

UNIVERSIDADE FEDERAL DO PIAUÍ
CENTRO DE CIÊNCIAS AGRÁRIAS
CAMPUS PROFESSORA CINOBELINA ELVAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ZOOTECNIA TROPICAL
DOUTORADO ACADÊMICO EM ZOOTECNIA TROPICAL

Ewerton Costa Sousa

A COMPUTER VISION END-TO-END METHOD FOR SHEEP WEIGHING FROM IMAGES

Teresina

2026

Ewerton Costa Sousa

A COMPUTER VISION END-TO-END METHOD FOR SHEEP WEIGHING FROM IMAGES

Doctoral Thesis submitted to the Graduate Program in Tropical Animal Science (PPGZT) at the Federal University of Piauí (UFPI), within the concentration area of Animal Production in the Tropics, under the research line of Genetics, Animal Breeding, and Animal Reproduction in the Tropics, in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Ph. D.) in Tropical Animal Science.

Supervisor: Prof. José Lindenberg Rocha Sarmento, Ph.D.
Co-supervisor: Prof. Ricardo de Andrade Lira Rabelo, Ph.D.

Teresina

2026

FICHA CATALOGRÁFICA
Universidade Federal do Piauí
Biblioteca Setorial Campus Professora Cinobelina Elvas
Setor de Processamento Técnico

S725c Sousa, Ewerton Costa.
A computer vision end-to-end method for sheep weinghin from images / Ewerton Costa Sousa. -- 2026.
71 f.

Tese (Doutorado) – Universidade Federal do Piauí. Programa de Pós-Graduação em Zootecnia Tropical, Bom Jesus, Piauí, 2026.
"Orientação: Prof. Dr. José Lindenberg Rocha Sarmento".
"Coorientação: Prof. Dr. Ricardo de Andrade Lira Barreto".

1. Zootecnia. 2. Zootecnia de precisão. 3. Visão computacional. I. Sarmento, José Lindenberg Rocha. II. Barreto, Ricardo de Andrade Lira. III. Título.

CDD 636

Bibliotecária: Nayra Fontinele Feijão – CRB3/1249

Ewerton Costa Sousa

A Computer Vision End-to-End Method for Sheep Weighing from Images

Doctoral Thesis submitted to the Graduate Program in Tropical Animal Science (PPGZT) at the Federal University of Piauí (UFPI), within the concentration area of Animal Production in the Tropics, under the research line of Genetics, Animal Breeding, and Animal Reproduction in the Tropics, in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Ph. D.) in Tropical Animal Science.

Approved on June 25, 2026

EXAMINATION COMMITTEE

gov.br Documento assinado digitalmente
JOSE LINDENBERG ROCHA SARMENTO
Data: 14/08/2026 11:02:05-0300
Verifique em https://validar.iti.gov.br

Prof. José Lindenberg Rocha Sarmento, Ph.D.
Supervisor

gov.br Documento assinado digitalmente
RICARDO DE ANDRADE LIRA RABELO
Data: 14/08/2026 10:10:17-0300
Verifique em https://validar.iti.gov.br

Prof. Ricardo de Andrade Lira Rabelo, Ph.D.
Co-supervisor

gov.br Documento assinado digitalmente
NATANAEL PEREIRA DA SILVA SANTOS
Data: 13/08/2026 17:28:44-0300
Verifique em https://validar.iti.gov.br

Prof. Natanael Pereira da Silva Santos, Ph.D.
Internal Examiner

gov.br Documento assinado digitalmente
KELSON ROMULO TEIXEIRA AIRES
Data: 13/08/2026 14:32:15-0300
Verifique em https://validar.iti.gov.br

Prof. Kelson Romulo Teixeira Aires, Ph.D.
Internal Examiner

gov.br Documento assinado digitalmente
TIAGO GONCALVES PEREIRA ARAUJO
Data: 12/08/2026 09:34:35-0300
Verifique em https://validar.iti.gov.br

Prof. Tiago Gonçalves Pereira Araújo, Ph.D.
External Examiner

gov.br Documento assinado digitalmente
PATRICIA GUIMARAES PIMENTEL
Data: 09/08/2026 20:52:39-0300
Verifique em https://validar.iti.gov.br

Prof. Patrícia Guimarães Pimentel, Ph.D.
External Examiner

Teresina
2026

This work is dedicated to those who motivate me every day to strive to be better and who support me unconditionally: Rochelle Rufino (wife); Saulo and Beatriz (children); Maria Auxiliadora (mother); and Gilvan Rodrigues (father).

"[...] Keep the illusion that your flight always takes you higher [...]
(The Flight, Menotti del Picchia)

RESUMO

A pesagem de animais usando balanças tradicionais é um processo que requer muito esforço e tempo das pessoas envolvidas e expõe os animais e os humanos a situações de estresse e risco à saúde. Visando superar esses desafios, métodos de Visão Computacional vem sendo explorados ao longo dos anos com o objetivo de estimar o peso de forma indireta e não invasiva por meio do uso de modelos de regressão nos quais as variáveis preditoras correspondem a atributos extraídos de imagens. Entretanto, os métodos para estimação do peso de ovinos baseados em Visão Computacional encontrados na literatura, de forma geral, ainda apresentam baixa acurácia de predição e/ou baixa capacidade de generalização, principalmente por fazerem a segmentação do animal e a extração dos atributos para predição por meio de técnicas tradicionais de Processamento Digital de Imagens combinadas com imagens RGB. Nesta tese, propõe-se e avalia-se a acurácia, capacidade de generalização e eficiência computacional de um método para estimação do peso de ovinos que se baseia no modelo de Rede Neural Convolucional Efficient-Net-B3 e em imagens de profundidade (depth) capturadas em perspectiva dorsal (top-down). O método proposto opera em um estilo ponta a ponta que dispensa a implementação de etapas explícitas de segmentação e extração de atributos. A acurácia e a generalização do método foram avaliadas por meio de uma combinação de análises quantitativas e qualitativas, garantindo a confiabilidade dos resultados. O método proposto demonstrou ser uma ferramenta não invasiva, simples e eficiente que alcançou acurácia e capacidade de generalização promissoras (R^2 de 0.96, MAE de 1.66 kg e MAPE de 6.48%), contribuindo para o avanço dos métodos tecnológicos para o mercado de pequenos ruminantes. A análise rigorosa da generalização do modelo resultou em informações valiosas sobre as condições que afetam o desempenho da predição, as quais poderiam não ter sido detectadas se o estudo tivesse se baseado em um único conjunto de treinamento/teste. No tocando à eficiência computacional, o resultados demonstram que o modelo foi capaz de predizer o peso de 276 animais por hora em dispositivos com recursos compatíveis com aplicações de pecuária de precisão, o que contribui, portanto, para o desenvolvimento de tecnologias de pesagem não invasivas mais precisas, acessíveis e aplicáveis em condições reais de produção.

Palavras-chave: Visão Computacional, Redes Neurais Convolucionais, Imagem de Profundidade, Zootecnia de Precisão, Ovino, Predição de Peso

ABSTRACT

Animal weighing using traditional scales is a labor-intensive and time-consuming process that exposes both animals and handlers to stressful situations and potential health risks. To overcome these challenges, Computer Vision methods have been explored over the years to estimate body weight indirectly and non-invasively through regression models, in which the predictor variables correspond to features extracted from images. However, Computer Vision-based methods for sheep weight estimation reported in the literature generally still exhibit limited prediction accuracy and/or poor generalization capability, mainly because animal segmentation and feature extraction are performed using traditional Digital Image Processing techniques combined with RGB images. In this thesis, the accuracy, generalization capability, and computational efficiency of a sheep weight estimation method based on the EfficientNet-B3 Convolutional Neural Network architecture and top-down depth images are proposed and evaluated. The proposed method follows an end-to-end approach that eliminates the need for explicit animal segmentation and feature extraction stages. The accuracy and generalization performance of the method were assessed through a combination of quantitative and qualitative analyses, ensuring the reliability of the results. The proposed approach proved to be a simple, efficient, and non-invasive tool, achieving promising accuracy and generalization performance (R^2 of 0.96, MAE of 1.66 kg, and MAPE of 6.48%), thereby contributing to the advancement of technological solutions for the small-ruminant sector. The rigorous evaluation of model generalization provided valuable insights into the conditions affecting prediction performance, which might not have been identified if the study had relied on a single training/testing dataset split. Regarding computational efficiency, the results demonstrated that the model was capable of predicting the weight of up to 276 animals per hour on resource-constrained devices compatible with precision livestock farming applications, thereby contributing to the development of more accurate, accessible, and practical non-invasive weighing technologies for use under real production conditions.

Keywords: Computer Vision, Convolutional Neural Networks, Depth Image, Precision Livestock Farming, Sheep, Weight Prediction

LIST OF ABBREVIATIONS AND ACRONYMS

ACL	Agent Communication Language
AI	Artificial Intelligence
AIC	Akaike Information Criteria
AIoT	Artificial Intelligence of Things
ANN	Artificial Neural Networks
API	Application Programming Interface
BIC	Bayesian Information Criteria
ConvNet	Convolutional Neural Networks
CPU_{95th}	Top 95th Mean CPU Utilization
CV	Computer Vision
DIP	Digital Image Processing
DL	Deep Learning
EAM	Error Analysis Model
Erel	Relative Error
FPS	Frames per Second
GLMM	Generalized Linear Mixed Model
GPS	Global Positioning System
GSM	Global System for Mobile Communications
IoT	Internet of Things
IoU	Intersection over Union
ITH	Inference Throughput
ITR	Inference Time Ratio
MAE	Mean Absolute Error

MAPE	Mean Absolute Percentage Error
MAS	Multi-Agent System
mAP	mean Average Precision
NVR	Network Video Recorder
pHash	Perceptual Hash
PLF	Precision Livestock Farming
PRISMA	Systematic Reviews and Meta-Analyses
PVI	Prediction variability index
PWT	Power Consumption Total
QAM	QI Analysis Model
QI	Laplacian-based Quality Index
QoS	Quality of Service
QVI	QI Variability Index
RAM_{95th}	Top 95th Mean RAM utilization
RFID	Radio Frequency Identification
R^2	Coefficient of Determination
RLT	Response Latency
RMSE	Root Mean Square Error
ROI	Region of Interest
RR	Respiration Rate
SBC	Single Board Computer
SLR	Systematic Literature Review
SSH	Secure Shell
SSIM	Structural Similarity Index Measure
STH	System Throughput
SVM	Support Vector Machine

TOF	Time of Flight
VAM	Variability Analysis Model
ViT	Vision Transformers
WoS	Web of Science

CONTENTS

1 GENERAL INTRODUCTION 12

2 THEORETICAL BACKGROUND 15
2.1 Computer Vision 15
2.1.1 Digital images: concept and representation 15
2.1.2 CV-based systems: approaches and techniques 17
2.1.2.1 Digital image processing 18
2.1.2.2 Machine learning 19
2.1.2.3 Deep Learning 19
2.1.2.4 Statistical models 19
2.2 CV-based weighing systems 20
2.2.1 Main modules and responsibilities 20
2.2.1.1 1st generation 20
2.2.1.2 2nd generation 21
2.2.1.3 3rd generation 21
2.3 Sheep weight prediction from images 22
References 24

3 A SYSTEMATIC LITERATURE REVIEW OF VISION-BASED ANIMAL MONITORING IOT SYSTEMS 27
Preprint submitted to Elsevier 27

4 A COMPUTER VISION END-TO-END METHOD FOR PREDICTING SHEEP BODY WEIGHT FROM DEPTH IMAGES 43
Paper published in the Small Ruminant Research journal (2026) 43

5 A MULTI-VARIANT EVALUATION OF ENERGY AND COMPUTATIONAL EFFICIENCY OF A CONVNET-BASED AIOT EDGE COMPUTING SYSTEM FOR SHEEP WEIGHING 62
Preprinted submitted to Internet of Things (2026) journal 62

6 CONCLUDING REMARKS 73

1 GENERAL INTRODUCTION

In livestock production, regular monitoring of animal weight is an essential practice for herd management, animal husbandry, and genetic improvement. Body weight is considered a key indicator of several animal traits and behaviors, including productive potential, growth pattern, social behavior, optimal slaughter timing, and the detection of health disorders (ZHAO et al., 2023).

A strategy that is still widely used for animal weighing, particularly in small and medium-sized livestock farms in Brazil, is the direct measurement of body weight using conventional scales, on which animals must either be suspended or positioned, depending on the type of scale. This process is labor-intensive and time-consuming, while also exposing both animals and handlers to stressful situations and potential health risks (SANT'ANA et al., 2021). As a result, weighing is often performed less frequently than recommended for effective weight monitoring. As an alternative, some producers estimate body weight through visual assessment, which is an inexpensive approach but is prone to substantial errors, particularly among less experienced farmers (ZHANG et al., 2025). Estimating body weight from manually collected morphometric measurements is another possible approach; however, it causes the same level of discomfort associated with conventional weighing methods. In this context, body weight prediction based on Computer Vision techniques (SZELISKI, 2022) has gained increasing attention in recent years.

Computer Vision (CV)-based animal weighing methods aim to estimate weight using regression models in which the predictor variables correspond to traits extracted—preferably in an automated manner—from images. These methods enable non-invasive weight estimation, minimizing interference with the animals during the weighing process and contributing to improve animal welfare. As a result, they represent a viable alternative for frequent on-farm body weight monitoring (COMINOTTE et al., 2020). Over the years, numerous CV-based approaches for weight estimation in pigs and cattle have been extensively investigated, leading to well-established advances in weighing methods for these species (ZHAO et al., 2023). More recent studies have reported promising results by adopting modern Deep Learning (DL) techniques (AGGARWAL, 2018), particularly Convolutional Neural Networks (ConvNet). In contrast, research on sheep has shown only incremental progress in recent years, despite the growing importance of the small ruminant meat sector, especially in developing regions where production is expanding and accessible technological solutions are increasingly needed (OECD/FAO, 2025).

In general, CV-based methods for sheep weighing reported in the literature still exhibit either limited prediction accuracy or poor generalization capability, primarily be-

cause most rely on traditional Digital Image Processing (DIP) techniques and RGB images for animal segmentation and feature extraction. The methods proposed by Abdelhady et al. (2019), Gurgel et al. (2021), and Sant'Ana et al. (2021) not only achieved relatively low prediction accuracy but were also tailored to the specific breeds used in their studies, namely Barqi, Santa Inês, and Santa Inês–Texel crossbred sheep. Consequently, their performance tends to degrade when applied to sheep with different phenotypic characteristics, particularly coat color patterns. In contrast, Marques et al. (2025) reported highly promising results ($R^2 = 0.92$) by proposing a more advanced approach that employs ConvNet models for both segmentation and feature extraction. However, its reliance on RGB images still limits its generalization capability. In addition, explicit segmentation requires extensive manual image annotation, making the development process labor-intensive and costly (CHEN et al., 2025), while also hindering deployment in new farms with different flocks and environmental conditions.

In this study, generalization is defined as the ability of a method or model to maintain its performance when exposed to previously unseen data and patterns (GOODFELLOW; BENGICCCO; COURVILLE, 2016). More specifically, it refers to the method's ability to accurately predict the body weight of animals whose images were not presented during the training phase and that do not share certain characteristics with the training samples. In this context, the term **non-shared characteristics** refers to variations in animal phenotypic traits, such as coat color, wool level, body size, and age, as well as variations in the image acquisition environment, including background and lighting conditions. These sources of variation are commonly observed across farms, breeds, and crossbred, heterogeneous flocks.

Based on this premise, the following hypothesis is proposed in this thesis:

1. **H1**: An end-to-end Computer Vision model based on a ConvNet is capable of accurately estimating the body weight of sheep without requiring explicit segmentation or manual feature extraction.

2. **H2**: The use of top-view depth images reduces sensitivity to phenotypic and environmental variations, thereby improving the model's generalization capability across different farms.

3. **H3**: EfficientNet architectures can be executed efficiently on resource-constrained edge devices, making them suitable for practical livestock farming applications.

Unlike RGB, depth images represent the distance between the objects in a scene and the camera. When acquired from a top-view perspective, they implicitly encode information about the animal's height and capture anatomical characteristics that are strongly correlated with body weight, such as the spine, rib cage, and musculature

(ZHANG et al., 2025). Furthermore, depth images are not affected by the animal's coat color (MECKBACH; TIESMEYER; TRAULSEN, 2021), enable the acquisition of high-quality three-dimensional measurements even when animals are in motion (SANT'ANA et al., 2021), and accurately capture the three-dimensional surface geometry of objects (HOU et al., 2023). These characteristics contribute to improving the accuracy, robustness, and generalization capability of body weight prediction models.

Therefore, this thesis proposes and evaluates the prediction accuracy, generalization capability, and computational efficiency of an end-to-end Computer Vision method for sheep body weight estimation based on the EfficientNet-B3 ConvNet architecture and top-view depth images. The predictive performance of the method was assessed through both quantitative and qualitative analyses. Quantitative evaluation was conducted using standard regression goodness-of-fit metrics, while Generalized Linear Mixed Model (GLMM) was employed to investigate the effects of experimental, animal-related, and image-related factors on the model's performance. Furthermore, the computational and energy efficiency was evaluated on a resource-constrained edge device (Raspberry Pi 5), a platform commonly used in Precision Livestock Farming (PLF) applications.

The results demonstrate that the proposed method achieved promising performance in terms of prediction accuracy, generalization capability, and computational efficiency, contributing to the advancement of non-invasive sheep body weight estimation technologies and bringing the practical deployment of such solutions in real-world farming environments closer to reality. Furthermore, this study provides valuable insights into the factors that influence the performance and behavior of the proposed method, offering a foundation for future research in this field.

This thesis is organized as follows. Chapter 2 introduces the fundamental concepts of Computer Vision and provides a detailed discussion of the automated sheep weighing methods reported in the literature. Chapter 3 presents a Systematic Literature Review (SLR) aimed at analyzing the state of the art in image-based animal monitoring systems, including weighing methods, with a particular focus on identifying the main system components, quality attributes, and technologies employed in these solutions. Although this chapter is not directly related to the validation of the aforementioned hypotheses (H1–H3), it provides the theoretical foundation and contextual understanding that informed the design of the methods and systems presented in the subsequent chapters. Chapter 4 describes the design, implementation, and evaluation of the body weight prediction method proposed in this thesis, thereby contributing to the validation of hypotheses H1 and H2. Next, Chapter 5 presents the evaluation of the computational and energy efficiency of a weighing system, addressing hypothesis H3. Finally, Chapter 6 concludes the thesis by summarizing its main contributions, discussing their significance to the field, and outlining directions for future research.

2 THEORETICAL BACKGROUND

This chapter presents the main concepts and theoretical foundations required to understand the research and the experiments conducted in this thesis.

2.1 Computer Vision

Computer Vision is a field of Artificial Intelligence (AI) that aims to reconstruct and interpret natural scenes or real-world environments from images captured by cameras (SZELISKI, 2022). In the context of Computer Vision applied to livestock production, the objective is to interpret a scene in order to address a specific problem or operational need. Typical applications include detecting the presence of animals in images, identifying individual animals based on facial or muzzle characteristics, measuring and/or classifying morphological traits such as body weight, withers height, or body posture (e.g., lying down, feeding, standing, or walking), as well as detecting behavioral events (e.g., mounting activity or time spent drinking water). Throughout this thesis, the term scene refers to the real-world environment captured by a camera, whereas image denotes the digital representation of that scene.

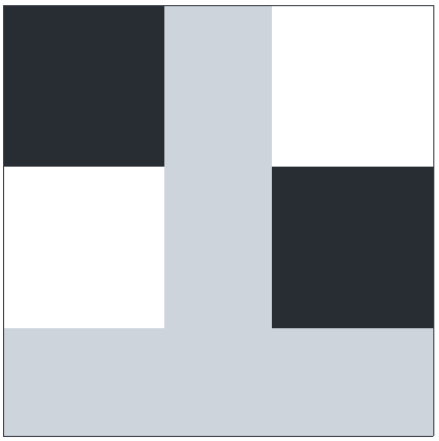
2.1.1 Digital images: concept and representation

A digital image is a representation of a real-world scene that has been captured, processed, and converted by one or more sensors, hereafter referred to simply as a camera. There are several ways to represent real-world scenes as images; however, this thesis considers only their digital representation.

In general, a digital image is represented as an n-dimensional matrix in which each element, referred to as a pixel, is assigned a numerical value whose meaning and range depend on the type of camera used to acquire the image. For RGB cameras, for example, each pixel represents the intensity of light at a particular location in the scene, producing an image that resembles the way humans perceive the world through colors and light intensities. This is the type of image typically generated by smartphone cameras, webcams, and surveillance cameras. However, other imaging modalities also exist. In the case of depth cameras, for instance, each pixel represents the distance between a point in the scene and the camera, typically measured in millimeters.

