.org/https://doi.org/10.1016/j.biosystemseng.2017.11.004

Nakrosis, A., Paulauskaite-Taraseviciene, A., Raudonis, V., Narusis, I., Gruzauskas, V., Gruzauskas, R., & Lagzdinyte-Budnike, I. (2023). Towards early poultry health prediction through non-invasive and computer vision-based dropping classification. *Animals*, *13*. https://doi.org/10.3390/ani13193041

Nasiri, A., Yoder, J., Zhao, Y., Hawkins, S., Prado, M., & Gan, H. (2022). A computer vision and deep learning approach for the detection and quantification of wooden breast in broiler fillets. *Computers and Electronics in Agriculture*, *197*, 106931. https://doi.org/https://doi.org/10.1016/j.compag.2022.106931

Norton, T., Chen, C., Larsen, M. L. V., & Berckmans, D. (2019). Precision livestock farming: Building 'digital representations' to bring the animals closer to the farmer. *Animal*, *13*(12), 3009–3017. https://doi.org/10.1017/S175173111900199X

Nyalala, I., Okinda, C., Makange, N., Korohou, T., Chao, Q., Nyalala, L., Jiayu, Z., Yi, Z., Yousaf, K., Chao, L., & Kunjie, C. (2021). On-line weight estimation of broiler carcass and cuts by a computer vision system. *Poult Sci*, *100*, 101474. https://doi.org/10.1016/j.psj.2021.101474

Okinda, C., Lu, M., Liu, L., Nyalala, I., Muneri, C., Wang, J., Zhang, H., & Shen, M. (2019). A machine vision system for early detection and prediction of sick birds: A broiler chicken model. *Biosys. Eng.*, *188*, 229–242. https://doi.org/10.1016/j.biosystemseng.2019.09.015

Okinda, C., Nyalala, I., Korohou, T., Okinda, C., Wang, J., Achieng, T., Wamalwa, P., Mang, T., & Shen, M. (2020). A review on computer vision systems in monitoring of poultry: A welfare perspective. *Artificial Intelligence in Agriculture*, *4*, 184–208.

Olanrewaju, O., Abdulhafiz, N., & Liman, A. (2024). Review of poultry monitoring using computer vision. *Nigerian J. Phys.*, *33*, 108–113. https://doi.org/10.62292/njp.v33i1.2024.216

Oliveira, D. A. B., Pereira, L. G. R., Bresolin, T., Ferreira, R. E. P., & Dorea, J. R. R. (2021). A review of deep learning algorithms for computer vision systems in livestock. *Livestock Science*, *253*, 104700. https://doi.org/10.1016/j.livsci.2021.104700

Orakwue, S., Al-Khafaji, H., & Chabuk, M. (2022). Iot based smart monitoring system for efficient poultry farming. *Webology*, *19*, 4105–4112.

Oso, O. M., Mejia-Abaunza, N., Bodempudi, V. U. C., Chen, X., Chen, C., Aggrey, S. E., & Li, G. (2025). Automatic analysis of high, medium, and low activities of broilers with heat stress operations via image processing and machine learning. *Poultry Science*, *104*(4), 104954.

Panconi, G., Grasso, S., & Guarducci, S. (2024). Deep-learning-based markerless pose estimation systems in gait analysis: Deeplabcut custom training and the refinement function. *bioRxiv*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Chitta, S., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, *12*(Oct), 2825–2830.

Peña Fernández, A., Norton, T., Tullo, E., van Hertem, T., Youssef, A., Exadaktylos, V., Vranken, E., Guarino, M., & Berckmans, D. (2018). Real-time monitoring of broiler flock's welfare status using camera-based technology. *Biosys. Eng.*, *173*, 103–114. https://doi.org/10.1016/j.biosystemseng.2018.05.008

Peng, X., & Schmid, C. (2016). Multi-region two-stream r-cnn for action detection. *European Conference on Computer Vision*, 744–759.

Pereira, D., Nääs, I. d. A., & da Silva Lima, N. (2021). A Review of Computer Vision Systems in the Poultry Industry. *AgriEngineering*, *3*(2), 394–402. https://doi.org/10.3390/agriengineering3020026

Pereira, D., Nääs, I., & Lima, N. (2021). Movement analysis to associate broiler walking ability with gait scoring. *AgriEngineering*, *3*(2), 394–402.

Pereira, T. D., Aldarondo, D. E., Willmore, L., Kislin, M., Wang, S. S. H., Murthy, M., & Shaevitz, J. W. (2019). Fast animal pose estimation using deep neural networks. *Nature Methods*, *16*(1), 117–125. https://doi.org/10.1038/s41592-018-0234-5

Pramanik, S., Mujumdar, S., & Patel, H. (2020). Towards a multi-modal, multi-task learning based pre-training framework for document representation learning. *arXiv preprint arXiv:2009.14457*.