Figure 1 illustrates a simple digital image (left) and its corresponding two-dimensional matrix representation (right). The image consists of 64 pixels arranged in 8 rows and 8 columns, where each pixel value represents the brightness intensity of the corresponding

point in the scene. Pixel values range from 0, indicating the absence of brightness (black), to 255, representing the maximum brightness (white). Therefore, the image has a spatial resolution of 8 × 8 pixels.



0	0	0	90	90	255	255	255
0	0	0	90	90	255	255	255
0	0	0	90	90	255	255	255
255	255	255	90	90	0	0	0
255	255	255	90	90	0	0	0
255	255	255	90	90	0	0	0
90	90	90	90	90	90	90	90
90	90	90	90	90	90	90	90

Figure 1 – Grayscale image with a spatial resolution of 8 × 8 pixels and the corresponding matrix representation of its pixel values.

RGB color images, in turn, are represented as three-dimensional matrices, where the third dimension consists of three channels corresponding to the intensities of the red (R), green (G), and blue (B) color components. Figure 2 illustrates an image composed of randomly generated colors (left), followed by the corresponding two-dimensional (8 × 8) matrices for the R, G, and B channels. The final color of each pixel is obtained by combining the intensity values from these three channels. Consequently, the image has dimensions of 8 × 8 × 3, where the first two dimensions represent the number of rows and columns in each channel, and the third dimension corresponds to the three color channels. In addition to the RGB color model illustrated in Figure 2 and the grayscale representation shown in Figure 1, several other image representations exist, each designed for specific applications (AFRIDI et al., 2024).

In general, the higher the spatial resolution and the greater the number of channels in an image, the larger the number of pixels used to represent it, resulting in a more detailed representation of the scene. Consequently, higher-resolution images require more memory for storage and greater computational resources for processing. Images are therefore considered high-dimensional data structures with inherent spatial relationships among their pixels, making them a rich source of information. However, these characteristics also increase the computational complexity associated with their storage and processing.

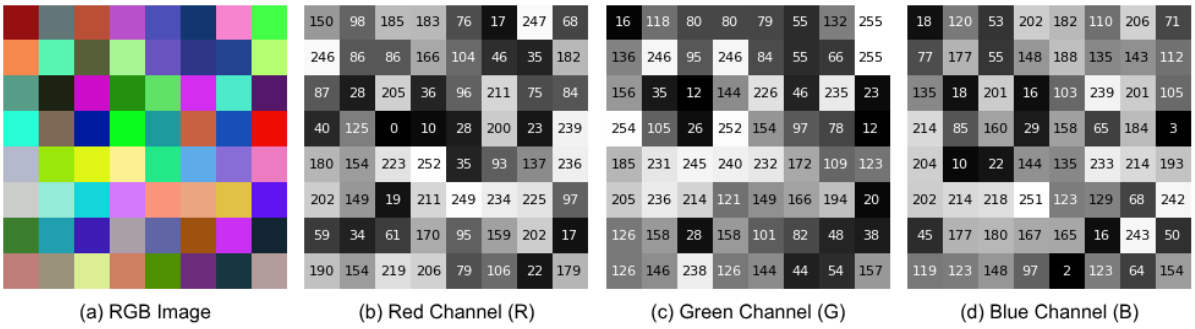


Figure 2 – Original colored image (a) and the corresponding matrix representations of the red, green, and blue channels of the RGB image.

2.1.2 CV-based systems: approaches and techniques

Since the term Computer Vision refers to the research field itself, this thesis adopts the term CV-based system to denote systems that employs CV techniques or algorithms to accomplish a predefined task. They are typically composed of a sequence of image processing stages or modules organized as a pipeline, in which each module receives the output of the previous stage, performs a specific processing task, and passes its output to the next stage until the final result is produced.

A hypothetical example of such a pipeline is illustrated in Figure 3. In this example, Module 01 receives an input image and determines whether an animal is present in the scene. Module 02 identifies the animal by reading the identification number from its ear tag. Next, Module 03 segments the animal by isolating the pixels corresponding to its body and computes its projected body area. Module 04 then estimates the animal's body weight based on the computed area, and finally, Module 05 presents the estimated weight to the user.

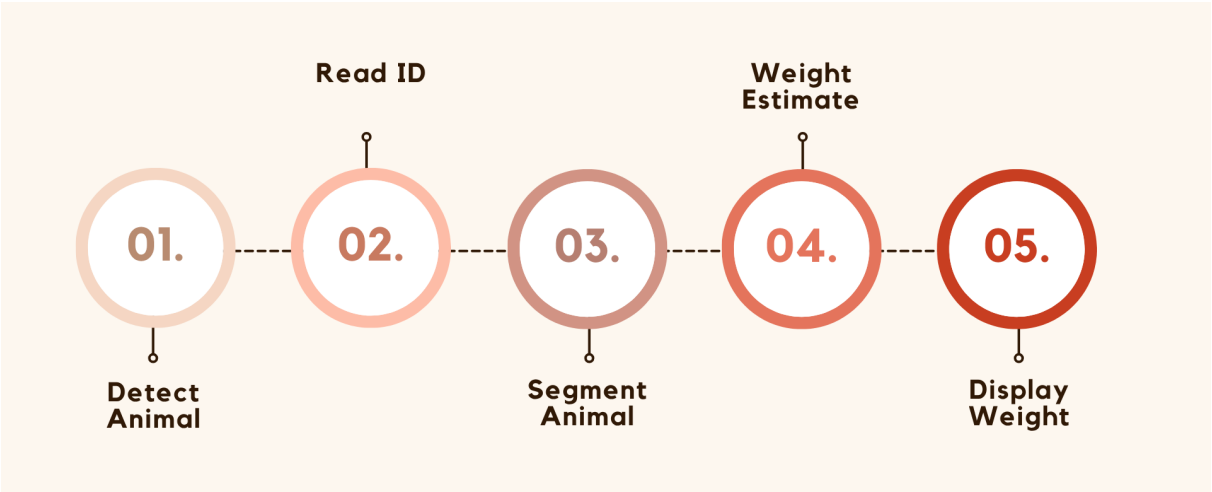


Figure 3 – Architecture of a hypothetical CV-based system composed of five processing modules.

Throughout the processing stages illustrated in Figure 3, different techniques may be employed to accomplish the objective of each module. The following sections introduce the main subfields of Computer Vision and the techniques most commonly used in CV-based systems. Before proceeding, however, it is important to define a few additional terms that will be used throughout the remainder of this thesis:

1. **Technique**: Refers to a specific approach or procedure used to accomplish a particular task or solve a given problem. A technique is typically implemented through one or more algorithms and is based on previously established methods or formal approaches.
2. **Model**: A computational mechanism capable of receiving input data of a given type and producing predictions or outputs, possibly of a different type (ZHANG et al., 2023). Models are built upon techniques that define how these predictions are generated and may belong to different categories, such as Statistical, Machine Learning, or Deep Learning models.
3. **Algorithm**: A well-defined and ordered sequence of operations designed to perform a specific task on a computer. In practical terms, an algorithm is the computational implementation of a technique or model in a programming language.

2.1.2.1 Digital image processing

The concept of Digital Image Processing (DIP) is often confused with Computer Vision or related fields, such as Image Analysis, and there is no consensus in the literature regarding the boundaries between these research areas. However, within the scope of this thesis, DIP refers to a set of processing techniques whose input is necessarily a digital image. The output may also be an image, but it may additionally involve techniques for feature extraction and the recognition of individual objects within images (GONZALEZ; WOODS, 2018). These image processing techniques can be classified into three levels: low-level, mid-level, and high-level.

Low-level techniques are characterized by the fact that both their input and output are images. They generally involve preprocessing operations aimed at reducing noise or enhancing image quality through modifications of pixel values. These techniques take an n-dimensional matrix **X** as input and produce a new matrix **X'**, potentially transformed as a result of the applied operations. The resulting matrix does not necessarily preserve the same dimensions as the original input.

Mid-level techniques, in turn, involve the detection of objects or primitive structures within images, such as lines, contours, and edges, the segmentation of images into regions, and the classification (recognition) of individual objects with relatively simple characteristics. These techniques are characterized by taking images as input and representing them in a different domain that is more suitable for computational analysis.

Finally, high-level techniques involve the interpretation of a scene within the context of the specific problem to be solved. For instance, a segmentation technique may determine the region occupied by an animal in an image and discard all irrelevant background information. These techniques are typically preceded by, or integrate, mid-level techniques that identify primitive structures, which are subsequently used to perform higher-level interpretations, such as prediction, classification, or inference.

2.1.2.2 Machine learning

Machine Learning (ML) is a branch of Artificial Intelligence focused on the development of algorithms and statistical models that enable computational systems to automatically identify patterns in data and perform predictions or decisions based on these patterns, without the need for explicitly programmed rules.

In general, ML techniques applied to CV-based systems do not directly process images, but instead operate on collections of features extracted from images in previous stages. Consequently, they are commonly adopted in CV system modules that are preceded by feature extraction procedures. Neural network-based ML techniques, for instance, are more effective than traditional statistical models in capturing complex patterns and nonlinear relationships in image data, as observed in tasks such as body weight prediction and image classification.

2.1.2.3 Deep Learning

Deep Learning (DL) is a subfield of Machine Learning that relies on computational architectures based on artificial neural networks composed of multiple hierarchical layers, which are capable of automatically learning high-level representations from raw data (GOODFELLOW; BENGICCCO; COURVILLE, 2016). Through successive nonlinear transformations, these models learn increasingly complex feature representations, enabling state-of-the-art performance in tasks such as classification, pattern recognition, natural language processing, and Computer Vision, especially in data-intensive scenarios. A key distinction between DL and conventional ML approaches is the ability of DL models to automatically learn feature representations directly from raw images, reducing the need for handcrafted feature extraction and improving model generalization.

2.1.2.4 Statistical models

Statistical models are parametric mathematical representations that describe relationships among one or more variables based on empirical data. They are employed to identify patterns, estimate model parameters, generate predictions, and quantify uncertainty.

2.2 CV-based weighing systems

Body weight prediction from images can be formulated as a typical regression problem, in which the goal is to estimate an animal's body weight based on features extracted from images that exhibit some degree of correlation with body weight, such as body area, morphometric measurements, and other relevant visual attributes.

2.2.1 Main modules and responsibilities

In general, CV-based systems reported in the literature consist of at least four sequential and interdependent stages: image enhancement, image segmentation, feature extraction, and prediction.

1. **Image enhancement**: commonly referred to as preprocessing, this stage aims to improve image quality in order to make the images more suitable for the specific problem being addressed.
2. **Segmentation**: the objective of this stage is to locate the region (set of pixels) corresponding to the animal in the image, separating it from the remaining content.
3. **Feature extraction**: once the animal(s) have been located, this stage aims to transform the pixels corresponding to the animal into features that represent characteristics from which body weight can be estimated.
4. **Prediction**: the stage in which body weight is estimated based on the features extracted from the image and on a technique capable of learning the relationships between these features and body weight.

The following sections present the main approaches adopted in studies that propose CV-based systems for animal body weight prediction from images. In this chapter, the term **approach** refers to the way in which DIP, ML, DL, and statistical model techniques are employed within the modules of a CV-based system. The categories were defined based solely on the grouping of studies identified in the literature and are not intended to be exhaustive. Instead, they aim to facilitate the understanding of how the proposed approaches have evolved and been improved over time (Table 1).

2.2.1.1 1st generation

In this approach, DIP techniques are predominantly employed in the image enhancement, segmentation, and feature extraction modules. In the prediction module, body weight estimation is performed by adopting linear or nonlinear statistical regression models, such as Multiple Linear, LASSO, and Partial Least Squares regression. Song et al. (2018),

for example, predicts cattle body weight using multiple linear regression models in which the predictor variables represent morphological characteristics of the animals correlated with body weight. To extract these measurements from images, they apply DIP techniques capable of identifying specific skin markings on the animals and subsequently calculate the distances between the boundary points of these markings to estimate the animal's body area.

Approach	Modules			
	Image enhancement	Segmentation	Feature extraction	Prediction
1st generation	DIP	DIP	DIP	Statistical model
2nd generation	DIP	DIP	DIP	ML
3rd generation	DIP	ML	DL	DL

Table 1 – Techniques commonly used in each CV-based system module for each approach.

2.2.1.2 2nd generation

In this approach, DIP techniques are still predominantly employed; however, ML techniques are introduced in the prediction stage, including Support Vector Machines, Artificial Neural Networks (ANN), and Random Forest, as these methods are capable of more accurately identifying nonlinear relationships from image-derived features. As previously mentioned, Cominotte et al. (2020) employs ML techniques, specifically ANN, to predict cattle body weight. As expected, the results obtained using ML techniques outperformed those achieved with statistical models. In turn, Sant'Ana et al. (2021) uses DIP techniques to extract eleven features from sheep images. In that study, the performance of eight ML techniques was evaluated, and the Random Forest-based regressor achieved the best results.

2.2.1.3 3rd generation

This is the most advanced approach, representing the current state of the art. It is characterized by reducing the use of DIP techniques, which are mainly employed during the image enhancement stage to improve image quality and reduce noise, while incorporating DL techniques in the remaining stages. In some scenarios, DL techniques are used as feature extractors, encompassing the segmentation, feature extraction, and prediction stages. For example, Meckbach, Tiesmeyer e Traulsen (2021) applies DIP techniques, including Image Blur Detection, the Structural Similarity Index Measure (SSIM), and image resizing, for image enhancement, while the remaining stages—animal localization, feature extraction, and prediction—were performed using a ConvNet based on EfficientNet-B0.

2.3 Sheep weight prediction from images

Methods for predict sheep body weight from images have shown only incremental advances in recent years and still present limited prediction accuracy and generalization capability. These methods remain largely focused on traditional DIP techniques, which makes them highly specialized for the morphological characteristics of the animals investigated in each study. Consequently, the reported results cannot be directly extrapolated to animals that do not exhibit similar characteristics. Furthermore, estimating sheep body weight from images is particularly challenging because this species may exhibit considerable phenotypic variation among breeds, such as differences in wool levels, and stress-related evasive behaviors, including escape attempts, running, jumping, and other movements (ATKINSON; DOYLE; JONGMAN, 2024). These factors can compromise image quality and negatively affect prediction performance.

Image-based sheep body weight prediction methods were preceded by studies aimed at estimating animal weight from biometric measurements. For example, Gurgel et al. (2021) proposes a linear regression model to estimate the body weight of lambs based on biometric measurements collected from 51 animals, including 26 males and 25 females, all with at least 50% Santa Inês genetic composition and ages ranging from 7 to 91 days. In addition to body weight, the following biometric measurements were collected: withers height (WH)—the distance from the highest point of the withers to the ground; rump height (RH)—the distance from the coxal tuberosity to the ground; body length (BL)—the distance from the acromion to the end of the ischium; chest width (CW)—the distance between the acromions; rump width (RW)—the distance between the ischial tuberosities; heart girth (HG)—measured around the thoracic cavity; and abdominal circumference (AC)—measured around the abdominal region. These measurements were manually collected using a measuring tape. After fitting and evaluating the regression model, only the biometric measurements BL, AC, and RH were selected to estimate the animals' body weight. Thus, the following equation was generated to estimate lamb body weight: $Weight = 0.4455 * CC - 0.5794 * AC + 0.0053 * AC^2 + 0.0019 * RH^2$. The model achieved a coefficient of determination (R^2) of 0.88 and a root mean square error of prediction (RMSE) of 1.25. Despite the promising results, methods based on this approach require manual measurement of animals' biometric traits, which is a time-consuming and labor-intensive activity for producers and therefore limits the feasibility of frequent body weight monitoring.

To automate the measurement of animals' biometric traits, Abdelhady et al. (2019) proposes a linear regression model for body weight estimation in which these measurements are automatically obtained from lateral images of the animals. In that study, data were collected from Barqi sheep at different physiological stages and ages, including non-pregnant, pregnant, lactating, adult, and growing animals. The proposed regression model considers

only chest girth and height measurements, as the authors state that previous studies in the literature support these traits as the most suitable characteristics for estimating sheep body weight. As a result, the model defined as $Weight = 0.0644 * breadth + 0.0100 * length.$ achieved an accuracy of 98.75% in weight estimation and an R^2 of 0.99. Despite these promising results, the animal segmentation strategy relied on the animals' color patterns due to the homogeneity of the dataset, which contained only Barqi sheep. Therefore, the proposed model cannot be generalized to animals with different characteristics.

Sant'Ana et al. (2021) proposes a CV-based approach for sheep weight prediction, based on the sequential application of image processing algorithms for extracting animal features, followed by the use of ML algorithms to estimate body weight from these features. The dataset used contains images corresponding to 32 sheep aged between 6 and 7 months, including 17 females and 15 males, with genetic backgrounds resulting from crosses between Santa Inês and Texel breeds. Due to this crossbreeding, the animals exhibit variations in size, color, and body structure (shape). Given this diversity of characteristics, which represents a challenge for fitting prediction models, they automatically extract a total of 30 attributes from the images through the application of the following techniques: *K-Curvature, Hu Moments, Area, Perimeter, Extreme Points with Euclidean distance, Aspect Ratio, Equivalent Diameter, Extent, Solidity, Major Axis, and Minor Axis*. According to the authors, these techniques enable the extraction of features representing shapes, sizes, and other attributes that may be correlated with animal weight.

Thus, the 30 extracted attributes were evaluated in a feature selection step, aiming to identify those with the highest correlation with body weight. After this stage, only 11 of the 30 attributes were selected, namely: *K-Curvature, Hu Moments, Area, Perimeter, Extreme Points with Euclidean distance, Aspect Ratio, Equivalent Diameter, Extent, Solidity, Major Axis, and Minor Axis*. In the following stage, the dataset containing only the selected attributes was used to evaluate the performance of ML algorithms for predicting animal weight. The evaluated algorithms were: Linear Regression, Support Vector Regression, K-Neighbors Regressor, Multi-layer Perceptron Regressor, Light Gradient Boosting Machine, Extreme Gradient Boost Regressor, Gradient Boosting Regression, and Random Forest Regressor. The Random Forest algorithm achieved the best performance, with an adjusted R^2 of 0.687 (± 0.09) and a Mean Absolute Error (MAE) of 3.099 kg. However, it is important to highlight that animal segmentation, which is a crucial step for isolating the animal from the background image, was performed manually. This limitation makes the proposed approach impractical for real-world scenarios, particularly in large herds and/or production systems with frequent animal entry and exit. Therefore, as pointed out by the authors themselves, there is significant potential for applying DL techniques both to perform segmentation autonomously and to evaluate new models for body weight prediction.

Marques et al. (2025) proposes an RGB video-based monitoring and weight estimation system for sheep, in which the segmentation and feature extraction modules rely on Artificial Intelligence (**AI**) techniques. For the segmentation module, a ConvNet model based on the U-Net architecture was adopted to highlight the animal in the image and remove the background. In the feature extraction module, four ConvNet models were evaluated: EfficientNetB2, DenseNet121, EfficientNetB1, and MobileNetV2. In the proposed system, these features, represented as feature maps, are used as input to the prediction module, in which seven ML algorithms were evaluated for weight estimation: Linear Regression, Random Forest, Support Vector Regression, Lasso Regression, ElasticNet, and Ridge Regression. The best result was achieved by adopting an ensemble of four ConvNet models for feature extraction and performing the prediction using Linear Regression, achieving an RMSE of 2.99 ± 0.37 and an R^2 of 0.92. Although the results are promising, processing multiple DL techniques simultaneously significantly increases the computational cost of the system, making deployment on devices with limited computational resources more challenging. By adopting the *K-fold cross-validation* technique in the experiments involving the segmentation and prediction modules, the study contributes to ensuring the generalizability of the results. However, segmentation based on RGB images tends to have limited generalization capability for animals with different coat color patterns. Furthermore, no in-depth analyses were conducted regarding the main factors influencing prediction errors, such as their relationship with animal weight or the environmental conditions under which the images were captured. Studies addressing these aspects are crucial for enabling the deployment of image-based weight estimation systems in real-world environments, avoiding the need to modify the adopted techniques whenever a new herd is introduced.

References

ABDELHADY, A. S. et al. Automatic sheep weight estimation based on k-means clustering and multiple linear regression. In: HASSANIEN, A. E. et al. (Ed.). *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2018*. Cham: Springer International Publishing, 2019. p. 546–555. ISBN 978-3-319-99010-1. Cited 2 times on pages 13 and 22.

AFRIDI, H. et al. Analyzing data modalities for cattle weight estimation using deep learning models. *Journal of Imaging*, v. 10, n. 3, 2024. ISSN 2313-433X. Cited on page 16.

AGGARWAL, C. C. *Neural Networks and Deep Learning: A Textbook*. Cham: Springer, 2018. ISBN 978-3-319-94462-3. Cited on page 12.

ATKINSON, L.; DOYLE, R. E.; JONGMAN, E. C. Temperament behaviours in individually tested sheep are not related to behaviours expressed in the presence of conspecifics. *Animals (Basel)*, Switzerland, v. 14, n. 1, jan. 2024. Cited on page 22.

CHEN, M. et al. Real-time pig weight assessment and carbon footprint monitoring based on computer vision. *Animals*, v. 15, n. 17, 2025. ISSN 2076-2615. Cited on page 13.

COMINOTTE, A. et al. Automated computer vision system to predict body weight and average daily gain in beef cattle during growing and finishing phases. *Livestock Science*, v. 232, p. 103904, 2020. ISSN 1871-1413. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1871141319310856>. Cited 2 times on pages 12 and 21.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 4. ed. [S.l.]: Pearson, 2018. ISBN 978-0133356724. Cited on page 18.

GOODFELLOW, I.; BENGICCCO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Cited 2 times on pages 13 and 19.

GURGEL, A. et al. Use of biometrics in the prediction of body weight in crossbred lambs. *Arquivo Brasileiro de Medicina Veterinária e Zootecnia*, Universidade Federal de Minas Gerais, Escola de Veterinária, v. 73, n. 1, p. 261–264, Jan 2021. ISSN 0102-0935. Disponível em: <https://doi.org/10.1590/1678-4162-12087>. Cited 2 times on pages 13 and 22.

HOU, Z. et al. Body weight estimation of beef cattle with 3d deep learning model: Pointnet++. *Computers and Electronics in Agriculture*, v. 213, p. 108184, 2023. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169923005720>. Cited on page 14.

MARQUES, J. V. M. et al. Non-invasive and low-cost approach for estimating sheep weight using computer vision. *Engineering Reports*, v. 7, n. 8, p. e70329, 2025. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/eng2.70329>. Cited 2 times on pages 13 and 24.

MECKBACH, C.; TIESMEYER, V.; TRAULSEN, I. A promising approach towards precise animal weight monitoring using convolutional neural networks. *Computers and Electronics in Agriculture*, v. 183, p. 106056, 2021. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169921000740>. Cited 2 times on pages 14 and 21.

OECD/FAO. *OECD-FAO Agricultural Outlook 2025-2034*. Rome and Paris: Food and Agriculture Organization of the United Nations and Organisation for Economic Co-operation and Development, 2025. Accessed August 2025. Cited on page 12.

SANT'ANA, D. A. et al. Weighing live sheep using computer vision techniques and regression machine learning. *Machine Learning with Applications*, v. 5, p. 100076, 2021. ISSN 2666-8270. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666827021000384>. Cited 5 times on pages 12, 13, 14, 21, and 23.

SONG, X. et al. Automated body weight prediction of dairy cows using 3-dimensional vision. *Journal of Dairy Science*, Elsevier, v. 101, n. 5, p. 4448–4459, May 2018. ISSN 0022-0302. Disponível em: <https://doi.org/10.3168/jds.2017-13094>. Cited on page 20.

SZELISKI, R. *Computer Vision: Algorithms and Applications*. 2nd. ed. [S.l.]: Springer, 2022. ISBN 978-3-031-06483-4. Cited 2 times on pages 12 and 15.

ZHANG, A. et al. *Dive into Deep Learning*. [S.l.]: Cambridge University Press, 2023. <https://D2L.ai>. Cited on page 18.

ZHANG, H. et al. Neural network-based method for contactless estimation of carcass weight from live beef images. *Computers and Electronics in Agriculture*, v. 229, p. 109830, 2025. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169924012213>. Cited 2 times on pages 12 and 14.

ZHAO, Y. et al. Review on image-based animals weight weighing. *Computers and Electronics in Agriculture*, v. 215, p. 108456, 2023. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S016816992300844X>. Cited on page 12.

3 A SYSTEMATIC LITERATURE REVIEW OF VISION-BASED ANIMAL MONITORING IOT SYSTEMS

A Systematic Literature Review of Computer Vision-based Animal Monitoring Systems

Ewerton Costa Sousa[a,c,*,1], Milena Arysa Brito Portela[b], Maria do Rosário de F. M. Ferreira[b], Ricardo de Andrade Lira Rabelo[b] and José Lindenberg Rocha Sarmento[a]

[a]Department of Animal Science, Federal University of Piauí, Campus Petrônio Portela, Teresina, 64049-550, Piauí, Brazil
[b]Department of Computer Science, Federal University of Piauí, Campus Petrônio Portela, Teresina, 64049-550, Piauí, Brazil
[c]Department of Computer Science, Federal Institute of Piauí, Campus Angical, Angical, 64410-000, Piauí, Brazil

ARTICLE INFO

Keywords:
Precision Livestock Farming (PLF)
Animal monitoring
Computer Vision (CV)
Internet of Things (IoT)
Systematic Literature Review (SLR)
Deep Learning (DL)

ABSTRACT

Animal monitoring is a fundamental component of Precision Livestock Farming (PLF), enabling the continuous assessment of animal health, welfare, and productivity. Recent advances in computer vision, machine learning, and deep learning have stimulated significant interest in CV-based animal monitoring systems. However, despite the growing number of published studies, the extent to which these solutions are ready for deployment in commercial livestock environments remains unclear. This study presents a systematic literature review of CV-based animal monitoring systems from an Internet of Things (IoT) perspective, following the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2020 guidelines. The reviewed studies were analyzed with respect to research maturity, monitoring objectives, system architecture, quality-of-service (QoS) requirements, and communication technologies. The results show that research in this field has grown considerably in recent years, with a clear trend toward the adoption of deep learning techniques. Nevertheless, most studies remain at the experimental stage and focus primarily on the development and evaluation of monitoring methods. Only a limited number of works address the complete system or consider deployment-related aspects such as computational efficiency, energy consumption, latency, communication, data management, and model maintenance. Based on the system-stage studies identified in the literature, this review characterizes the common modules of CV-based monitoring systems and discusses key architectural considerations for their design and deployment in IoT environments. The findings reveal an important gap between advances in monitoring algorithms and the requirements of real-world applications, highlighting the need for future research to adopt a more holistic, system-level perspective that supports the development of scalable and deployable CV-based animal monitoring solutions.

1. Introduction

Animal monitoring is a core activity in Precision Livestock Farming (PLF) that aims to continuously – ideally in real time – assess animal conditions to improve productivity and welfare through the early detection of abnormal situations (Marino et al., 2023). Achieving this goal requires fully automated systems capable of collecting data from animals and their environment, identifying patterns associated with animal traits and behaviors, and detecting events that require farmer intervention. These systems can collect data using various sensing devices, such as temperature sensors, Global Positioning System (GPS) receivers, microphones, and Radio Frequency Identification (RFID) tags. More recently, cameras have emerged as a promising non-contact monitoring technology. By capturing high-dimensional data, they provide richer information about animals and their environment than conventional sensors (Soro and Heinzelman, 2009).

Despite the growing body of research on Computer Vision (CV)-based animal monitoring, most proposed works remain at the experimental stage and lack the characteristics required for deployment in commercial environments (Neethirajan and Kemp, 2021). According to Li et al. (2020), the scarcity of operational systems has limited the practical advancement of the field. Although several reviews on CV-based applications have been published, limited attention has been given to the challenges involved in transitioning these solutions from experimental prototypes to deployable systems. From this perspective, CV-based animal monitoring can be viewed as an Internet of Things (IoT) application, since real-time monitoring relies on enabling technologies such as sensors, actuators, and communication networks (Akhigbe et al., 2021). Consequently, the design of fully automated monitoring systems must address challenges arising from the distributed nature of IoT environments, including constraints on energy, bandwidth, storage, and computational resources.

Therefore, this study conducts a systematic literature review (SLR) of CV-based animal monitoring systems for precision livestock applications, following the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2020 guidelines. The review provides a comprehensive overview of the field, with particular emphasis on the deployment-related aspects required to transition these systems from experimental prototypes to commercial environments. The main contributions are:

*Corresponding author
ewertoncosta@ifpi.edu.br (E.C. Sousa)
ORCID(s): 0000-0003-4695-454X (E.C. Sousa); 0000-0002-8025-8161 (M.A.B. Portela); 0009-0007-7818-9893 (M.d.R.d.F.M. Ferreira); 0000-0003-1482-6404 (R.d.A.L. Rabelo); 0000-0002-4215-1515 (J.L.R. Sarmento)

1. An overview of the maturity of existing research is provided, classifying studies as either experimental or system-stage;
2. The core modules required for fully automated CV-based animal monitoring systems are identified;
3. The quality-of-service (QoS) requirements and network technologies considered in the literature are analyzed;
4. Key architectural considerations for the design of CV-based animal monitoring systems from an IoT perspective are derived.
5. This review identifies key research gaps that limit the advancement of the proposals from experimental prototypes to deployable systems.

The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 describes the research methodology. Section 4 reports the review results. Section 5 discusses the findings and their implications for the design and deployment of CV-based animal monitoring systems. Finally, Section 6 concludes the paper and highlights future research directions.

2. Related Works

With regard to CV-based animal monitoring, other review papers and basic research have been published in recent years. However, these works do not address the deployment requirements and arrangements associated with fully automated systems, which may delay the commercial interest and adoption of such systems. Wang et al. (2022) provided a comprehensive overview of the advances in the use of intelligent sensing and artificial intelligence in precision pig farming. The authors provided an overview of the primary sensor technologies employed in pig farming, along with an examination of Artificial Intelligence (AI) based vision and sound models utilized for pig detection and tracking, behavioral recognition, and disease detection. Furthermore, they examined the potential benefits, obstacles, and constraints associated with the implementation of intelligent pig farming. Computer vision has been employed to discern pig behavior, including activities such as drinking, mounting, nursing, estrus, and aggression. The researchers have primarily employed a range of algorithms, including Faster R-CNN, Mask R-CNN, YOLO, ResNet, VGG, and others, for the purpose of recognizing pig behavior. Additionally, the findings indicate the extensive utilization of sophisticated embedded devices, RFID, RGB and thermal imaging surveillance cameras (e.g., Kinect 3D) and electret microphones (e.g., MAX4466 and LIQI LM 320E) in this type of system. This work addresses the integration of the IoT and AI, but it lacks depth regarding the interconnection and data transmission between the physical and cloud levels. The authors limit their review of the IoT aspect to data sensing and capturing, which is insufficient for a comprehensive analysis.

Wang et al. (2023b) reviewed three-dimensional CV used in livestock monitoring for smart farming, offering insights into its potential in cattle monitoring systems and covering data acquisition and analysis for cattle growth management tasks. They observed that cameras using infrared (IR) light have been widely applied due to their low cost, but they do have drawbacks regarding object-camera distance, sunlight exposure and inter-camera interference. They state that a good cattle growth management system is welcome when it offers the least possible interference in the routine of the herd while maintaining high accuracy. That way, the authors posit that to reduce the need for human intervention in cattle management, these systems should be full automated. However, they observed that there is no standard approach or method yet for automated 3D cattle growth management. They also concluded that the recognition task is an essential component of automated individual monitoring. In this context, the authors' review was limited to the automatic acquisition of data, and thus did not address the questions about the modules that should be implemented to transform these data and what kind of communication technology would be the more suitable for share data to each other.

Sundaram and Meena (2023) also conducted a review of CV-based animal monitoring systems incorporating AI for animal detection and classification. Indeed, their comprehensive literature review encompassed the remote monitoring of wild animals with the objective of mitigating human-animal conflict and animal-vehicle conflict, thereby facilitating the conservation of endangered species. While the scope of this study is considerably more wide-ranging than that of livestock monitoring, where the environment is much more controlled, data capture represents a challenge that is common to both. In order to achieve this, they undertook a review of the various image acquisition methods that could be employed, including camera traps, satellites, drones, and bio-loggers (animal-borne sensors), and also considered the different image modalities that could be used, such as visible, infrared, thermal, or a fusion of two or more. The authors also presented a variety of datasets that are available for use in animal studies and then proceeded to compare the accuracy of classification models on three benchmark datasets that contain information about animal breeds. In comparison to image processing techniques, the authors conclude that DL solutions demonstrate superior performance than Machine Learning (ML) based techniques. However, they also observed that DL models have high time complexity and system configuration requirements. As with other secondary works discussed in this section, this paper acknowledges the importance of continuous and automated monitoring of animals for the well-being of both humans and animals. The authors posit that a comprehensive animal monitoring system must incorporate animal detection, classification, and deterrence techniques. However, it also fails to acknowledge key steps towards realistic automation, which must inevitably encompass aspects such as communication between data capturing devices and data processing units. Once more, as with other works, IoT is not addressed. However, the use of terms such as "remote" and "automatic" suggests

the existence of a crucial, often overlooked communication level. This is because, from a practical standpoint, remote monitoring systems must operate in a wireless network, whether basic or complex.

Ma et al. (2024) underscore the significance of phenotypic data in livestock management, noting that manual data collection is arduous, time-consuming, dangerous, stressful, and susceptible to errors. In this context, they present an extensive review of the literature on CV-based measurement techniques for estimating the dimensions and weight of livestock. The authors review the potential of computer vision technology for automated, non-contact data collection, focusing on three key areas: 3D reconstruction, body dimension acquisition, and weight estimation. They initially examine the process of three-dimensional reconstruction of livestock, which they identify as a fundamental step in obtaining phenotypic data that will subsequently be utilized for the estimation of body dimensions and weight. Initially, the research explored the use of 2D RGB image-based approaches, which, despite being more economical and exhibiting well-established applicability, have been deemed suboptimal due to environmental sensitivities and the necessity for multiple cameras. Alternatively, laser scan-based techniques provide more accurate depth information, but are costly and are susceptible to errors due to target mobility. Ultimately, depth camera-based 3D reconstruction is identified as a more practical solution, offering real-time data capture and superior suitability for livestock environments. However, it is not without limitations, such as higher cost and sensitivity to lighting conditions. The authors conclude that PLF will evolve towards non-contact, automated, real-time, and continuous detection, facilitated by advances in smart sensors, big data, and deep learning. They predict that automated neural network models will become the primary method for collecting non-contact phenotypic data in livestock, with increasingly accurate architectures and extensive datasets becoming standard. Additionally, the authors highlight the lack of high-quality, publicly available livestock point cloud datasets as a significant challenge for future research, as large datasets are essential for training DL models in phenotypic data acquisition and weight estimation. Notably, the paper overlooks the complexity of maintaining such automated, real-time and continuous operating systems, which inevitably require the wireless integration of visual sensing devices, whether its purpose is to automatically create a large dataset or to monitor animal health.

3. Methodology

This study was carried out using the guideline developed in accordance with the PRISMA 2020 statement (Page et al., 2021). A SLR is a research method used to systematically collect, identify, and critically analyze existing studies, such as articles and dissertations. Its purpose is to provide an updated overview of current literature on a topic, highlighting key findings and suggesting areas for future research (Carrera-Rivera et al., 2022). The PRISMA 2020 statement provides updated reporting guidelines for systematic reviews, reflecting advances in methods for identifying, selecting, evaluating, and synthesizing studies and ensures that all stages of the review process are presented in an understandable and reproducible manner (Page et al., 2021). This systematic review was conducted in accordance with the PRISMA guidelines, with each step of the process and its corresponding PRISMA checklist item outlined in detail below:

1. Planning
   (a) Identifying the need for the SLR (#3)
   (b) Defining research questions (#4)
   (c) Selecting digital library sources (#6 and 7)
   (d) Defining inclusion and exclusion criteria (#5 and 8)
   (e) Defining the Quality Assessment (QA) protocol (#14 and 15)
   (f) Defining the data extraction approach (#9)
2. Conducting
   (a) Building digital library search strings (#7)
   (b) Searching and Gathering studies (#13 and 16)
   (c) Study Selection and Refinement (#16, 17 and 18)
   (d) Data extraction (#9, 10 and 12)
3. Reporting
   (a) Analysis and Synthesis (#13, 19, 20, 21 and 22)
   (b) Displaying results (#16, 17 and 19)

A preliminary overview of existing literature and related works indicated a need for a comprehensive review of CV-based animal monitoring IoT systems. This prompted the pursuit of this research project. Upon identifying this gap, the same preliminary search could be employed to formulate the research questions delineated in subsection Research Questions. Clearly defining research questions is an indispensable preliminary step in establishing the focus of the study and identifying pertinent data (Carrera-Rivera et al., 2022), as well as guiding subsequent procedures. First, all searches were conducted in the Web of Science (WoS) database, and the retrieved results were then subjected to manual filtering through the application of two distinct filters. The initial search, performed on January 1st, 2024, yielded 1,842 papers using the specified search string:

((smart farm OR livestock OR precision*) OR ((animal OR crop OR cow OR cattle OR calf OR pig OR lamb OR sheep OR goat) AND (monitor* OR surveill*))) AND ((visual OR vision OR camera OR media OR image OR video OR IoT))*

Following the application of filter 1 to each of the aforementioned works based on their titles and abstracts, only 175 met the defined requirements and were thus deemed relevant for advancement to the subsequent step. Subsequently, the second round of filtering resulted in the rejection of six works. Ultimately, 168 papers deemed relevant were identified.

In the initial stage of the filtering process, the selection criteria were applied to eliminate irrelevant papers based on a review of the titles and abstracts. Initially, only peer-reviewed journal articles or proceedings of scientific events in the field written in the English language were included. Additionally, to be selected, the work had to meet the following criteria:

1. The study must be published and fully available in scientific databases.
2. The study must be primary research.
3. The study must propose and evaluate an application for image monitoring in agricultural scenarios, whether animal or crop.
4. The study must have been approved by the scientific community and published after 2015.

Accordingly, this review excludes papers that are not fully accessible in the searched databases, as well as secondary and tertiary studies, reports, patents, technical notes, tutorials, conference workshops, books, and book chapters. Studies lacking detailed descriptions of practical experiments testing their hypotheses, and papers not written in English are also excluded. Only papers specifically addressing the extraction and/or classification of images or videos from agricultural records are considered eligible for inclusion. Studies published prior to 2015 were also excluded from the review, following the findings of Marino et al. (2023), which indicate that this date marks the introduction of deep learning applications in animal monitoring.

Following the initial filtration process, the selected works underwent a comprehensive reading and rigorous evaluation in accordance with the aforementioned criteria. Before proceeding to the next filter, due to different objectives and methodologies, the papers were divided in: Livestock=56; Wildlife=10; Crop Monitoring=102. This decision was made with the nature of the studies in mind: some of the essential steps were not necessarily present in Wildlife or crop monitoring studies, also some of the challenges are presented in a different form as the livestock ones. In the second filter, all Livestock-themed works that were deemed invalid for the purposes of this review upon being read in their entirety were excluded.

In the subsequent phase, the 57 selected papers were thoroughly reviewed, and data were systematically extracted and stored in a datasheet. A standardized data extraction form was developed to capture study characteristics from the eligible papers. During the design of this form, the research questions outlined in section Research Questions were taken into consideration. In addition to the basic information (bibliographic data, date of publication, abstract, etc.), these forms should contain a summary of the work and the answers to secondary questions, where applicable.

3.1. Research Questions

1. Do the studies propose complete IoT-based animal monitoring systems or only evaluate computer vision techniques for specific monitoring tasks (**Q1**)?

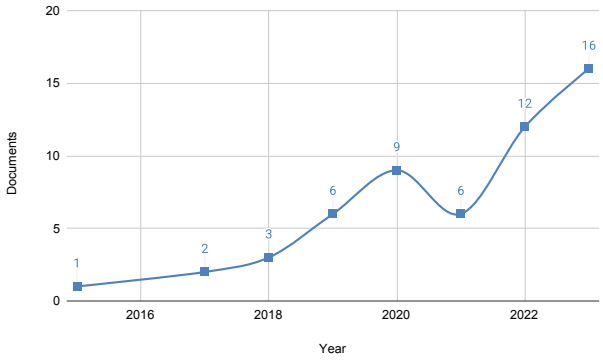


Figure 1: Number of CV-based monitoring systems related documents published per year from 2015 to 2024 and retrieved from the WoS Core Collection.

2. What are the main processing stages of automated CV-based animal monitoring systems (**Q2**)?
3. Which QoS attributes are considered when evaluating the proposed solutions (**Q3**)?
4. How is network communication incorporated into the system design, what data are transmitted, and what is its impact on system performance (**Q4**)?

4. Results

The number of publications in the CV-based animal monitoring field showed an overall upward trend from 2015 to 2023, increasing from two studies in 2015 to 16 in 2023 (Fig. 1). Research activity gained momentum in 2019 and remained relatively stable during the COVID-19 pandemic (2020–2021). From 2022 onward, publication output increased again, indicating sustained and growing interest in the field. The three most investigated monitoring tasks are behavior classification (e.g., walking, resting, feeding, and drinking), body or weight measurement, and animal tracking (Fig. 2). Recent studies have increasingly adopted DL techniques (Table 2), replacing more traditional approaches. This shift is largely driven by the ability of DL models to automatically learn representations from data, identify complex patterns, and make accurate predictions with minimal human intervention (Khalil et al., 2021).

4.1. Q1 - Research maturity

In this review, experimental studies are those that propose and evaluate an animal monitoring method, technique, or algorithm without addressing the additional components required for automated deployment, such as image acquisition, method execution, and the delivery of results and insights to farmers. Their focus was to evaluate the performance of a monitoring method, generally in terms of accuracy, do not taking account how this method will be deployed in a real scenario. In contrast, studies that describe how these components would operate, even if not fully implemented or detailed, were classified as being at the system stage.