Ravi, N., Gabeur, V., Hu, Y., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., & Mintun, E. (2024). Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, G., Lin, T., Ying, Y., Chowdhary, G., & Ting, K. (2020). Agricultural robotics research applicable to poultry production: A review. *Comput Electron Agric*, *169*, 105216. https://doi.org/10.1016/j.compag.2020.105216

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster r-cnn: Towards real-time object detection with region proposal networks. *ITPAM*, *39*, 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Rezazadegan, D., Shirazi, S., Upcroft, B., & Milford, M. (2017). Action recognition: From static datasets to moving robots.

Riber, A., Herskin, M., Foldager, L., Sandercock, D., Murrell, J., & Tahamtani, F. (2021). Review of pain and potential pain indicators in broiler chickens. *Veterinary Record*, *189*(7), e454. https://doi.org/10.1002/vetr.454

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, *22*, 400–407. https://doi.org/10.1214/aoms/1177729586

Rohan, A., Rafaq, M., Hasan, M., Asghar, F., Bashir, A., & Dottorini, T. (2024). Application of deep learning for livestock behaviour recognition: A systematic literature review. *Computers and Electronics in Agriculture*, *224*, 109115.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical image computing and computer-assisted intervention – miccai 2015* (pp. 234–241). Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Saeidifar, M., Li, G., Chai, L., Bist, R., Rasheed, K., Lu, J., Banakar, A., Liu, T., & Yang, X. (2024a). A deep learning-based framework for real-time detection and tracking of individual laying hens in cage-free housing systems. *Computers and Electronics in Agriculture*, *226*, 109436. https://doi.org/https://doi.org/10.1016/j.compag.2024.109436

Saeidifar, M., Li, G., Chai, L., Bist, R., Rasheed, K., Lu, J., Banakar, A., Liu, T., & Yang, X. (2024b). Zero-shot image segmentation for monitoring thermal conditions of individual cage-free laying hens. *Computers and Electronics in Agriculture*, *226*, 109436.

Saeidifar, M., Li, G., Lu, J., Chai, L., Bist, R., & Yang, X. (2024). Automatic segmentation of birds using a combination of object detection and foundation image segmentation models. *International Journal of Advances in Electronics and Computer Science*, *11*(7), 1–8.

Sellahewa, H., & Jassim, S. A. (2010). Image-quality-based adaptive face recognition. *IEEE Transactions on Instrumentation and Measurement*, *59*(4), 805–813.

Seraj, E., & Gombolay, M. (2020). Coordinated control of uavs for human-centered active sensing of wildfires. *2020 American control conference (ACC)*, 1845–1852.

Shah, J. H., Sharif, M., Raza, M., Murtaza, M., & Ur-Rehman, S. (2015). Robust face recognition technique under varying illumination. *Journal of Applied Research and Technology*, *13*(1), 97–105.

Shahabinia, A. R., Parsa, V. A., Jafari, H., Karimi, S., & Karami, O. (n.d.). Estimating the recreational value of lighvan chay river uses contingent valuation method.

Shams, A., Becker, D., Becker, K., Amirian, S., & Rasheed, K. (2023). Evolving efficient cnn based model for image classification. *2023 Congress in Computer Science, Computer Engineering, Applied Computing (CSCE)*, 228–235. https://doi.org/10.1109/CSCE60160.2023.00047

Shams, A., Becker, K., Becker, D., Amirian, S., & Rasheed, K. (2024). Evolutionary cnn-based architectures with attention mechanisms for enhanced image classification. In L. Deligiannidis, G. Dimitoglou, & H. Arabnia (Eds.), *Artificial intelligence: Machine learning, convolutional neural networks and large language models* (pp. 107–132). De Gruyter. https://doi.org/10.1515/9783111344126-007

Shenavarmasouleh, F., & Arabnia, H. R. (2019). Causes of misleading statistics and research results irreproducibility: A concise review. *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 465–470.

Shenavarmasouleh, F., & Arabnia, H. R. (2020). Drdr: Automatic masking of exudates and microaneurysms caused by diabetic retinopathy using mask r-cnn and transfer learning.

Shynkaruk, T., Long, K., LeBlanc, C., & Schwean-Lardner, K. (2023). Impact of stocking density on the welfare and productivity of broiler chickens reared to 34 d of age. *Journal of Applied Poultry Research*, *32*(2), 100344.

Silvera, A., Knowles, T., Butterworth, A., Berckmans, D., Vranken, E., & Blokhuis, H. (2017). A review of the welfare of broiler chickens in Latin America. *Poultry Science*, *96*(7), 2013–2017. https://doi.org/10.3382/ps/pex048

Simonyan, K., & Zisserman, A. (2014a). Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, *27*.