Table 1
Behaviour Monitored

Monitoring Task	Works
Animal counting	Koh et al. (2023); Cakic et al. (2023); Xu et al. (2020); Han et al. (2019); Meena and Agilandeeswari (2021)
Animal detection	Wang et al. (2018); Čakić et al. (2022); Li et al. (2019); Son et al. (2022)
Animal recognition	Zhang et al. (2023b); Kounalakis et al. (2019)
Animal tracking	Zhi et al. (2019); Salau and Krieter (2020); Bonneau et al. (2022); Su et al. (2021); Nilsson et al. (2015) Guo et al. (2023a); Prior et al. (2023)
Behavior classification	Guo et al. (2020); Ahmed et al. (2024); Biglari and Tang (2022); Tsai et al. (2020); Lynge et al. (2022) Guo et al. (2021); Qiao et al. (2022a); Han et al. (2023); Schmidt et al. (2022); Saitoh and Kato (2021) Sumi et al. (2021); Achour et al. (2020); Yujuan et al. (2017); Fernández-Carrión et al. (2017) Zhuang et al. (2023); Guo et al. (2023b)
Body or weight measurement	Du et al. (2022); Zhang et al. (2018); Le Cozler et al. (2019); McVey et al. (2022); Lee et al. (2019) Xi et al. (2023); Xavier et al. (2022)
Body temperature	Goyal et al. (2021)
Digestive health	Atkinson et al. (2020)
Disease progression	Pessanha et al. (2020)
Environment monitoring	Ding et al. (2023); Liu et al. (2023)
Estrus detection	Wang et al. (2023c); Arago et al. (2022)
Image segmentation	Zhang et al. (2023a); Qiao et al. (2022b)
Kinship assignment	Szymanski and Lee (2020)
Lameness detection	Fernández-Carrión et al. (2020); Van Hertem et al. (2018); Anagnostopoulos et al. (2023)
Respiration or heart rate	Wang et al. (2023a, 2021); Wu et al. (2023); Jorquera-Chavez et al. (2021)
Tail-biting	Hakansson and Jensen (2023)

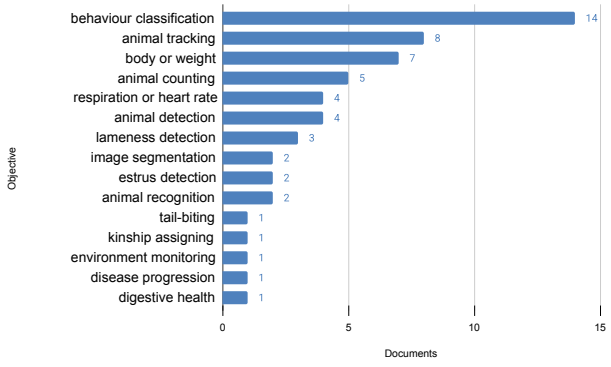


Figure 2: Number of CV-based monitoring systems related documents published per monitoring objective from 2015 to 2024 retrieved from the WoS Core Collection.

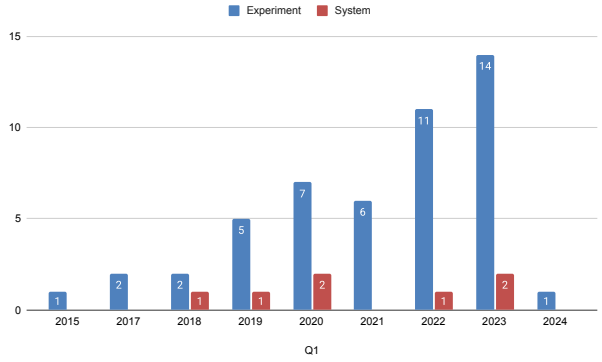


Figure 3: Number of CV-based animal monitoring related documents per publication year grouped by research stage

Table 2
CV-based animal monitoring works grouped by the approach of monitoring method

Monitoring Approach	**Count**
Digital Image Processing	8
Deep Learning	31
Digital Image Processing and Deep Learning	12
Other Approaches	8

Regarding research maturity, only seven studies were classified as being at the system stage, whereas 49 remained at the experimental stage. That is, approximately 88% of the found works are still in the experimental stage and are not capable of being implemented in commercial environments.

Furthermore, the observed growth in annual publication output was driven predominantly by experimental studies (Fig. 3), indicating that most research efforts remain focused on developing and validating monitoring methods rather than advancing them toward deployable systems. Achour et al. (2020) employed four Convolutional Neural Networks (ConvNet) for monitoring dairy cows: (1) detecting their presence in the feeder zone, (2) determining their position (standing or feeding), (3) assessing food availability and category, and (4) identifying individual cows. Additionally, it explores the use of a ConvNet combined with an Support Vector Machine (SVM) and the fusion of multiple ConvNets to enhance individual identification. Tsai et al. (2020) proposed a method for monitoring drinking behavior in dairy cows to identify heat stress. The study presented a node

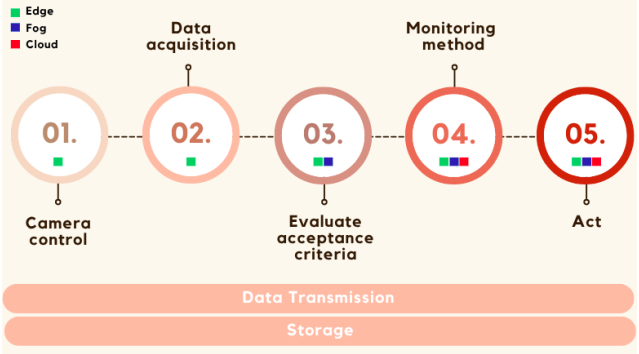


Figure 4: Common main modules of automatic CV-based animal monitoring systems.

for image collection with TinyYOLO processing for animal detection, information transmission via WiFi, server storage and web view. Zhuang et al. (2023) presented a system to monitor feeding and drinking behavior of growing pigs based on Machine Vision, as well as physical devices with cameras attached to serve water and food to the animals. Lee et al. (2019) proposed a method for monitoring undergrown pigs, using TinyYOLO and bounding boxes for detection of animals in the scene. As for automation triggers, the camera is on nonstop and frames are evaluated "constantly". When a movement threshold is met, the detection process starts. Xi et al. (2023) experimented with Mask R-CNN, Cascade Mask R-CNN and HTC for prawn farm management, understanding growth and distribution of the animals. They also implemented a client-side viewfinder, as well as a streaming/recording server and a form of power supply for the node. Arago et al. (2022) implemented a estrus detection method, based on standing-heat/mounting behavior, using Faster R-CNN and presenting the collection (hardware for data), processing (software) and application (output for users) layers. Van Hertem et al. (2018) proposed a cow locomotion monitoring based on 3D vision in a commercial farm, employing Kinect V2 to capture RGB, IR e depth images as well as RFID for presence detection and identification.

4.2. Q2 - Common modules of CV-based animal monitoring systems

Based on the seven studies classified as system-stage research, five core modules of CV-based animal monitoring systems were identified (Fig. 4): i) Camera control, ii) Data acquisition, iii) Evaluate acceptance criteria, iv) Monitoring method, and v) Act.

4.2.1. Camera control

This module aims to manage camera operation by switching the sensor between active and inactive states. In studies that implemented this module, cameras were activated when an animal was detected and deactivated when no animal was present. For example, Van Hertem et al. (2018) employed RFID technology for animal detection, whereas Xi et al. (2023) controlled camera operation through user-issued commands. However, implementation of this module is not mandatory, since several studies, including Lee et al. (2019); Tsai et al. (2020); Zhuang et al. (2023); Achour et al. (2020); Arago et al. (2022), operated cameras continuously.

4.2.2. Data acquisition

This module is responsible for acquiring and processing data from the camera sensor. When active, cameras generate a continuous stream of data, requiring real-time frame acquisition and processing. Frames are captured at a predefined rate, measured in frames per second (FPS), which is determined according to system requirements. As such, this module typically serves as the entry point of CV-based animal monitoring systems. In most studies, images are acquired directly from camera sensors connected to the processing hardware through interfaces such as USB or serial communication. Alternative acquisition strategies have also been reported. For example, Arago et al. (2022) employed a PTZ camera to record live video of cattle and transmit the footage to a Network Video Recorder (NVR) via Ethernet. The captured media were stored on the NVR, and the monitoring system retrieved video frames through Application Programming Interface (API) requests.

4.2.3. Evaluate acceptance criteria

This module evaluates whether the acquired images satisfy the conditions required by the target monitoring task (e.g., weight prediction or lameness detection). Images that do not meet these conditions are discarded and not forwarded to subsequent modules. This step is essential for reducing selective bias and ensuring that the data accurately represent the monitored characteristics (Khalil et al., 2021). For example, images without animals, blurred images, or highly redundant frames may degrade model performance and increase the risk of overfitting during training.

Several studies have implemented automated data-quality assessment procedures. Lee et al. (2019) applied TinyYOLO to detect individual pigs and used image-processing techniques to retain only images containing moving animals with adequate posture and separability, both critical for accurate body-size estimation. Meckbach et al. (2021) employed two complementary strategies: i) the Laplacian operator to identify blurred images that could compromise weight estimation performance, and ii) the Structural Similarity Index Measure (SSIM) to detect highly similar consecutive frames and mitigate data redundancy. Similarly, Van Hertem et al. (2018) reduced the number of processed frames by 83% through an automated filtering procedure that excluded videos containing multiple cows, an insufficient number of frames for analysis, or irregular gait patterns, such as stopping or running.

4.2.4. Monitoring method

Although the monitoring objectives identified in this review are diverse (Fig. 2 and Table 1), most CV-based monitoring methods are founded on the premise that animal characteristics (e.g., body measurements and facial features) and behaviors (e.g., feeding activity, posture, and estrus)

can be inferred from image data. To achieve this, relevant attributes are extracted from images and subsequently analyzed using computational techniques capable of identifying patterns associated with the target characteristic or behavior.

Like the overall architecture of CV-based monitoring systems, monitoring methods are typically organized as a sequence of processing stages in which data undergo successive transformations. For example, Wang et al. (2023a) proposed an RGB video-based system for monitoring the respiration rate (RR) of group-housed pigs that consists of three stages: animal detection, region-of-interest (ROI) extraction, and RR estimation. Across the studies reviewed, these processing pipelines generally comprise three main tasks: object detection, which determines the animal's location; image segmentation, which isolates the animal from the background; and data reasoning, which estimates or classifies the target outcome.

The reviewed approaches employ a wide range of techniques, with Digital Image Processing (DIP) and DL methods often used either independently or in combination (Table 2). A clear trend toward the adoption of DL, particularly ConvNet, can be observed, reflecting their growing prominence in the implementation of animal monitoring methods.

4.2.5. Act

Monitoring systems must be capable of rapidly identifying abnormal situations and either notifying producers or triggering corrective actions. The purpose of this module is to define the actions performed after the execution of the monitoring method, particularly when anomalies are detected. Among the seven studies that proposed complete CV-based animal monitoring systems, the post-processing actions included: i) storing results in a database for visualization and future consultation (Tsai et al., 2020); ii) displaying results on local monitors at the deployment site (Zhuang et al., 2023; Xi et al., 2023); and iii) sending notifications to responsible personnel (Le Cozler et al., 2019; Arago et al., 2022). In contrast, (Achour et al., 2020; Van Hertem et al., 2018) did not specify any post-processing actions following method execution.

Although the reviewed systems primarily focused on data storage, visualization, and notifications, this module can also support automated interventions in the production environment. Examples include controlling fan operation in response to detected heat stress, opening or closing gates based on animal status, and activating other actuators to mitigate identified issues.

4.3. Experimental studies categories

The studies classified as experimental-stage research primarily propose and evaluate computational models intended for use within the Monitoring Methods module of future CV-based monitoring systems. Since this module may itself comprise multiple processing tasks, the reviewed studies are organized into three categories: animal detection, animal segmentation, and behavior classification.

4.3.1. Animal detection

This category comprises studies focused on the development and evaluation of computational models for animal detection and localization in images. Animal detection is typically the first stage of a CV-based monitoring method, as it identifies the presence and location of animals or body parts within a scene. The resulting spatial information can then be used by subsequent processing stages, such as animal segmentation, which isolates the animal from the background, and data reasoning, which estimates or classifies the target characteristic or behavior. Depending on the adopted architecture, these tasks may be implemented as separate models arranged in a processing pipeline or integrated into a single end-to-end model.

Proposed approaches range from traditional computer vision techniques, such as Template Matching (Li et al., 2019), to modern DL models, including TinyYOLOv4 (Son et al., 2022), Fast R-CNN (Wang et al., 2018), and Faster R-CNN-based architectures (Čakić et al., 2022). Some studies further integrated detection models with IoT technologies to support applications such as the identification of dead poultry and uneven growth patterns (Čakić et al., 2022).

Animal counting was addressed by several studies using DL techniques (Han et al., 2019; Meena and Agilandeeswari, 2021; Xu et al., 2020). Other applications included weight and growth assessment (Cakic et al., 2023), handling occlusions and 3D animal orientation (Koh et al., 2023), sheep identification through facial recognition using Vision Transformers (ViT) (Zhang et al., 2023b), weed recognition and control in grazing environments (Kounalakis et al., 2019), and cattle digestive health monitoring from fecal images using ResNet-101 (Atkinson et al., 2020).

Animal tracking also emerged as a prominent application. Approaches included the use of VIAME software for fish tracking (Prior et al., 2023), YOLO-based methods for monitoring goat grazing behavior (Bonneau et al., 2022), individual pig tracking with YOLOv5s (Guo et al., 2023a), and dairy goat tracking using an augmented backbone network combined with a Siamese architecture (Su et al., 2021). Additional studies investigated cow space-use patterns through adaptive background subtraction (Salau and Krieter, 2020), video summarization techniques based on K-means clustering and color features (Zhi et al., 2019), fan-operation monitoring in livestock houses using the Hough transform (Ding et al., 2023), and multi-cattle tracking through deep learning models (Han et al., 2023).

Several detection-based approaches were also developed to support behavior and welfare monitoring. These included monitoring water intake in cattle (Biglari and Tang, 2022), detecting slowdown and weakness in pigs through optical flow analysis (Fernández-Carrión et al., 2017), assessing cow welfare (Guo et al., 2023b), monitoring rumination behavior using Faster R-CNN (Yujuan et al., 2017), and detecting tail-biting behavior in pigs with pre-trained ConvNets (Hakansson and Jensen, 2023). Other applications focused on lameness detection in cattle (Anagnostopoulos et al., 2023), estrus monitoring from thermal infrared

imagery (Wang et al., 2023c), and lethargy detection in pigs using binary image segmentation and Kalman filtering (Fernández-Carrión et al., 2020).

4.3.2. Animal segmentation

Animal segmentation methods aim to isolate animals or regions of interest from the background at the pixel level, providing detailed spatial information for subsequent monitoring tasks. These methods are often applied after animal detection, although both tasks may be integrated within a single model. Among the reviewed studies, Nilsson et al. (2015) proposed a learning-based segmentation method for estimating pig location and extracting body-related features, Qiao et al. (2022b) presented a one-shot learning approach with pseudo-labeling for cattle video segmentation, Wu et al. (2023) employed a YOLACT-based model for monitoring cattle respiratory behavior, and Zhang et al. (2023a) proposed a DeepLabv3-based network for dairy goat image segmentation.

4.3.3. Behavior classification

Studies were categorized as behavior classification when the primary objective was to infer, estimate, or classify an animal characteristic or behavior. In these approaches, tasks such as animal detection, tracking, and feature extraction serve as intermediate processing steps, while the final output corresponds to the target monitoring outcome.

Several studies focused on the classification of animal behaviors. For cattle, Sumi et al. (2021) and Guo et al. (2020) combined DIP techniques with behavior recognition to classify activities such as entering or leaving an area, resting, feeding, drinking, and calving. Similarly, Qiao et al. (2022a) employed a ConvNet to identify feeding, exploring, grooming, walking, and standing behaviors, while Guo et al. (2021) used a BiGRU-based model to classify the same set of activities. In poultry farming, Ahmed et al. (2024) applied random forest and decision tree algorithms to classify pecking, preening, and dustbathing behaviors. Other studies addressed posture and activity recognition, including the classification of finishing pig postures using ResNet-18 (Lynge et al., 2022) and the identification of lying, standing, walking, distance traveled, ATF, and ATW behaviors through height and speed thresholds (Schmidt et al., 2022).

Beyond behavior recognition, several studies focused on estimating physiological and morphological traits from image data. These included livestock body measurement using keypoint detection and depth cameras (Du et al., 2022), estimation of cattle body traits using the Morpho3D acquisition system (Le Cozler et al., 2019), biometric measurement through morphometric analysis (McVey et al., 2022), and sheep body-size monitoring using image enhancement and filtering techniques (Zhang et al., 2018). More specialized applications included heart-rate monitoring (Wang et al., 2021) and respiration-rate estimation (Wang et al., 2023a) in pigs.

A smaller group of studies focused on prediction tasks, in which image-derived features were used to estimate future outcomes, detect conditions at an early stage, or infer genetic relationships. Examples include the early detection of respiratory diseases in pigs through CV-based monitoring (Jorquera-Chavez et al., 2021), prediction of disease progression stages in sheep from facial pain indicators (Pessanha et al., 2020), parent-offspring matching using ConvNets (Szymanski and Lee, 2020), and body-weight prediction in Holstein cows using the Morpho3D system (Xavier et al., 2022).

4.4. Q3 - Data transmission strategies

Only four studies explicitly considered network communication and provided details on its implementation, all of which were among the seven system-stage studies. Several other works mentioned that data could be transmitted over a network for storage, visualization, or further processing, but did not describe the underlying communication architecture. Among the studies that detailed network operation, Achour et al. (2020) and Arago et al. (2022) acquired data on nodes separate from those responsible for processing. In Achour et al. (2020), animal images were captured and filtered locally before being transmitted via a WLAN for processing. In Arago et al. (2022), images were retrieved from a DVR through an API and processed locally for estrus detection. When estrus was identified, the system sent a signal to an Arduino microcontroller equipped with a Global System for Mobile Communications (GSM) module, which initiated a phone call to notify farmers. The system also updated a cloud database through a Secure Shell (SSH) connection and made the results available through a web application.

In contrast, Tsai et al. (2020) performed both data acquisition and processing on the same device, where the camera was connected through a serial interface. Images and environmental data were then transmitted via a wireless router to a remote server and stored in a MySQL database. Xi et al. (2023), in turn, acquired prawn images using a camera connected to a Raspberry Pi. In this case, network communication was used primarily to support user interaction. Commands to start and stop video recording were transmitted between a Google Glass device and the Raspberry Pi over a local network. The processed frames were subsequently sent back to the Google Glass for real-time visualization.

4.5. Q4 - Common QoS attributes evaluated

All studies included in this review evaluated monitoring methods in terms of predictive performance, typically using metrics such as Precision, Recall, Accuracy, F1-score, and mean Average Precision (mAP), depending on the task. For image segmentation tasks, Intersection over Union (IoU) was the most commonly adopted evaluation metric. However, only a limited number of studies assessed computational efficiency, despite its importance for real-world deployment.

Koh et al. (2023) proposed Aqua3DNet, a fish-orientation estimation pipeline that integrates YOLOv3, Simple Online

and Real-Time Tracking (SORT), and UDepth algorithms. In addition to evaluating predictive performance, the authors reported processing speeds of 5.12 FPS on a GPU and 0.52 FPS on a CPU. Although the study concluded that the system could operate in real time on consumer-grade hardware, including single-board computers (SBC) such as the Raspberry Pi, the experiments were conducted on an Intel i7–1165G7 processor and an NVIDIA GTX 1060 GPU. Consequently, the feasibility of deploying the system on more resource-constrained devices remains uncertain and requires further validation. Similarly, Wang et al. (2023a) evaluated the computational efficiency of an RGB video-based respiration-rate monitoring system for group-housed pigs. The region-of-interest extraction stage achieved an inference speed of 11.62 FPS, while the respiration-rate estimation stage reached 283.88 FPS. Performance was measured using a single NVIDIA GeForce RTX 3090 GPU.

Overall, the limited reporting of computational metrics highlights a gap in the literature, as deployment feasibility depends not only on predictive performance but also on the ability to meet the computational constraints of real-world livestock production environments.

5. Discussion

Through a systematic literature review conducted in accordance with the PRISMA 2020 guidelines, this study aims to assess the state of the art in CV-based animal monitoring field and the efforts toward deploying these systems in commercial environments. The reviewed studies were analyzed with respect to key deployment-related aspects, including research maturity, core modules, QoS requirements, and the network technologies employed in the proposals.

The majority of research found in this review are in experimental stage, approximately 88% of the works, despite the growing amount of published research year by year. Although there are limitations in implementing these systems in commercial settings, existing enabling technologies are continually advancing, with the hope of being more widely utilized in future commercial application. To advance to the system stage, the future research should cover all modules necessary for automated execution of these systems, addressing how to overcome obstacles related to the integration of IoT and AI technologies. Developing IoT systems in itself already requires knowledge about different areas of research, including network topology, sensing, processing, communication and resource utilization (Hengstler et al., 2007).

5.1. System design considerations

CV-based monitoring systems typically consist of multiple components, or modules, organized into an automated processing pipeline, with each module responsible for a specific function. Their implementation becomes particularly challenging in IoT environments due to the distributed nature of the architecture, where modules may be deployed across the Edge, Fog, and Cloud layers. Consequently, the design and development of these systems require expertise spanning multiple domains, including sensing, data processing, communication, network architecture, and resource management (Hengstler et al., 2007). The central challenge is to balance monitoring performance against IoT constraints such as limited computational resources, energy consumption, bandwidth availability, and latency requirements.

In this context, Sundaram and Meena (2023) identified several factors relevant to the development of animal monitoring systems, including image acquisition strategies, data types, imaging modalities, photogrammetry techniques, and the use of unmanned vehicles for data collection. While these factors are important, they are not sufficient for designing automated CV-based monitoring systems in IoT environments. Developers must also define the system architecture, including its constituent modules, communication mechanisms, deployment across Edge, Fog, and Cloud layers, and the impact of architectural decisions on QoS attributes such as energy consumption, computational efficiency, memory usage, and latency (Bass et al., 2012; Brown and Wilson, 2011).

The following sections discusses key architectural considerations for addressing these requirements. The proposed considerations are derived from the studies reviewed in this work and the common system modules identified in Section Q2 - Common modules of CV-based animal monitoring systems. For each design decision, its implications for energy, computational, and memory resource consumption, as well as system latency, are analyzed.

5.1.1. Controlling the camera sensor status

"Should the camera sensor be continuously active, or should its operation be dynamically controlled?"

Developers must decide how camera operation will be managed, namely whether the sensor will remain continuously active or switch between active and inactive states. This decision is strongly influenced by energy availability. In environments where power is not a limiting factor, cameras may operate continuously, as reported by Lee et al. (2019); Tsai et al. (2020); Zhuang et al. (2023); Achour et al. (2020); Arago et al. (2022). Conversely, in energy-constrained scenarios, keeping the camera permanently active can lead to unnecessary power consumption. In such cases, energy-efficient strategies may be adopted, activating the camera only when required. Typically, the camera is turned on when an animal is detected and turned off when no animal is present. For example, Van Hertem et al. (2018) used RFID technology to detect animal presence, whereas Xi et al. (2023) controlled camera operation through user-triggered commands.

This decision is also influenced by the storage and computational resources available at the sensor node. A continuously active camera generates a constant stream of data, increasing memory usage and processing demands. Consequently, camera-management strategies affect not only the performance and energy consumption of the sensor node but

also the latency and resource utilization of the system as a whole, as discussed in the following sections.

5.1.2. Acquiring data

"How are camera data acquired for processing?"

Data acquisition and processing are key design considerations in CV-based animal monitoring IoT systems. When active, cameras generate a continuous stream of data, requiring mechanisms for real-time frame acquisition and processing. Frames are typically captured at a predefined rate, measured in FPS, which must be selected according to the monitoring objective. Tasks that rely on temporal information and object tracking across consecutive frames generally require higher acquisition rates. Examples include gait analysis (Van Hertem et al., 2018; Anagnostopoulos et al., 2023), respiration- and heart-rate monitoring (Wang et al., 2023a, 2021; Jorquera-Chavez et al., 2021; Wu et al., 2023), and fish-orientation estimation, for which Koh et al. (2023) adopted a rate of 60 FPS. In contrast, monitoring methods based on still-image analysis can operate at substantially lower frame rates. For example, Schmidt et al. (2022) and Tsai et al. (2020) captured images at 5 FPS to monitor nursery pig behaviors and dairy cow drinking behavior, respectively, while Achour et al. (2020) used 1 FPS for dairy cow identification and feeding-behavior monitoring. Therefore, the frame acquisition rate should be defined according to the temporal requirements of the target monitoring task.

As the data acquisition module typically serves as the entry point of the monitoring system, every captured frame undergoes some form of processing. Consequently, increasing the acquisition rate directly increases computational workload, memory usage, energy consumption, and communication demands. In resource-constrained environments, these costs can be mitigated by carefully selecting the acquisition frequency and adopting strategies that dynamically control camera operation, as discussed in the previous section. By activating the camera only when animals are present, the system can significantly reduce the volume of data that must be acquired and processed (Van Hertem et al., 2018; Xi et al., 2023).

Another important architectural decision concerns how image data are delivered for processing. In most studies, images are acquired directly from cameras connected to the processing hardware through interfaces such as USB or serial communication. However, alternative architectures have also been explored. For example, Arago et al. (2022) used a PTZ camera to record live video of cattle and transmit the footage via Ethernet to a NVR, where the data were stored. The monitoring system subsequently retrieved video frames through API requests to the NVR. Although this approach introduces additional communication overhead and latency, it enables centralized storage and decouples image acquisition from processing, providing greater flexibility for distributed IoT deployments.

5.1.3. Evaluating acceptance conditions of images

"All acquired frames must be processed by our animal monitoring method or we should filter apropriated frames?"

CV-based IoT systems can generate a large amount of data, depending on how frequently images are captured and processed, which can lead to very high consumption of computational, memory and power resources, as well network resources if the modules are distributed over different IoT layers, potentially impacting in the latency of the system. However, many of theses collected images may not present the necessary elements for the learning of the proposed monitoring models. For instance, imagens that do not contains animal in scene, that are blured, or have great similarity to another will not be useful in the learning or in the inference stage causing decrease on model performance or overfitting.

In this context, the developers must think about strategies that locally evaluate if the acquired data has the acceptable conditions to contribute with the proposed animal monitoring method (weight prediction, lameness detection, etc). Images that don't present the acceptable condition must be discarded do not forwarding them to next modules cause it is vital to avoid data that causes selective bias and to select data that truly represents the characteristics monitored (Khalil et al., 2021). Lee et al. (2019) applied TinyYOLO to detect individual pigs and image processing techniques to select only images that include moving pigs with acceptable posture and separability, characteristics that influence the computation of pig's size. Meckbach et al. (2021) adopted two strategies to evaluate acceptable conditions of images: i) used Laplacian Operator to detect imagens that present blur noise that could impact the performance of animal weight monitoring and ii) SSIM to measure the similarity of subsequent frames that can result in overstimation issue. Van Hertem et al. (2018) adopted an automated filtering procedure removing the videos that contain (a) multiple cows in the video, (b) not enough frames for analysis and (c) an irregular cow gait (stop or run).

In addition to the positive impact in the model accuracy, this image selection efforts can also contribute to reduce the data transmitted over network in scenarios where the monitoring model run in distributed cloud servers by influencing the lattency of the whole system. In Van Hertem et al. (2018), they achieve a reduction of 83% in frames processed by the monitored models, that is, this amount of data will not be processed by the model neither sent over network contributing with the system lattency by avoiding indiscriminate sending of data to a cloud processing center. However, how this step generally runs on Edge node it can compromise the consumption of energy, computational and memory resources. Therefore, acceptance criteria evaluation methods should be more lightweight and the developers should balance the trade-off between amount of data acquired and the quality of this data cause it is related to the performance in terms of acuraccy of the model, the lattency efficiency and the consumption of energy, computation and storage resources.

5.1.4. Implementing the monitoring method

"Which monitoring approach should be adopted and where should it be deployed?"

Monitoring methods typically comprise a sequence of tasks, including object detection, image segmentation, and data reasoning. These methods employ a wide range of DIP, ML, and DL techniques, either independently or in combination (Table 2). The number of processing stages and the techniques selected directly affect the complexity of the monitoring method and, consequently, its energy consumption, computational requirements, memory usage, and latency.

Therefore, selecting the architecture of the monitoring method is a critical design decision that should consider not only predictive performance but also the resources available on the target device and the latency requirements of the application. Nevertheless, most studies focused exclusively on predictive accuracy, with only a small subset evaluating computational efficiency or resource consumption. Moreover, studies that reported resource-related metrics typically assessed them in isolation, without considering their impact on the performance of the complete monitoring system.

As shown in Table 2, there is a clear trend toward the adoption of DL methods, particularly ConvNet. Although these approaches often achieve superior predictive performance, they generally require large volumes of training data and tend to impose higher computational and memory demands during inference. Consequently, the choice of monitoring approach should also consider factors such as dataset availability, image quality, data variability, and the computational capabilities of the deployment platform.

From a deployment perspective, the studies reviewed predominantly executed monitoring methods at the Edge, thereby avoiding the need to transmit image data over the network. While this approach reduces communication latency and bandwidth consumption, it requires either lightweight models or sufficiently capable hardware. An alternative is to offload processing to Fog or Cloud resources, where more computationally intensive models can be executed. However, this strategy introduces network communication overhead and may increase end-to-end latency, making the trade-off between processing capacity and responsiveness a key design consideration. A promising alternative is the adoption of hybrid architectures. For example, Fouda et al. (2022) proposed a hierarchical artificial intelligence framework for forest-fire detection in which a lightweight model performs an initial assessment and delegates uncertain cases to a more sophisticated model. A similar strategy could be applied to animal monitoring systems, with lightweight models deployed at the Edge for rapid screening and more complex models hosted in the Fog or Cloud to handle computationally demanding analyses. Such an approach has the potential to balance predictive performance, resource consumption, and latency requirements.

5.1.5. Post-processing actions

"What should be done with the monitoring results?"

The developers should consider the actions to be performed after the execution of the monitoring method, particularly when abnormal situations are detected. Among the seven studies that proposed complete monitoring systems, post-processing actions included: i) storing results in a database for visualization and future consultation (Tsai et al., 2020); ii) displaying results on local monitors at the deployment site (Zhuang et al., 2023; Xi et al., 2023); and iii) sending notifications to responsible personnel (Le Cozler et al., 2019; Arago et al., 2022). In contrast, (Achour et al., 2020; Van Hertem et al., 2018) did not specify any actions following monitoring execution.

When results must be communicated to remote users, the choice of communication technology becomes a critical design factor, particularly in rural environments where connectivity may be limited. Although the reviewed systems primarily focused on data storage, visualization, and notifications, automated interventions can also be implemented. Examples include controlling fan operation in response to detected heat stress, opening or closing gates according to animal status, or activating other actuators to mitigate identified issues. The selected response strategy should consider not only technical feasibility but also timeliness. In practice, system effectiveness depends on how quickly the generated information leads to a corrective action, whether through a notification reaching the responsible person or through an automated intervention performed directly by the system.

5.1.6. Defining communication strategies

"Where should system modules be deployed and how should they communicate?"

As discussed previously, CV-based animal monitoring IoT systems are typically composed of multiple modules organized in a processing pipeline. Given the distributed nature of IoT architectures, these modules may be deployed across the Edge, Fog, and Cloud layers, making communication design a critical aspect of system development. Communication between modules can be implemented through a variety of mechanisms, ranging from local method calls, when modules execute on the same device and share a common code base, to network-based communication when modules are distributed across different devices or infrastructures. In the latter case, communication may rely on technologies such as Wi-Fi, Ethernet, ZigBee, cellular networks, or other IoT communication protocols. Consequently, the choice of communication strategy directly influences system latency, bandwidth consumption, energy usage, scalability, and reliability.

The selection of a communication technology depends on the application context, deployment constraints, and the characteristics of the data being exchanged. For example,

transmitting raw video streams typically requires substantially more bandwidth than transmitting filtered frames, extracted features, or monitoring results. Among the studies reviewed, most adopted an on-device processing strategy, in which communication occurred through local method calls and data exchange within the same hardware platform. However, some studies explored distributed communication architectures.

For instance, Xi et al. (2023) proposed a smart-headset and computer vision system for automatic prawn-size measurement in which a Google Glass application, acting as the user interaction module (Camera control), communicated with a Raspberry Pi over a Wi-Fi connection to control image acquisition (Data acquisition). Similarly, Arago et al. (2022) implemented a distributed acquisition architecture in which a PTZ camera transmitted live video of cattle to a NVR through an Ethernet connection. The monitoring system subsequently retrieved the stored video data by issuing API requests to the NVR, from which the frames required by the monitoring module were extracted and processed. These examples illustrate that module deployment and communication decisions are closely related. While distributing modules across multiple devices can improve scalability and facilitate the use of specialized hardware, it also introduces communication overhead and additional latency that must be carefully considered during system design.

5.1.7. Planning for adequate storage solutions

"What data should be stored, where should they be stored, and how should they be managed?"

Because images are high-dimensional data, they require substantial memory for storage and significant computational resources for processing. Consequently, developers must carefully plan storage and memory management to ensure that sufficient resources are available throughout the monitoring pipeline. This planning should consider factors such as image quality (e.g., resolution and sensor type), acquisition rate, the number of processing stages, and the computational requirements of the monitoring method. For example, Lee et al. (2019) reported that Mask R-CNN, one of the most widely used DL-based instance segmentation models, could not be executed on an NVIDIA TX-2 due to memory limitations. Therefore, the design decisions discussed in previous sections—such as camera operation, frame acquisition frequency, data filtering, and monitoring method selection—must also be evaluated from a storage and memory perspective.

Another important consideration is whether and how data should be persistently stored. Long-term storage can support historical analyses, result visualization, auditing, and the retraining or refinement of monitoring models. Consequently, developers must determine where data will be stored and how storage resources will be managed. When data are stored at the Edge, limited storage capacity may require mechanisms to monitor disk usage and remove obsolete data. For example, Xi et al. (2023) implemented a basic storage management strategy that monitored storage utilization and automatically deleted old recordings, while Arago et al. (2022) relied on the storage management capabilities provided by a NVR.

Alternatively, data may be transmitted to Fog or Cloud infrastructures for storage and processing. Although this approach alleviates local storage constraints, it introduces communication overhead and may increase system latency. To reduce these impacts, developers may choose to transmit only the outputs of image processing, such as extracted features, metadata, monitoring results, or model updates, rather than raw images or video streams. Besides reducing bandwidth consumption, this strategy can also support distributed learning approaches, such as federated learning, in which models are improved collaboratively without requiring the transfer of large volumes of image data.

5.1.8. Animal identification

"How should acquired data be associated with individual animals?"

CV-based monitoring systems must have the ability to identify and recognize individual animals, allowing the association and tracking of relevant features for an individual over time. In order to bind acquired data and the characteristics or behaviors extracted from images to an individual animal, is mandatory to implement an strategy for animal identification. Qiao et al. (2021) classify the identification methods in ear-tag based aproachs, like traditional plastic and RFID tags, biometric and visual feature-based, and DL-based approaches. Therefore, developers should define the more suitable identification technology. Only Van Hertem et al. (2018) and Achour et al. (2020) explicitly presented an strategy for animal identification. While the former adopted a RFID technology, the last adopted an ConvNet model that recognize the animal by features extracted from the head.

5.1.9. Deploying, retraining and syncing models

"How should models be trained, synchronized, and updated?"

Model updating remains a challenge that must be addressed and is largely overlooked by current studies, which predominantly adopt standalone on-device processing strategies without network connectivity. None of the reviewed works discussed how deployed models could be synchronized, updated, or retrained over time, highlighting an important research gap for future work.

6. Conclusion

Animal monitoring is a fundamental component of PLF, and recent advances in computer vision, machine learning, and deep learning have driven substantial growth in research on CV-based monitoring systems. Through a systematic literature review conducted according to the PRISMA 2020 guidelines, this study assessed the state of the art

in CV-based animal monitoring from an IoT perspective, examining research maturity, system architecture, QoS requirements, and communication technologies. The results indicate that, despite the increasing number of publications, most studies remain at the experimental stage and focus primarily on improving monitoring accuracy, while critical deployment-related aspects such as computational efficiency, energy consumption, latency, communication, data management, and model maintenance receive limited attention. To address this gap, the review identified the common modules of CV-based monitoring systems and discussed key architectural considerations for their design and deployment in commercial environments. These findings provide a foundation for future research and development efforts, highlighting the need to move beyond algorithm-centric evaluations toward system-level approaches that consider the constraints and requirements of real-world livestock production systems.

References

Achour, B., Belkadi, M., Filali, I., Laghrouche, M., Lahdir, M., 2020. Image analysis for individual identification and feeding behaviour monitoring of dairy cows based on convolutional neural networks (cnn). Biosystems Engineering 198, 31–49. URL: https://www.sciencedirect.com/science/article/pii/S1537511020302129, doi:https://doi.org/10.1016/j.biosystemseng.2020.07.019.

Ahmed, M.M., Hassanien, E.E., Hassanien, A.E., 2024. A smart iot-based monitoring system in poultry farms using chicken behavioural analysis. Internet of Things 25, 101010. doi:10.1016/j.iot.2023.101010.

Akhigbe, B.I., Munir, K., Akinade, O., Akanbi, L., Oyedele, L.O., 2021. Iot technologies for livestock management: A review of present status, opportunities, and future trends. Big Data and Cognitive Computing 5. URL: https://www.mdpi.com/2504-2289/5/1/10, doi:10.3390/bdcc5010010.

Anagnostopoulos, A., Griffiths, B.E., Siachos, N., Neary, J., Smith, R.F., Oikonomou, G., 2023. Initial validation of an intelligent video surveillance system for automatic detection of dairy cattle lameness. Frontiers in Veterinary Science 10. URL: https://www.frontiersin.org/articles/10.3389/fvets.2023.1111057, doi:10.3389/fvets.2023.1111057.

Arago, N., Alvarez, C., Mabale, A., Legista, C., Repiso, N., Amado, T., Jorda, R.J., Thio-ac, A., Tolentino, L.K., Velasco, J., 2022. Smart dairy cattle farming and in-heat detection through the internet of things (iot). International Journal of Integrated Engineering 14. doi:10.30880/ijie.2022.14.01.014.

Atkinson, G.A., Smith, L.N., Smith, M.L., Reynolds, C.K., Humphries, D.J., Moorby, J.M., Leemans, D.K., Kingston-Smith, A.H., 2020. A computer vision approach to improving cattle digestive health by the monitoring of faecal samples. Scientific Reports 10, 17557. doi:10.1038/s41598-020-74511-0.

Bass, L., Clements, P., Kazman, R., 2012. Software Architecture in Practice: Software Architect Practice_c3. SEI Series in Software Engineering, Pearson Education. URL: https://books.google.com.br/books?id=-II73rBDXCYC.

Biglari, A., Tang, W., 2022. A vision-based cattle recognition system using tensorflow for livestock water intake monitoring. IEEE Sensors Letters 6, 1–4. doi:10.1109/LSENS.2022.3215699.

Bonneau, M., Godard, X., Bambou, J.C., 2022. Assessing goats' fecal avoidance using image analysis-based monitoring. Frontiers in Animal Science 3, 835516. doi:10.3389/fanim.2022.835516.

Brown, A., Wilson, G., 2011. The Architecture of Open Source Applications: Elegance, Evolution, and a Few Fearless Hacks. volume 1. Lulu.com. URL: https://api.semanticscholar.org/CorpusID:14285235.

Čakić, S., Popović, T., Krčo, S., Nedić, D., Babić, D., 2022. Developing object detection models for camera applications in smart poultry farms, in: 2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS), IEEE. pp. 1–5. doi:10.1109/COINS54846.2022.9854975.

Cakic, S., Popovic, T., Krco, S., Nedic, D., Babic, D., Jovovic, I., 2023. Developing edge ai computer vision for smart poultry farms using deep learning and hpc. Sensors 23, 3002. doi:10.3390/s23063002.

Carrera-Rivera, A., Ochoa, W., Larrinaga, F., Lasa, G., 2022. How-to conduct a systematic literature review: A quick guide for computer science research. MethodsX 9, 101895. URL: https://doi.org/10.1016/j.mex.2022.101895, doi:10.1016/j.mex.2022.101895.

Ding, L., Lv, Y., Yu, L., Ma, W., Li, Q., Gao, R., Yu, Q., 2023. Real-time monitoring of fan operation in livestock houses based on the image processing. Expert Systems with Applications 213, 118683. doi:10.1016/j.eswa.2022.118683.

Du, A., Guo, H., Lu, J., Su, Y., Ma, Q., Ruchay, A., Marinello, F., Pezzuolo, A., 2022. Automatic livestock body measurement based on keypoint detection with multiple depth cameras. Computers and Electronics in Agriculture 198, 107059. doi:10.1016/j.compag.2022.107059.

Fernández-Carrión, E., Barasona, J.Á., Sánchez, Á., Jurado, C., Cadenas-Fernández, E., Sánchez-Vizcaíno, J.M., 2020. Computer vision applied to detect lethargy through animal motion monitoring: a trial on african swine fever in wild boar. Animals 10, 2241. doi:10.3390/ani10122241.