Simonyan, K., & Zisserman, A. (2014b). Very deep convolutional networks for large-scale image recognition.

Soans, N., Asali, E., Hong, Y., & Doshi, P. (2020). Sa-net: Robust state-action recognition for learning from observations. *IEEE International Conference on Robotics and Automation (ICRA)*, 2153–2159.

So-In, C., Poolsanguan, S., & Rujirakul, K. (2014). A hybrid mobile environmental and population density management system for smart poultry farms. *Comput Electron Agric*, *109*, 287–301. https://doi.org/10.1016/j.compag.2014.10.004

Sotoodeh, M., & Ho, J. C. (2019). Improving length of stay prediction using a hidden markov model. *AMIA Summits on Translational Science Proceedings*, *2019*, 425.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Supriyanto, E., Isnanto, R., & Purnomo, S. (2023). Application of Deep Learning in Poultry Farming: A Systematic Literature Review. *E3S Web of Conferences*, *448*, 02014. https://doi.org/10.1051/e3sconf/202344802014

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.

Tahmasebian, F., Xiong, L., Sotoodeh, M., & Sunderam, V. (2020). Crowdsourcing under data poisoning attacks: A comparative study. *Data and Applications Security and Privacy XXXIV: 34th Annual IFIP WG 11.3 Conference, DBSec 2020, Regensburg, Germany, June 25–26, 2020, Proceedings 34*, 310–332.

Tan, M., Pang, R., & Le, Q. (2020). Efficientdet: Scalable and efficient object detection. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10781–10790.

Teed, Z., & Deng, J. (2020). Raft: Recurrent all-pairs field transforms for optical flow. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 402–419.

Tian, Y., Ye, Q., & Doermann, D. (2025). Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. *Proceedings of the IEEE international conference on computer vision*, 4489–4497.

USDA National Agricultural Statistics Service. (2024). Poultry - production and value: 2023 summary [Accessed on July 2024]. https://downloads.usda.library.cornell.edu/usda-esmis/files/m039k491c/b2775j31b/9k4213149/plva0424.pdf

Vaezi Joze, H. R., Shaban, A., Iuzzolino, M. L., & Koishida, K. (2019). Mmtm: Multimodal transfer module for cnn fusion. *arXiv preprint arXiv:1911.08670*.

van Erp-van der Kooij, E., & Rutter, S. M. (2020). Using precision farming to improve animal welfare. *CABI Reviews*, (2020).

van der Sluis, M., Ellen, E., de Klerk, B., Rodenburg, T., & de Haas, Y. (2021a). A review of the application of sensor technology in poultry production. *Poultry Science*, *100*(10), 101300. https://doi.org/https://doi.org/10.1016/j.psj.2021.101300

van der Sluis, M., Ellen, E., de Klerk, B., Rodenburg, T., & de Haas, Y. (2021b). The relationship between gait and automated recordings of individual broiler activity levels. *Poultry Science*, *100*(9), 101300.

Van Hertem, T., Norton, T., Berckmans, D., & Vranken, E. (2018). A review of the potential of monitoring technologies for early-warning of health and welfare problems in pigs and poultry. *Biosystems Engineering*, *173*, 93–102. https://doi.org/https://doi.org/10.1016/j.biosystemseng.2017.08.010

Varghese, R., & Sambath, M. (2024). Yolov8: A novel object detection algorithm with enhanced performance and robustness. *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*.

Viazzi, S., Bahr, C., Van Hertem, T., Schlageter-Tello, A., Romanini, C., Halachmi, I., Lokhorst, C., & Berckmans, D. (2014). Comparison of a three-dimensional and two-dimensional camera system for automated measurement of back posture in dairy cows. *Comput Electron Agric*, *100*, 139–147. https://doi.org/10.1016/j.compag.2013.11.005

Voghoei, S., Tonekaboni, N. H., Wallace, J. G., & Arabnia, H. R. (2018). Deep learning at the edge. *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 895–901.

Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., et al. (2024). Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*, *37*, 107984–108011.

Wang, C., Wang, C., Li, W., & Wang, H. (2021). A brief survey on rgb-d semantic segmentation using deep learning. *Displays*, *70*, 102080.

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*.

Wang, W. (2023). X-anylabeling: An open-source tool for image annotation [Repository: https://github.com/ultralytics/u License: GPL-3.0].

Wang, Z. Y., Liu, J., Chen, J., & Chellappa, R. (2025). Vm-gait: Multi-modal 3d representation based on virtual marker for gait recognition. *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 5326–5335.

Wang, Z., Li, F., Taha, T., & Arabnia, H. (2018). 2d multi-spectral convolutional encoder-decoder model for geobody segmentation. *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 1193–1198.