Fernández-Carrión, E., Martínez-Avilés, M., Ivorra, B., Martínez-López, B., Ramos, Á.M., Sánchez-Vizcaíno, J.M., 2017. Motion-based video monitoring for early detection of livestock diseases: The case of african swine fever. PloS one 12, e0183793. doi:10.1371/journal.pone.0183793.

Fouda, M.M., Sakib, S., Fadlullah, Z.M., Nasser, N., Guizani, M., 2022. A lightweight hierarchical ai model for uav-enabled edge computing with forest-fire detection use-case. IEEE Network 36, 38–45. doi:10.1109/MNET.003.2100325.

Goyal, V., Yadav, A., Mukherjee, R., 2021. Thermo graphic camera-based e-iot enabled architecture for smart poultry farm, in: 2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), IEEE. pp. 772–775. doi:10.1109/ICEECCOT52851.2021.9708062.

Guo, Q., Sun, Y., Orsini, C., Bolhuis, J.E., de Vlieg, J., Bijma, P., de With, P.H., 2023a. Enhanced camera-based individual pig detection and tracking for smart pig farms. Computers and Electronics in Agriculture 211, 108009. doi:10.1016/j.compag.2023.108009.

Guo, Y., He, D., Chai, L., 2020. A machine vision-based method for monitoring scene-interactive behaviors of dairy calf. Animals 10, 190. doi:10.3390/ani10020190.

Guo, Y., Hong, W., Wu, J., Huang, X., Qiao, Y., Kong, H., 2023b. Vision-based cow tracking and feeding monitoring for autonomous livestock farming: The yolov5s-ca+ deepsort-vision transformer. IEEE Robotics & Automation Magazine doi:10.1109/MRA.2023.3310857.

Guo, Y., Qiao, Y., Sukkarieh, S., Chai, L., He, D., 2021. Bigru-attention based cow behavior classification using video data for precision livestock farming. Transactions of the ASABE 64, 1823–1833. doi:10.13031/trans.14658.

Hakansson, F., Jensen, D.B., 2023. Automatic monitoring and detection of tail-biting behavior in groups of pigs using video-based deep learning methods. Frontiers in Veterinary Science 9, 1099347. doi:10.3389/fvets.2022.1099347.

Han, L., Tao, P., Martin, R.R., 2019. Livestock detection in aerial images using a fully convolutional network. Computational Visual Media 5, 221–228. doi:10.1007/s41095-019-0132-5.

Han, S., Fuentes, A., Yoon, S., Jeong, Y., Kim, H., Park, D.S., 2023. Deep learning-based multi-cattle tracking in crowded livestock farming using video. Computers and Electronics in Agriculture 212, 108044. doi:10.1016/j.compag.2023.108044.

Hengstler, S., Aghajan, H., Goldsmith, A., 2007. Application-oriented design of smart camera networks, in: 2007 First ACM/IEEE International Conference on Distributed Smart Cameras, pp. 12–19. doi:10.1109/ICDSC.2007.4357500.

Jorquera-Chavez, M., Fuentes, S., Dunshea, F.R., Warner, R.D., Poblete, T., Unnithan, R.R., Morrison, R.S., Jongman, E.C., 2021. Using imagery and computer vision as remote monitoring methods for early detection

of respiratory disease in pigs. Computers and Electronics in Agriculture 187, 106283. URL: https://www.sciencedirect.com/science/article/pii/S0168169921003008, doi:https://doi.org/10.1016/j.compag.2021.106283.

Khalil, R.A., Saeed, N., Masood, M., Fard, Y.M., Alouini, M.S., Al-Naffouri, T.Y., 2021. Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications. IEEE Internet of Things Journal 8, 11016–11040. doi:10.1109/JIOT.2021.3051414.

Koh, M.E., Fong, M.W.K., Ng, E.Y.K., 2023. Aqua3dnet: Real-time 3d pose estimation of livestock in aquaculture by monocular machine vision. Aquacultural Engineering 103, 102367. doi:10.1016/j.aquaeng.2023.102367.

Kounalakis, T., Triantafyllidis, G.A., Nalpantidis, L., 2019. Deep learning-based visual recognition of rumex for robotic precision farming. Computers and Electronics in Agriculture 165, 104973. doi:10.1016/j.compag.2019.104973.

Le Cozler, Y., Allain, C., Caillot, A., Delouard, J., Delattre, L., Luginbuhl, T., Faverdin, P., 2019. High-precision scanning system for complete 3d cow body shape imaging and analysis of morphological traits. Computers and Electronics in Agriculture 157, 447–453. doi:10.1016/j.compag.2019.01.019.

Lee, S., Ahn, H., Seo, J., Chung, Y., Park, D., Pan, S., 2019. Practical monitoring of undergrown pigs for iot-based large-scale smart farm. IEEE Access 7, 173796–173810. doi:10.1109/ACCESS.2019.2955761.

Li, B., Liu, L., Shen, M., Sun, Y., Lu, M., 2019. Group-housed pig detection in video surveillance of overhead views using multi-feature template matching. Biosystems Engineering 181, 28–39. doi:10.1016/j.biosystemseng.2019.02.018.

Li, N., Ren, Z., Li, D., Zeng, L., 2020. Automated techniques for monitoring the behaviour and welfare of broilers and laying hens: towards the goal of precision livestock farming. animal 14, 617–625.

Liu, C.S., Chen, X.T., Shih, W.Y., Lin, C.C., Yen, J.H., Huang, C.J., Yen, Y.T., 2023. Smart water quality monitoring technology for fish farms using cellphone camera sensor. Sensors & Materials 35. doi:10.18494/SAM4513.

Lynge, M.S., Henriksen, K.S., Moeslund, T.B., 2022. Autonomous monitoring of finishing pigs using side-view cameras and deep learning, in: Fourteenth International Conference on Machine Vision (ICMV 2021), SPIE. pp. 395–403. doi:10.1117/12.2623618.

Ma, W., Qi, X., Sun, Y., Gao, R., Ding, L., Wang, R., Peng, C., Zhang, J., Wu, J., Xu, Z., Li, M., Zhao, H., Huang, S., Li, Q., 2024. Computer vision-based measurement techniques for livestock body dimension and weight: A review. Agriculture 14. URL: https://www.mdpi.com/2077-0472/14/2/306, doi:10.3390/agriculture14020306.

Marino, R., Petrera, F., Abeni, F., 2023. Scientific productions on precision livestock farming: An overview of the evolution and current state of research based on a bibliometric analysis. Animals 13. URL: https://www.mdpi.com/2076-2615/13/14/2280, doi:10.3390/ani13142280.

McVey, C., Egger, D., Pinedo, P., 2022. Improving the reliability of scale-free image morphometrics in applications with minimally restrained livestock using projective geometry and unsupervised machine learning. Sensors 22, 8347. doi:10.3390/s22218347.

Meckbach, C., Tiesmeyer, V., Traulsen, I., 2021. A promising approach towards precise animal weight monitoring using convolutional neural networks. Computers and Electronics in Agriculture 183, 106056. URL: https://www.sciencedirect.com/science/article/pii/S0168169921000740, doi:https://doi.org/10.1016/j.compag.2021.106056.

Meena, S.D., Agilandeeswari, L., 2021. Smart animal detection and counting framework for monitoring livestock in an autonomous unmanned ground vehicle using restricted supervised learning and image fusion. Neural Processing Letters 53, 1253–1285. doi:10.1007/s11063-021-10439-4.

Neethirajan, S., Kemp, B., 2021. Digital livestock farming. Sensing and Bio-Sensing Research 32, 100408.

Nilsson, M., Herlin, A.H., Ardö, H., Guzhva, O., Åström, K., Bergsten, C., 2015. Development of automatic surveillance of animal behaviour and welfare using image analysis and machine learned segmentation technique. Animal 9, 1859–1865. doi:10.1017/S1751731115001342.

Page, M.J., McKenzie, J.E., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., Shamseer, L., Tetzlaff, J.M., Akl, E.A., Brennan, S.E., Chou, R., Glanville, J., Grimshaw, J.M., Hróbjartsson, A., Lalu, M.M., Li, T., Loder, E.W., Mayo-Wilson, E., McDonald, S., McGuinness, L.A., Stewart, L.A., Thomas, J., Tricco, A.C., Welch, V.A., Whiting, P., Moher, D., 2021. The prisma 2020 statement: an updated guideline for reporting systematic reviews. BMJ 372. URL: https://www.bmj.com/content/372/bmj.n71, doi:10.1136/bmj.n71, arXiv:https://www.bmj.com/content/372/bmj.n71.full.pdf.

Pessanha, F., McLennan, K., Mahmoud, M., 2020. Towards automatic monitoring of disease progression in sheep: A hierarchical model for sheep facial expressions analysis from video, in: 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), IEEE. pp. 387–393. doi:10.1109/FG47880.2020.00107.

Prior, J.H., Campbell, M.D., Dawkins, M., Mickle, P.F., Moorhead, R.J., Alaba, S.Y., Shah, C., Salisbury, J.R., Rademacher, K.R., Felts, A.P., et al., 2023. Estimating precision and accuracy of automated video post-processing: A step towards implementation of ai/ml for optics-based fish sampling. Frontiers in Marine Science 10, 1150651. doi:10.3389/fmars.2023.1150651.

Qiao, Y., Guo, Y., Yu, K., He, D., 2022a. C3d-convlstm based cow behaviour classification using video data for precision livestock farming. Computers and electronics in agriculture 193, 106650. doi:10.1016/j.compag.2021.106650.

Qiao, Y., Kong, H., Clark, C., Lomax, S., Su, D., Eiffert, S., Sukkarieh, S., 2021. Intelligent perception for cattle monitoring: A review for cattle identification, body condition score evaluation, and weight estimation. Computers and Electronics in Agriculture 185, 106143. doi:https://doi.org/10.1016/j.compag.2021.106143.

Qiao, Y., Xue, T., Kong, H., Clark, C., Lomax, S., Rafique, K., Sukkarieh, S., 2022b. One-shot learning with pseudo-labeling for cattle video segmentation in smart livestock farming. Animals 12, 558. doi:10.3390/ani12050558.

Saitoh, T., Kato, Y., 2021. Evaluation of wearable cameras for monitoring and analyzing calf behavior: A preliminary study. Animals 11, 2622. doi:10.3390/ani11092622.

Salau, J., Krieter, J., 2020. Analysing the space-usage-pattern of a cow herd using video surveillance and automated motion detection. Biosystems Engineering 197, 122–134. doi:10.1016/j.biosystemseng.2020.06.015.

Schmidt, T.B., Lancaster, J.M., Psota, E., Mote, B.E., Hulbert, L.E., Holliday, A., Woiwode, R., Pérez, L.C., 2022. Evaluation of a novel computer vision-based livestock monitoring system to identify and track specific behaviors of individual nursery pigs within a group-housed environment. Translational Animal Science 6, txac082. doi:10.1093/tas/txac082.

Son, S., Ahn, H., Baek, H., Yu, S., Suh, Y., Lee, S., Chung, Y., Park, D., 2022. Staticpigdet: Accuracy improvement of static camera-based pig monitoring using background and facility information. Sensors 22, 8315. doi:10.3390/s22218315.

Soro, S., Heinzelman, W., 2009. A survey of visual sensor networks. Advances in Multimedia 2009, 640386. URL: https://onlinelibrary.wiley.com/doi/abs/10.1155/2009/640386, doi:https://doi.org/10.1155/2009/640386, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1155/2009/640386.

Su, Q., Tang, J., Zhai, J., Sun, Y., He, D., 2021. Automatic tracking of the dairy goat in the surveillance video. Computers and Electronics in Agriculture 187, 106254. doi:10.1016/j.compag.2021.106254.

Sumi, K., Zin, T.T., Kobayashi, I., Horii, Y., 2021. Framework of cow calving monitoring system using video images. Journal of Advances in Information Technology Vol 12. doi:10.12720/jait.12.3.240-245.

Sundaram, N., Meena, D.S., 2023. Integrated animal monitoring system with animal detection and classification capabilities: a review on image modality, techniques, applications, and challenges. Artificial Intelligence Review 56, 1–51. doi:10.1007/s10462-023-10534-z.

Szymanski, L., Lee, M., 2020. Deep sheep: kinship assignment in livestock from facial images, in: 2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ), IEEE. pp. 1–6. doi:10.1109/ivcnz51579.2020.9290558.

Tsai, Y.C., Hsu, J.T., Ding, S.T., Rustia, D.J.A., Lin, T.T., 2020. Assessment of dairy cow heat stress by monitoring drinking behaviour using an embedded imaging system. Biosystems Engineering 199, 97–108. URL: https://www.sciencedirect.com/science/article/pii/S1537511020300763, doi:https://doi.org/10.1016/j.biosystemseng.2020.03.013. environmental stressors and animal production: improved performance through understanding and engineering advances.

Van Hertem, T., Schlageter Tello, A., Viazzi, S., Steensels, M., Bahr, C., Romanini, C.E.B., Lokhorst, K., Maltz, E., Halachmi, I., Berckmans, D., 2018. Implementation of an automatic 3d vision monitor for dairy cow locomotion in a commercial farm. Biosystems Engineering 173, 166–175. URL: https://www.sciencedirect.com/science/article/pii/S1537511017300570, doi:https://doi.org/10.1016/j.biosystemseng.2017.08.011. advances in the Engineering of Sensor-based Monitoring and Management Systems for Precision Livestock Farming.

Van Hertem, T., Tello, A.S., Viazzi, S., Steensels, M., Bahr, C., Romanini, C.E.B., Lokhorst, K., Maltz, E., Halachmi, I., Berckmans, D., 2018. Implementation of an automatic 3d vision monitor for dairy cow locomotion in a commercial farm. Biosystems Engineering 173, 166–175. doi:10.1016/j.biosystemseng.2017.08.011.

Wang, D., Tang, J., Zhu, W., Li, H., Xin, J., He, D., 2018. Dairy goat detection based on faster r-cnn from surveillance video. Computers and Electronics in Agriculture 154, 443–449. doi:10.1016/j.compag.2018.09.030.

Wang, M., Li, X., Larsen, M.L., Liu, D., Rault, J.L., Norton, T., 2023a. A computer vision-based approach for respiration rate monitoring of group housed pigs. Computers and Electronics in Agriculture 210, 107899. URL: https://www.sciencedirect.com/science/article/pii/S0168169923002879, doi:https://doi.org/10.1016/j.compag.2023.107899.

Wang, M., Youssef, A., Larsen, M., Rault, J.L., Berckmans, D., Marchant-Forde, J.N., Hartung, J., Bleich, A., Lu, M., Norton, T., 2021. Contactless video-based heart rate monitoring of a resting and an anesthetized pig. Animals 11, 442. doi:10.3390/ani11020442.

Wang, S., Jiang, H., Qiao, Y., Jiang, S., Lin, H., Sun, Q., 2022. The research progress of vision-based artificial intelligence in smart pig farming. Sensors 22. URL: https://www.mdpi.com/1424-8220/22/17/6541, doi:10.3390/s22176541.

Wang, Y., Mücher, S., Wang, W., Guo, L., Kooistra, L., 2023b. A review of three-dimensional computer vision used in precision livestock farming for cattle growth management. Computers and Electronics in Agriculture 206, 107687. URL: https://www.sciencedirect.com/science/article/pii/S0168169923000753, doi:https://doi.org/10.1016/j.compag.2023.107687.

Wang, Z., Wang, S., Wang, C., Zhang, Y., Zong, Z., Wang, H., Su, L., Du, Y., 2023c. A non-contact cow estrus monitoring method based on the thermal infrared images of cows. Agriculture 13, 385. doi:10.3390/agriculture13020385.

Wu, D., Han, M., Song, H., Song, L., Duan, Y., 2023. Monitoring the respiratory behavior of multiple cows based on computer vision and deep learning. Journal of Dairy Science 106, 2963–2979. doi:10.3168/jds.2022-22501.

Xavier, C., Le Cozler, Y., Depuille, L., Caillot, A., Lebreton, A., Allain, C., Delouard, J., Delattre, L., Luginbuhl, T., Faverdin, P., et al., 2022. The use of 3-dimensional imaging of holstein cows to estimate body weight and monitor the composition of body weight change throughout lactation. Journal of Dairy Science 105, 4508–4519. doi:10.3168/jds.2021-21337.

Xi, M., Rahman, A., Nguyen, C., Arnold, S., McCulloch, J., 2023. Smart headset, computer vision and machine learning for efficient prawn farm management. Aquacultural Engineering 102, 102339. URL: https://www.sciencedirect.com/science/article/pii/S0144860923000262, doi:https://doi.org/10.1016/j.aquaeng.2023.102339.

Xu, B., Wang, W., Falzon, G., Kwan, P., Guo, L., Sun, Z., Li, C., 2020. Livestock classification and counting in quadcopter aerial images using mask r-cnn. International Journal of Remote Sensing 41, 8121–8142. doi:10.1080/01431161.2020.1734245.

Yujuan, C., Dongjian, H., Yinxi, F., Huaibo, S., 2017. Intelligent monitoring method of cow ruminant behavior based on video analysis technology. International Journal of Agricultural and Biological Engineering 10, 194–202. doi:10.25165/j.ijabe.20171005.3117.

Zhang, A.L.N., Wu, B.P., Jiang, C.X.H., Xuan, D.C.Z., Ma, E.Y.H., Zhang, F.Y.A., 2018. Development and validation of a visual image analysis for monitoring the body size of sheep. Journal of Applied Animal Research 46, 1004–1015. doi:10.1080/09712119.2018.1450257.

Zhang, L., Han, G., Qiao, Y., Xu, L., Chen, L., Tang, J., 2023a. Interactive dairy goat image segmentation for precision livestock farming. Animals 13, 3250.

Zhang, X., Xuan, C., Ma, Y., Su, H., 2023b. A high-precision facial recognition method for small-tailed han sheep based on an optimised vision transformer. animal 17, 100886. doi:10.1016/j.animal.2023.100886.

Zhi, Q., Saitoh, T., Nakajima, M., Saitoh, T., 2019. A development of content-based video summarization system using machine-learning and its application to analysis of livestock behavior, in: International Workshop on Advanced Image Technology (IWAIT) 2019, SPIE. pp. 193–199. doi:10.1117/12.2522172.

Zhuang, Y., Zhou, K., Zhou, Z., Ji, H., Teng, G., 2023. Systems to monitor the individual feeding and drinking behaviors of growing pigs based on machine vision. Agriculture 13. URL: https://www.mdpi.com/2077-0472/13/1/103, doi:10.3390/agriculture13010103.

4 A COMPUTER VISION END-TO-END METHOD FOR PREDICTING SHEEP BODY WEIGHT FROM DEPTH IMAGES

Small Ruminant Research 261 (2026) 107799



Contents lists available at ScienceDirect

Small Ruminant Research

journal homepage: www.elsevier.com/locate/smallrumres





A computer vision end-to-end method for predicting sheep body weight from depth images

Ewerton Costa Sousa [a,c,*], Ricardo de Andrade Lira Rabelo [b], José Lindenberg Rocha Sarmento [a]

[a] Department of Animal Science, Federal University of Piauí, Campus Petrônio Portela, Teresina, 64049-550, Piauí, Brazil
[b] Department of Computer Science, Federal University of Piauí, Campus Petrônio Portela, Teresina, 64049-550, Piauí, Brazil
[c] Department of Computer Science, Federal Institute of Piauí, Campus Angical, Angical, 64410-000, Piauí, Brazil

ARTICLE INFO

Keywords:
Computer vision
General linear mixed model
Depth image
Precision livestock farming
Sheep
Weight prediction

ABSTRACT

Regular monitoring of animal weight is essential in livestock management; however, traditional weighing methods using scales are labor-intensive and stressful for both animals and humans. To address these challenges, computer vision (CV) techniques have been explored for predicting animal weight from images. However, current works with sheep still suffer from low accuracy and generalization. Within the scope of a coarse-to-fine research approach aimed at developing more efficient, non-invasive, and generalizable methods for small ruminants weighing in livestock farms, this study evaluates the accuracy and generalization of predicting the body weight of sheep from dorsal depth images. It was proposed a ConvNet method based on the EfficientNet-B3 network which operates in an end-to-end fashion: it automatically locates the animal, measures the traits most relevant to prediction, and estimates sheep's body weight. Model performance was evaluated using the Coefficient of Determination (R^2), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). Saliency maps were generated to qualitatively assess feature relevance and model interpretability, and the impacts of combining images from different farms on prediction performance were also analyzed. Moreover, the factors that influenced the performance (experimental, animal, and image-related sources of variation) were investigated using a novel approach based on Generalized Linear Mixed Model (GLMM). The proposed model achieved promising results, with an R^2 of 0.97, MAE of 1.68 kg, and MAPE of 6.74%, demonstrating competitive performance with existing approaches. These findings contributes to advance technological methods for small ruminants meat sector.

1. Introduction

Regular monitoring of animal weight is essential to improve management and handling decisions contributing to reduce herd mortality rates and costs in livestock farms (Zhang et al., 2025). Body weight can be considered an indicator of different animal traits and behaviors, such as health conditions (Dohmen et al., 2022) and the best slaughter time (Bo et al., 2023), as well as can help to optimize feeding strategies contributing to reduce the carbon footprint (Chen et al., 2025). Over the years, several technologies have been proposed to enhance the efficiency and accuracy of animal weighing, thereby reducing the effort and time required when using traditional scales and facilitating more frequent monitoring of weight (Dohmen et al., 2022). Among the proposed approaches, techniques for predicting the weight from images have gained prominence in the last decade, particularly those based on Computer Vision (**CV**) (Zhao et al., 2023). These methods aim to estimate the weight by applying regression models in which the predictor variables correspond to traits automatically extracted from images.

Recent studies, primarily with pigs and cattle, have achieved promising accuracy by adopting more modern approaches based on Convolutional Neural Networks (ConvNet). Examples include work on cattle by Li and Teng (2022), Hou et al. (2023), Afridi et al. (2024), Liao et al. (2025), and Zhang et al. (2025); and on pigs by Meckbach et al. (2021). ConvNets, recognized for their ability to accurately capture spatial relationships in images (Goodfellow et al., 2016), have been used to measure relevant traits from images that correlate with animal mass (e.g. rump height, body length, chest width, heart girth, abdominal circumference), thereby supporting the weight estimation process. Conversely, methods for predicting the weight of sheep from images have demonstrated only incremental progress—a trend that contrasts with the inherent challenge to performing this task and the growing importance of the small ruminant meat sector, particularly in

* Corresponding author at: Department of Animal Science, Federal University of Piauí, Campus Petrônio Portela, Teresina, 64049-550, Piauí, Brazil.
E-mail address: ewertoncosta@ifpi.edu.br (E.C. Sousa).

https://doi.org/10.1016/j.smallrumres.2026.107799
Received 20 November 2025; Received in revised form 4 May 2026; Accepted 5 May 2026
Available online 15 May 2026
0921-4488/© 2026 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

developing regions where production is expanding and there is a critical need for accessible, technology-driven management tools (OECD/FAO, 2025). Weighing sheep based on images is challenging because coat colors and wool can varies across breeds and due to their behavior under stress, such as escape attempts, running, jumping, and other actions (Atkinson et al., 2024) that can affect image quality.

Current studies on sheep, such as those proposed in Abdelhady et al. (2019), Gurgel et al. (2021), and Sant'Ana et al. (2021), are still heavily focused on traditional digital image processing (DIP) techniques and suffer from low accuracy and generalization. Marques et al. (2025), in contrast, achieved promising accuracy, with a R^2 of 0.92, by proposing a method that uses a ConvNet model based on the U-Net network architecture to explicitly segment the animal from images, followed by an ensemble of ConvNet models that extract the features used for weight estimation. These studies share two key characteristics: they rely on explicit animal segmentation and estimate weight from morphometric traits derived from RGB images. This design introduces two major limitations: (i) by focusing on coat color patterns, the models tend to become specialized to the training data and may exhibit inadequate generalization to other breeds or multi-colored animals; and (ii) explicit segmentation requires a rigorous and labor-intensive manual image annotation process (Chen et al., 2025), making deployment in new farms costly, as models must be retrained for each new herd.

Therefore, this work aims to improve the accuracy and generalization of predicting the body weight of sheep from images. The proposed method combines the efficiency of the EfficientNet-B3 ConvNet network with the 3D surface information provided by dorsal depth images and was designed to automatically localize the animal in the image and estimate its weight, without requiring an explicit segmentation step. Depth images acquired in top-view perspective capture the shape and height of the animal, traits correlated with weight, and are less influenced by lighting or animal color than RGB images (Meckbach et al., 2021), enhancing the accuracy and robustness of the prediction. The images were acquired outdoors on two livestock farms with animals in motion, which is a less invasive strategy that causes less stress and consequently contributes to ensure animal welfare.

Since regression models may exhibit significant variations in performance when exposed to different datasets, the method was evaluated under multiple training–test splits designed to assess its ability to generalize across animals with distinct coat colors, farms, and weight ranges. Method performance was evaluated quantitatively, using the Coefficient of Determination (R^2), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) metrics, and qualitatively, based on saliency maps generated to assess whether the trained model precisely locates the animal and learns relevant traits. Moreover, the factors that influenced the performance (experimental, animal, and image-related sources of variation) were investigated using a novel approach based on Generalized Linear Mixed Model (GLMM).

The main contributions of this work are as follows: (i) the method proposed demonstrated be a non-invasive and efficient tool that, combined with dorsal depth images, achieved strong prediction accuracy and generalization capabilities (R^2 = 0.97, MAE = 1.68 kg, and MAPE = 6.74%), contributing to advance technological methods for small ruminants meat sector; (ii) the GLMM approach enabled the estimation of effects that generalize across different training–test splits and explained how sources of variation related to the experiment, animals, and images impact model performance, providing an efficient and indirect framework for understanding deep learning models; and (iii) it further shows that an end-to-end approach without explicit segmentation can create a network effect, reducing deployment time and effort on new farms.

The next sections are organized as follows: The materials and procedures used in this study are detailed in Section 2. The results are presented in Section 3, and the practical implications of the investigation are discussed in Section 4. The main conclusions of the study are presented in Section 5.

2. Materials and methods

This study employed a carefully designed methodology to ensure a rigorous and reliable evaluation of the accuracy and generalization of the proposed method, as well as to investigate the factors that most influence prediction performance. The approach comprises seven main stages, illustrated in Fig. 1 and described in the following sections.

All animal experiments were carried out according to the protocols reviewed and approved by the Ethics Committee for the Use of Production Animals (CEUA) of the Federal University of Piauí(UFPI), Teresina-PI, Brazil, under protocol number 851/2024.

2.1. Data acquisition

In this stage, the sheep depth images and associated metadata was collected in two livestock farms. Subsequently, a qualitative visual assessment was conducted to retain only relevant images for the experiments, resulting in the first version of the dataset.

2.1.1. Experimental setup

RGB, depth, and infrared (IR) images of moving sheep were collected from a dorsal top-view perspective using an MS-A075V 3D Time of Flight (TOF) camera module (Sipeed, 2022). This module captures depth images with a (240, 320, 1) resolution and is a low-cost alternative for Kinect or Intel Real Sense D family devices, the most used cameras in this kind of research, and can contribute to developing cheaper products for small and medium livestock farms. A self-developed platform, named **cv-app**, was used to capture the images, associate them with the animals' metadata, and to support the dataset analysis process.

The pixel values in a depth image represents the distance of the objects in the scene to the camera. Therefore, when acquired in a dorsal perspective, depth images implicitly include information about the animal height and can effectively capture traits significantly correlated with body weight, such as spine, ribes, and muscles (Zhang et al., 2025). In addition, these kind of images are not affected by the animal color (Meckbach et al., 2021), enable to obtain high-quality 3D measurements even with animals in motion (Sant'Ana et al., 2021), and can capture the 3D surface of an object (Hou et al., 2023). These characteristics contributes to improve the accuracy and robustness of weight prediction.

2.1.2. Data collection

Three rounds of data acquisition were carried out on two livestock farms, hereafter referred to as Farm A and Farm B, located in the city of Acauã-PI, Brazil, on January 24–25, 2025. The location and structures used were planned to resemble the livestock environment commonly adopted by farms in that region. The first data collection took place on Farm A on January 25, 2025, between 10:00 and 11:00 h. The activities happened in a corral where the animals are usually brought for various handling procedures. The site, which consists of four pens, was adapted for the experiment (Fig. 2(a)). After carrying out preliminary tests, the entrance of one of the pens was set up to create a corridor measuring 4 m in length and 80 cm using a 1-meter-high plastic mesh on one side and the corral fence on the other. In the middle of the corridor, the camera was mounted at a height of 1.95 m on a metal structure, from which images were captured at a rate of 60 frames per second (FPS) as the animals passed through the corridor (Fig. 2(b)). Throughout the experiment, the sky was clear and sunny, but with the occasional appearance of clouds.

The second data collection, conducted on Farm B, also took place on January 25, 2025, but between 16:50 and 17:50 h. The activities were also conducted in a corral where the animals are usually brought for various handling procedures. However, this corral already had a built-in corridor measuring 10 m in length and 80 cm in width, where the camera infrastructure was set up, as shown in Fig. 2(c). During the

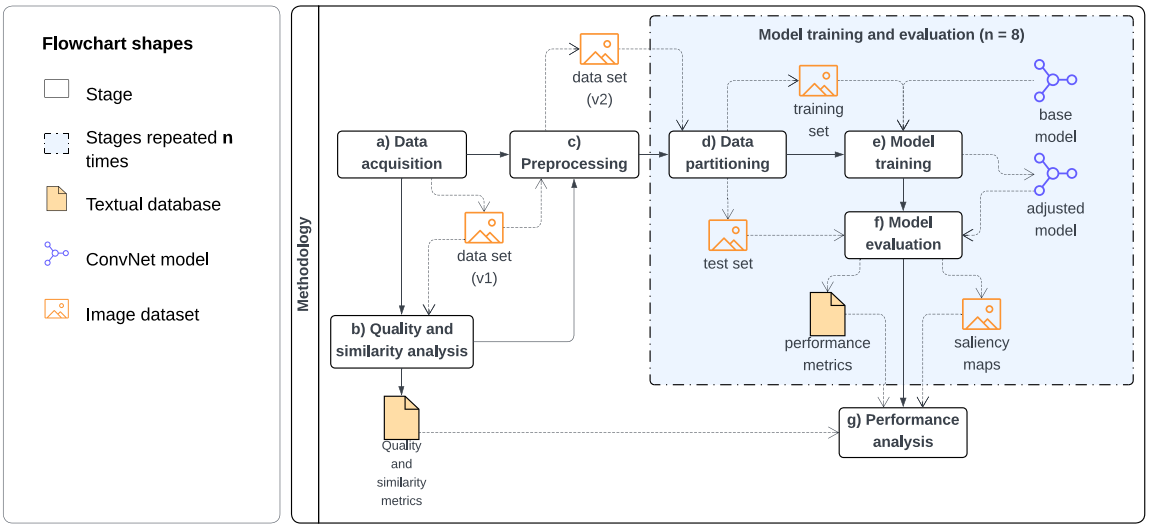


Fig. 1. Flowchart of the methodology illustrating the main stages of this study: **(a)** Data acquisition, **(b)** Quality and similarity analysis, **(c)** Preprocessing, **(d)** Data partitioning, **(e)** Model training, **(f)** Model evaluation, and **(g)** Performance analysis. Arrows indicate the flow of information between stages as well as their inputs and outputs. The blue dashed box indicates that the enclosed stages are repeated eight times (n = 8). Other symbols are explained in the caption on the left.



(a) (b) (c)

Fig. 2. Environmental setup for data acquisition. **(a)** Corral structure where images were acquired in Farm A; a similar arrangement was used in Farm B. **(b)** and **(c)** show the corridors through which sheep move during image acquisition in Farms A and B, respectively. All images are original and owned by the authors.

experiment, the sky alternated between cloudy and sunny with good lighting conditions. The third data collection took place on January 26, 2025, from 10:00 to 10:30 h, once again on Farm A. The length of the corridor increased from 4 to 5 m, while the width remained constant at 80 cm. This modification was implemented after observing animals jumping at the end of the corridor, which negatively affected image suitability for weight estimation. Following the adjustment, the incidence of jumping decreased. Once again, the sky was clear and very sunny.

2.1.3. Dataset

During data collections, images, measured weights, and metadata (animal identification, sex, age, breed, wool level, body condition score, and the occurrence of unexpected behaviors) of 221 sheep were acquired, producing an initial dataset of 16,938 depth images. Following a qualitative visual assessment, images from 32 animals were excluded: 26 because the animals jumped through the corridor during data collection, 5 due to issues with the cv-app platform, and 1 due to a recorded weight inconsistency. This filtering step resulted in 14,627 depth images corresponding to 189 sheep (Farm A = 91; Farm B = 98). For the subsequent stages, only images in which the animal appeared fully within the scene were kept, producing a new dataset of 1772 images named **dataset (v1)**.

Summary statistics of the dataset distribution are shown in Table 1. Variation in the number of images per animal is mainly driven by the time each sheep remains visible to the camera during data collection, which is largely influenced by its speed through the corridor. Animal size also contributes, as larger individuals tend to yield fewer full-body images. Outliers with unusually high number of images (Fig. 3(a)) correspond to animals that stopped in the middle of the corridor or attempted to turn back. Regarding weight, the images are approximately uniformly distributed within the 5–55 kg range, although a lower concentration is observed between 20–30 kg and around 50 kg (Fig. 3(b)). Nevertheless, the dataset can be considered diverse and well distributed, supporting robust weight prediction models (Tu and Jørgensen, 2023).

2.2. Dataset quality and similarity analysis

Given the outliers with high number of images observed in Fig. 3(a) and the apparent similarity between images perceived in Fig. 4, the quality and similarity of the images in dataset were analyzed.

2.2.1. Quality analysis

Image quality was assessed using a Laplacian-based Quality Index (QI), which estimates the high-frequency content of each image. First,

Table 1
Summary statistics of images by number of images per animal and weight for the dataset (column **General**) and by farm.

Number of images			Weight (kg)		
General	Farm A	Farm B	General	Farm A	Farm B
9.37 ± 12.68	6.08 ± 4.01	12.42 ± 16.66	26.28 ± 14.15	30.45 ± 14.53	24.39 ± 13.56

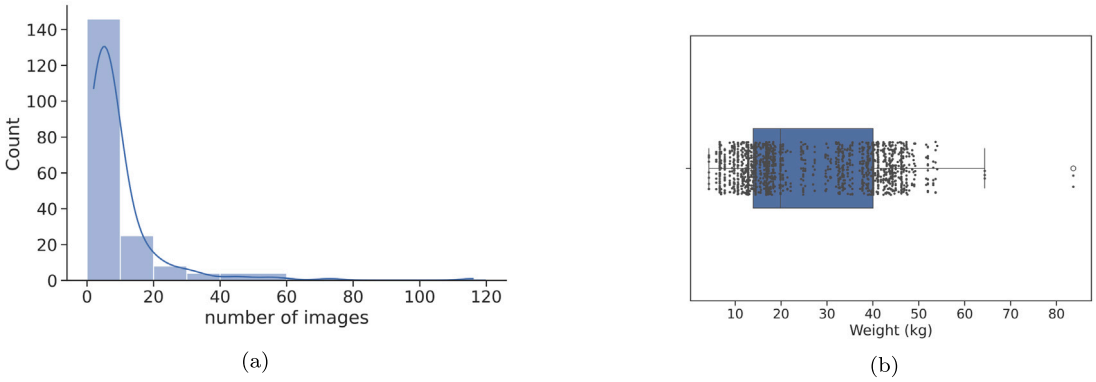


Fig. 3. **(a)** Histogram of the number of images per animal in the dataset. **(b)** Boxplot of images by weight. Black dots represent individual observations.

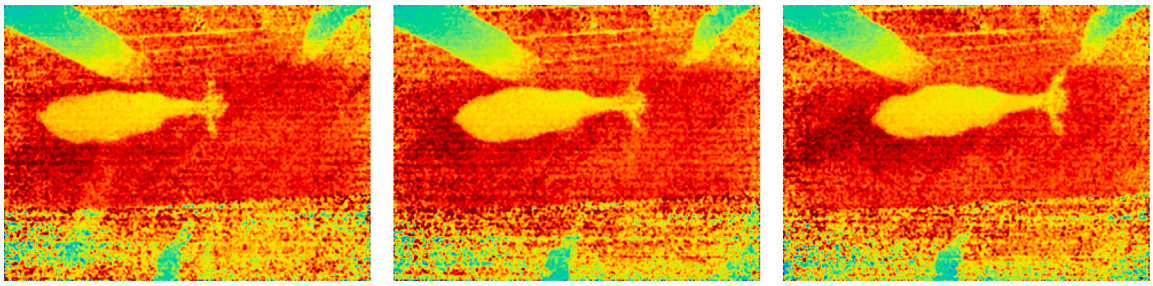


Fig. 4. Sample depth images of the same animal captured as it moves through the corridor. The animal appears in different positions and postures across frames despite high visual similarity. For visualization, pixel values were rescaled to 0–255 (original range: 0–1950).

the Laplacian operator (Gonzalez and Woods, 2018) with window size of 3 is applied to the original depth image (Eq. (1)), producing a new image whose pixel values represent local intensity variations associated with edges. The QI is then computed as the average of the pixel values of this image (Eq. (2)). Others window sizes (5 and 7) were evaluated, but no significant differences were observed. Higher QI values indicate greater high-frequency content and, consequently, a higher likelihood of noise. The QI variability index (QVI) was also computed to assess intra-animal variability in image quality. It is defined as the mean coefficient of variation (CV) of the QI across images of the same animal and was calculated using Eq. (3).

$$L(x, y) = \nabla^2 I(x, y) \tag{1}$$

where:

- $L(x, y)$ = value of the Laplacian at pixel (x, y) in image I;
- $I(x, y)$ = a pixel (x, y) of the original depth image I;
- ∇^2 = denotes the Laplacian operator.

$$QI_i = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} |L_i(x, y)| \tag{2}$$

where:

- QI_i = Laplacian-based Quality Index (QI) of image $_i$;
- M and N = the dimensions of the image $_i$;
- $L_i(x, y)$ = value of the Laplacian operator to pixel (x, y) of the image $_i$;

$$QVI = \frac{1}{N} \sum_{j=1}^{N} \left(\frac{\sigma_j}{\mu_j} \right) \tag{3}$$

where:

- QVI = QI variability index;
- N = number of animals evaluated;
- σ_j = standard deviation of QI computed for images of animal j;
- μ_j = mean of QI computed for images of animal j.

Following QI computation, its relationships with animal weight, farm, image capture time, and unexpected behaviors were investigated. Unexpected behaviors were limited to attempts to turn back or run, as images of jumping animals were removed. Linear associations between QI and these variables were first evaluated using the Pearson correlation coefficient. Subsequently, their influence on quality index was assessed using a GLMM, which extends Generalized Linear Model by incorporating fixed and random effects, allowing the analysis of hierarchical or grouped data structures (McCulloch and Searle, 2001).

Weight, Farm, CaptureTime, and UnexpectedBehavior predictors were modeled as fixed effects, with random effects accounting for repeated image captures of the same animal. A Gamma error distribution with a log link was used to accommodate data asymmetry. Model selection followed (Zuur et al., 2009): an initial model with all main effects was fitted, followed by evaluation of pairwise interactions, with models compared using Akaike (AIC) and Bayesian (BIC) information criteria.

UnexpectedBehavior was treated as a binary variable (True/False). CaptureTime included only the time-of-day component, expressed as minutes since midnight. For example, timestamps such as January 25, 2025 11:05:35 and January 26, 2025 11:05:40 were both converted to the value 665 ($11 \times 60 + 5$). Models were fitted using the **lme4** package (v2.0-1; Bates et al. (2015)) in R.

2.2.2. Similarity analysis

Image similarity within the dataset was evaluated using Root Mean Squared Error (RMSE), Structural Similarity Index (SSIM), and perceptual hash (pHash). RMSE measures the pixel-wise difference between two images (Eq. (4)).

$$RMSE_{ij} = \sqrt{\frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left[I_i(x,y) - I_j(x,y)\right]^2} \quad (4)$$

where:

- $RMSE_{ij}$ = RMSE metric calculated comparing images i and j;
- $I_i(x,y)$ and $I_j(x,y)$ = pixels (x,y) of the images i and j;
- M and N = the dimensions of the images;

Unlike RMSE, which is sensitive to minor pixel-level distortions, SSIM evaluates perceptual similarity by comparing local pixel neighborhoods, reducing the influence of small differences while accounting for luminance, contrast, and structure (Wang et al., 2004). The $SSIM_{ij}$ is computed as the mean of $SSIM_{ij}(x,y)$ over all corresponding pixels of images $_i$ and $_j$ (Eqs. (5) and (6)) and ranges from −1 to 1, though in practice it typically lies between 0 and 1, with higher values indicating greater similarity. The pHash method assesses similarity in the frequency domain by comparing compact hash representations derived from low-frequency components (Steinebach, 2012). Similarity is measured using the Hamming distance (Hamming, 1950) between hashes (Eq. (7)), ranging from 0 to the hash length, with lower values indicating greater similarity.

$$SSIM_{ij}(x,y) = \frac{(2\mu_i\mu_j + C_1)(2\sigma_{ij} + C_2)}{(\mu_i^2 + \mu_j^2 + C_1)(\sigma_i^2 + \sigma_j^2 + C_2)} \quad (5)$$

where:

- $SSIM_{ij}(x,y)$: local structural similarity computed when comparing pixels (x,y) of images $_i$ and $_j$;
- μ_i and μ_j = averages of pixels (x,y) in the neighborhood for images $_i$ and $_j$;
- σ_1^2 and σ_2^2 = variances of pixels (x,y) in the neighborhood for images $_i$ and $_j$;
- σ_{12} = covariance of pixels (x,y) in the neighborhood for images $_i$ and $_j$;
- C_1 and C_2 = stabilization constants.

$$SSIM_{ij} = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N} SSIM_{ij}(x,y) \quad (6)$$

where:

- $SSIM_{ij}$: Structural Similarity Index computed when comparing all pixels in the images $_i$ and $_j$;
- $SSIM_{ij}(x,y)$: local structural similarity computed when comparing pixels (x,y) of images $_i$ and $_j$;
- M and N = dimensions of the images;

$$d_H(h_i, h_j) = \sum_{x=1}^{n} h_{i,x} \oplus h_{j,x} \quad (7)$$

where:

- h_i and h_j = the binary hashes of the images $_i$ and $_j$;
- n = the hash length;
- $\oplus$ = denotes the XOR operation.