Webster, A. B., Fairchild, B. D., Cummings, T. S., & Stayer, P. A. (2008). Validation of a three-point gait-scoring system for field assessment of walking ability of commercial broilers. *Journal of Applied Poultry Research*, *17*(4), 529–539.

Webster, A., Fairchild, B., Cummings, T., & Stayer, P. (2008). Validation of a system for monitoring the activity of broiler breeders. *Journal of Applied Poultry Research*, *17*(4), 529–539. https://doi.org/10.3382/japr.2007-00100

Wu, D., Cui, D., Zhou, M., & Ying, Y. (2022). Information perception in modern poultry farming: A review. *Comput Electron Agric*, *199*, 107131. https://doi.org/10.1016/j.compag.2022.107131

Wu, Z., Zhang, T., Fang, C., Yang, J., Ma, C., Zheng, H., & Zhao, H. (2023). Super-resolution fusion optimization for poultry detection: A multi-object chicken detection method. *J. Anim. Sci.*, *101*, skad249. https://doi.org/10.1093/jas/skad249

Wurtz, K., & Riber, A. (2024). Overview of the various methods used to assess walking ability in broiler chickens. *Veterinary Record*, *195*(4).

Xian, Y., Lampert, C. H., Schiele, B., & Akata, Z. (2018). Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, *41*(9), 2251–2265.

Xiao, F., Lee, Y. J., Grauman, K., Malik, J., & Feichtenhofer, C. (2020). Audiovisual slowfast networks for video recognition.

Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., & Zhao, H. (2024). Depth anything v2. *Advances in Neural Information Processing Systems*, *37*, 21875–21911.

Yang, X., Molchanov, P., & Kautz, J. (2016). Multilayer and multimodal fusion of deep neural networks for video classification. *Proceedings of the 24th ACM International Conference on Multimedia*, 978–987.

Zahniser, S., Taylor, J. E., Hertz, T., & Charlton, D. (2018). Farm labor markets in the united states and mexico pose challenges for us agriculture.

Zhang, C., Han, D., Qiao, Y., Kim, J. U., Bae, S.-H., Lee, S., & Hong, C. S. (2023). Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*.

Zhang, K., Li, D., Huang, J., & Chen, Y. (2020). Automated video behavior recognition of pigs using two-stream convolutional networks. *Sensors*, *20*(4), 1085. https://doi.org/10.3390/s20041085

Zhang, X., Xu, T., Zhang, Y., Gao, Y., Pan, J., & Rao, X. (2024). Fcs-net: Feather condition scoring of broilers based on dense feature fusion of rgb and thermal infrared images. *Biosys. Eng.*, *247*, 132–142. https://doi.org/10.1016/j.biosystemseng.2024.09.002

Zhang, X., Zou, J., He, K., & Sun, J. (2015). Accelerating very deep convolutional networks for classification and detection. *IEEE transactions on pattern analysis and machine intelligence*, *38*(10), 1943–1955.

Zhang, Y., Yang, X., Liu, Y., Zhou, J., Huang, Y., Li, J., Zhang, L., & Ma, Q. (2024). A time-series neural network for pig feeding behavior recognition and dangerous detection from videos. *Computers and Electronics in Agriculture*, *218*, 108710.

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2022). Bytetrack: Multi-object tracking by associating every detection box. *European conference on computer vision*, 1–21.

Zhao, X., Ding, W., An, Y., Du, Y., Yu, T., Li, M., Tang, M., & Wang, J. (2023). Fast segment anything. *arXiv preprint arXiv:2306.12156*.

Zheng, Y., & Blasch, E. (2023). Facial micro-expression recognition enhanced by score fusion and a hybrid model from convolutional lstm and vision transformer. *Sensors*, *23*(12), 5650.

Zhong, E., del-Blanco, C. R., Berjón, D., Jaureguizar, F., & García, N. (2025). Animalmotionclip: Embedding motion in clip for animal behavior analysis. *arXiv preprint arXiv:2505.00569*.

Zhou, P., Han, X., Morariu, V. I., & Davis, L. S. (2017). Two-stream neural networks for tampered face detection. *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, 1831–1839.

Zhuang, X., Bi, M., Guo, J., Wu, S., & Zhang, T. (2018). Development of an early warning algorithm to detect sick broilers. *Computers and Electronics in Agriculture*, *144*, 102–113.

Zia, A., Sharma, R., Khamis, A., Li, X., Husnain, M., Shafi, N., Anwar, S., Schmoelzl, S., Stone, E., Petersson, L., et al. (2025). A review on coarse to fine-grained animal action recognition. *arXiv preprint arXiv:2506.01214*.

Zorriassatine, F., Burton, E., Boyd, J., Naser, A., & Lotfi, A. (2024). Cross-species insights: Drawing lessons between bird lameness detection and human gait anomaly detection. *Proceedings of the 17th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '24)*, 359–364. https://doi.org/10.1145/3652037.3663919