Similarity metrics were computed under four comparison settings designed to capture different sources of variation in the dataset: **general**, **intra-animal**, **intra-farm**, and **inter-farm**. In the **general** setting, image pairs were randomly selected from the entire dataset, and the similarity measurements were computed at a ratio of five comparisons per image, totaling 8860 repetitions. In the intra-animal setting, images belonging to the same animal were compared with the immediately subsequent image in the sequence. For the intra-farm setting, comparisons were performed between images from the same farm, with image pairs selected randomly. A proportion of five comparisons per image was maintained, resulting in 2770 repetitions for Farm A and 6090 repetitions for Farm B. Finally, in the inter-farm setting, images from Farm B were compared with images from Farm A. Image pairs were also selected randomly, maintaining the proportion of five comparisons per image based on the number of images available for Farm B, resulting in a total of 6090 repetitions.

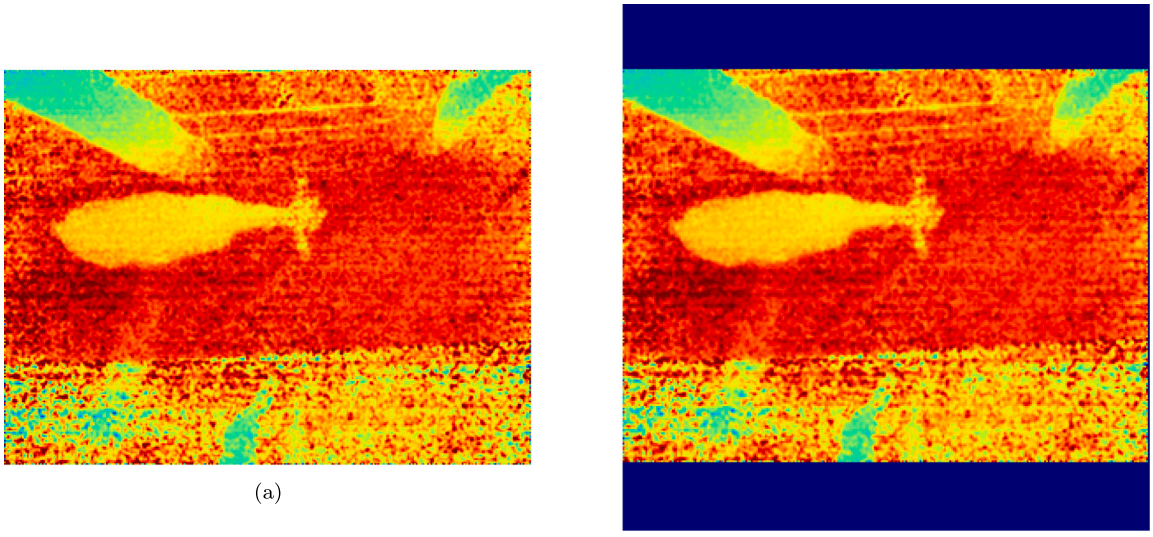
In all settings, images underwent minimal preprocessing. First, the maximum acceptable pixel value (**maxV**) was defined as 1950. Consequently, all pixels with values greater than this threshold were set to **maxV**. As a pixel value in the depth image represents the distance from an object to the camera in millimeters, and the camera was mounted at a height of 1.95 m, values above 1950 were considered invalid. Additionally, the depth images were normalized by dividing each pixel value by **maxV**. RMSE, SSIM, and pHash metrics were calculated using the Python libraries NumPy (v2.0.2; Harris et al. (2020)), scikit-image (v0.25.0; van der Walt et al. (2014)), and ImageHash (v4.3.2; Buchner (2020)), respectively. For SSIM computation, the default parameters settings of scikit-image were adopted for the constants $C1$ and $C2$ and a neighborhood window of size 11 was used, consistent with the configuration used in Wang et al. (2004). For the pHash computation, a hash size of 64 bits was used. All processing steps, scripts, and analyzes mentioned above are publicly available at Sousa et al. (2026b).

2.2.3. Preprocessing

In addition to clipping pixel values above **maxV** and normalizing the images, depth images were converted to three channels for model training by replicating the single-channel content. This step was required to use the pre-trained EfficientNet-B3 network, which expects 3-channel input. Finally, the images were resized to the resolution (300, 300, 3) using the resize with padding, a technique that preserves the aspect ratio of the original image without distorting it, especially when resizing rectangular images to square resolutions, as illustrated in Fig. 5. As a result, a revised version of the dataset was created and named **dataset (v2)**. From this point forward, any reference to the dataset refers to this version. No explicit segmentation, background removal, or direct morphological measurements were performed, as the proposed method automatically locates the animal and estimates sheep body weight.

2.3. Model training and evaluation

To ensure a reliable evaluation of the accuracy and generalization of the prediction, the stages **Data partitioning**, **Model training**, and **Model evaluation** were repeated eight times (hereafter referenced by **run-1** to **run-8**) using the same proposed method (Fig. 1). Since ConvNet models may exhibit significant variations in performance when exposed to different data, assessing the prediction performance through different combinations of training and test sets contributes to achieve greater reliability in the generalization evaluation and allows for a more in-depth analysis of the method's strengths and weaknesses.



(a) (b)

Fig. 5. Illustration of the resize-with-padding technique. (a) Original image. (b) Image resized from (240,320,3) to (300,300,3). The original pixels are preserved, with zero-valued padding added to the top and bottom to match the target resolution. For visualization, pixel values were rescaled to 0–255 (original range: 0–1950).

2.3.1. Data partitioning

The dataset was partitioned into training and test sets, where the former are used to fit the proposed ConvNet model adjusting its internal parameters, and the latter are used to measure the performance metrics in the Model evaluation stage. For runs 1 to 4, the partitioning of the dataset was performed randomly based on the animal identification code, in an 85:15 ratio strategy, with both sets containing images from both farms. In run 5, the training set included only images from Farm B, while the test set included only images from Farm A. In runs 6 and 7, only images of one farm were used, respectively, B and A, and the partitioning was performed randomly based on the animal identification code. Finally, in run-8, images in the 20–35 kg range were used to compose the test set, and the remaining images were used for training. This interval represents the gap with the lowest concentration of images in the dataset (Fig. 3(b)) and contributes to evaluating the model's ability to generalize to animals of other weight ranges. In all executions, images from the same animal were assigned to only one set, ensuring reliable performance evaluation.

2.3.2. Model training

The proposed method consists of a 2D ConvNet model based on the EfficientNet-B3 pre-trained network with parameters established from the ImageNet dataset. EfficientNet-B3 is a network of the EfficientNet family designed to achieve high accuracy using a relatively small number of parameters (Tan and Le, 2020), is suitable for devices with limited resources, and is able to handle complex patterns in RGB and depth images (Afridi et al., 2024). The variant B3 was chosen because it uses an image resolution of (300,300,3), imposing a smaller rescaling of acquired images, originally (240,320,1). The model architecture comprises the backbone network without its output layer, augmented with four layers for regression (Fig. 6). The first added layer is a Global Average Pooling 2D that compresses the spatial features produced by the pre-trained baseline network before passing to the subsequent dense regression layers. The second is a 128 unit dense layer with a *relu* activation function. The third is a Dropout(0.3) layer that randomly sets a fraction of the input units to zero at each training step. This strategy helps the model learn more robust representations, preventing overfitting and increasing robustness (Srivastava et al., 2014). Ultimately, the output layer is a dense layer with just one unit and a linear activation function. This end-to-end model automatically locates the animal in the image, identifies and extracts the traits most correlated to the prediction and estimates the sheep's body weight.

Model fitting was performed using augmented depth images as input and measured sheep weights as targets. The on-the-fly augmentation technique was applied during training, in which vertical and/or horizontal flip transformations were applied dynamically and randomly to the images at each epoch. Therefore, new images are not generated in a preprocessing step before training, but instead a variation of the original image is presented to the model at each epoch. This avoids overfitting without increasing the size of the dataset in memory (Cerqueira et al., 2024).

Given limited training data, which can hinder training ConvNet models from scratch, the model was adapted using transfer learning and fine-tuning, with mean squared error (MSE) as the loss function and the Adam optimizer (learning rate = 3×10^{-4}). The model training was carried out using the cross-validation technique, where for each epoch 80% of the training set was used to adjust the weights of the network and 20% for validation. Early stopping was used by monitoring the validation loss, halting training if no improvement greater than 1×10^{-4} was observed for 15 consecutive epochs, with weights restored to those of the best epoch. Additionally, the learning rate was reduced by a factor of 0.3 when the validation loss plateaued for 5 epochs, with a minimum learning rate of 1×10^{-6}. The dropout, learning rate and early stopping parameters were initialized from commonly used defaults and refined through preliminary experiments, with early stopping following the validation-based criteria as described in Prechelt (2012).

2.3.3. Model evaluation

The metrics R^2, MAE, and MAPE were computed on the test set, providing a more reliable assessment of prediction accuracy and generalization. Saliency maps were also generated from test images using the fitted model, enabling qualitative inspection of learned features, visualization of the most influential image regions, and assessment of whether the model focuses on the animal when making predictions (Simonyan et al., 2014).

Model training and evaluation were conducted in a computational environment equipped with an NVIDIA NV167 GPU with 4 GB of VRAM, 16 GB of RAM, and an 11th generation Intel® Core™ i7-11390H processor. The implementation and evaluation of the proposed method were performed using Python 3.12 and Keras framework (version 3.8.0) using tensorflow-gpu (version 2.18.0) as the backend. For

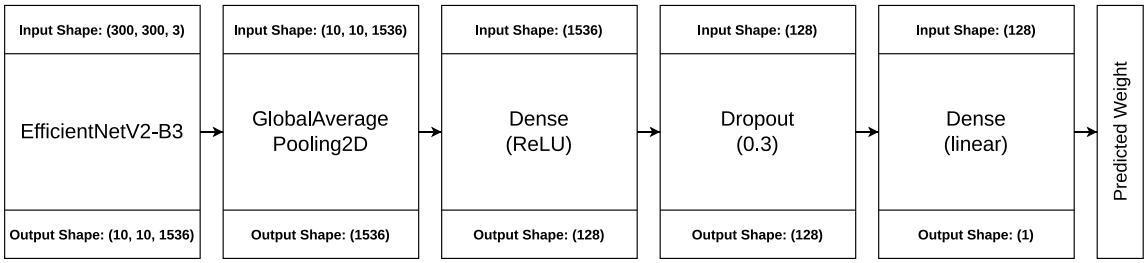


Fig. 6. Architecture of the proposed model. It consists of the pre-trained EfficientNet-B3 network without its output layer, augmented with four layers for regression: Global Average Pooling 2D, Dense (128 units, ReLU), Dropout (0.3), and Dense (1 unit, linear).

image processing, Pandas (version 2.2.3), NumPy (version 2.0.2), and scikit-image (version 0.25.0) libraries were used. The visualization of saliency maps was performed using the library tf-keras-vis (version 0.8.7). The entire pipeline is available online as supplementary material in Sousa et al. (2026c).

2.4. Performance analysis

This stage evaluates method performance under different training/test set arrangements designed to assess generalization across coat colors, farms, and weight ranges. First, a quantitative analysis was conducted using R^2, MAE and MAPE, and Pearson correlation between the measured and estimated weights. Next, a more detailed analysis examined intra-animal variability in estimated weight, error across weight ranges, and performance based on mean estimated weight. Finally, the influence of experimental, animal, and image-related factors on model performance and variability was assessed using GLMM. These analyses were complemented by a qualitative assessment of saliency maps generated in model evaluation stage.

2.4.1. Variability of estimated weight intra-animal

To assess intra-animal prediction variability, the prediction variability index (PVI) was computed across all executions. This index represents the mean CV of estimated weights from different images of the same animal (Eq. (8)). An ideal model exhibits low prediction variability.

$$PVI = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{\sigma_j}{\mu_j}\right) \tag{8}$$

where:

- PVI = model prediction variability index;
- N = number of animals evaluated;
- σ_j = standard deviation of estimated weights for images of animal j;
- μ_j = mean of estimated weights for images of animal j.

2.4.2. Error relevance by weight range

The relative error ($Erel$) was calculated from Eq. (9) and indicates how significant the prediction error is in relation to the measured weight. A good model should perform appropriately for all animals, regardless of their weight.

$$Erel_i = \frac{|\hat{Y}_i - Y_i|}{Y_i} * 100 \tag{9}$$

where:

- $Erel_i$ = relative error of the image $_i$;
- $\hat{Y}_i$ = the estimated weight for the image $_i$;
- Y_i = the measured weight of the image $_i$.

2.4.3. Performance using mean estimated weight

Performance metrics (R^2, MAE, and MAPE) were recomputed using a single estimate per animal (Eq. (10)), defined as the mean estimated weight, and the impact on results was analyzed.

$$\hat{Y}mean_j = \frac{\sum_{i=1}^{n_j}\hat{Y}_{ij}}{n_j} \tag{10}$$

where:

- $\hat{Y}mean_j$ = mean of the weights estimated for images of animal $_j$;
- $\hat{Y}_{ij}$ = estimated weight of image $_i$ and animal $_j$;
- n_j = number of images of animal $_j$.

2.4.4. Effects on model prediction and variability using GLMM

TrainingSize, Representativeness, MeanTrainingSetQI, Weight, UnexpectedBehavior, and ImageQI were modeled as fixed effects, with random effects accounting for repeated image captures of the same animal. These variables were defined to represent sources of variation related to the experiment, the animal, and the images. The first three capture differences between training and test conditions; Weight and UnexpectedBehavior describe animal-specific factors; and ImageQI reflects image noise levels. $Erel$ (Eq. (9)) and PVI (Eq. (8)) were defined as response variables to assess predictive performance and model variability, respectively. A Gamma error distribution with a log link was used to account for data asymmetry. Model selection also followed (Zuur et al., 2009).

Representativeness is defined as the number of training images within the same weight class as the predicted image. Due to gaps in the weight distribution and the limited number of animals above 55 kg (Fig. 3(b)), weight classes were defined using approximately 10 kg intervals, with wider upper ranges to mitigate data sparsity while preserving meaningful groupings: (0,10], (10,20], (20,30], (30,40], (40,55], and (55,99]. UnexpectedBehavior was modeled as a binary factor (True/False).

2.4.5. Qualitative analysis

The precision with which the models located the animal in the images across all executions was evaluated by visually inspecting the saliency maps and a corresponding binarized version, hereafter referred to as the Top-95% Saliency Mask, in which pixels with values above the 95th percentile are set to white. This binarized representation was used exclusively to highlight the similarity between the pattern formed by the most prominent pixels and the shape of the animal, thereby facilitating perceptual comparison of segmentation performance between executions. The 95th percentile threshold was defined empirically based on visual inspection.

3. Results

3.1. Dataset quality and similarity

3.1.1. Image quality analysis

As shown in Table 2, images from Farm A presented higher QI and QVI values than those from Farm B, indicating greater noise incidence and variability among images of the same animal. No strong correlation

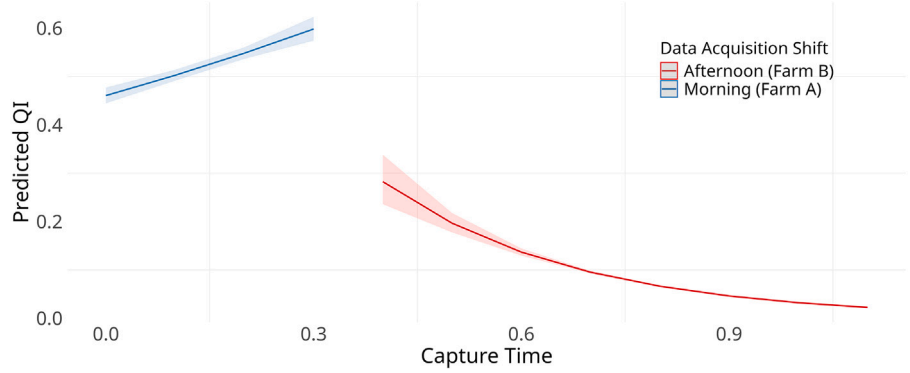


Fig. 7. Predicted QI values across capture time (minutes of the day) for each farm, obtained from the **QAM**. Lines represent model predictions, and shaded areas indicate 95% confidence intervals. Scales were normalized for model fitting; a value of 0.35 in *x*-axis represents midday.

Table 2
Mean values of QI and QVI computed for all images in the dataset. The Pearson column reports the correlations between QI and the animals' weight as well as the image capture time. QI values are on the order of 10^3.

Images	QI		QVI		Pearson (r)	
	Mean	CI95%	Value	CI95%	Weight	Time
All	4.25 ± 1.43	4.18–4.31	2.84	−15.00–97.90	0.14 ***	−0.92 ***
Farm A	3.91 ± 7.52	6.20–6.32	4.42	3.69–5.95	−0.24 ***	0.41 ***
Farm B	3.03 ± 1.04	3.32–3.34	1.01	0.93–1.10	8.32×10^{-4}	−0.37 ***

Significance codes: '***' 0.001 '**' 0.01 '*' 0.05.

Table 3
Summary of the three best-performing candidate QAM, including the error distribution (Family), model selection criteria (AIC and BIC), and the standard errors (SE) for the random effect. The arrows next to the metric names indicate the direction of best fit. Model equations are detailed in the table notes.

Model	Distribution	AIC↓	BIC↓	Random effect (SE)↓
A	Gamma (link = log)	−9203.7	−9165.3	0.92
B	Gamma (link = log)	−9204.0	−9171.1	0.92
C	Gamma (link = log)	−9210.9	−9172.6	0.89

A = *QI ~ 1 + Farm + CaptureTime + Weight + UnexpectedBehavior + (1|AnimalID).*
B = *QI ~ 1 + Farm + CaptureTime + Weight + (1|AnimalID).*
C = *QI ~ 1 + Farm * CaptureTime + Weight + (1|AnimalID).*

Table 4
Fixed-effect coefficients of the **QAM**. The table reports the estimated coefficients, standard errors (SE), t-values, and p-values for each predictor.

Fixed effect	Estimate	SE	T-value	P-value
Intercept	0.32	1.35	0.23	0.81
Farm B	−0.96	1.32	−0.72	0.46
CaptureTime	−3.62	1.42	−2.53	0.01 *
Weight	−0.49	0.16	−3.00	< 0.01 **
Farm B * CaptureTime	4.49	1.46	3.07	< 0.01 **

Significance codes: '***' 0.001 '**' 0.01 '*' 0.05.

was observed between animal weight and QI, either for the entire dataset or within farms. In contrast, QI was strongly correlated with image capture time; however, this relationship was negative when considering all images and showed opposite directions when analyzed separately by farm.

The three best-performing candidate models used to investigate the effects of farm, image capture time, animal weight, and unexpected behaviors in QI are presented in Table 3. Model **C**, hereafter referred to as the QI Analysis Model (QAM), showed the best fit based on the AIC, BIC, and estimated variance components of the random (animal) effect.

The estimated coefficients and their significance levels computed for fixed effects (Table 4) show that the interaction between the Farm and CaptureTime predictors presented a significant effect on QI (p-value < 0.01). Thus, the capture time affected QI differently between farms: for Farm A, QI increased over time, whereas for Farm B the opposite trend was observed. In this study, this behavior is attributed to the sunlight incidence during data acquisition. At Farm A, the images were captured in the morning, between 10:00 and 11:00, and the exposure to sunlight increased as the collection progressed. In contrast, on Farm B the images were captured in the afternoon, between 16:50 and 17:50, when the sunlight decreased over time. The influence of CaptureTime on QI for each data acquisition shift can be visualized in the effect plot shown in Fig. 7.

Furthermore, the predictor Weight also presented a significant effect on QI (p-value < 0.01), indicating that QI decreased as weight increased. UnexpectedBehavior had no significant effect on QI in candidate model **A**, and model fit improved when it was removed.

3.1.2. Image similarity analysis

Table 5 shows the similarity metric values (RMSE, SSIM, and pHash) calculated in all comparison settings. For the **general** setting, in which image pairs were randomly sampled from the entire dataset, the results indicate low similarity (SSIM < 0.85). This threshold is based on Meckbach et al. (2021). Regarding **intra-animal** comparison setting, the SSIM value demonstrate that the similarity between images of the same animal is also low. However, while SSIM suggests a lower intra-animal structural similarity for Farm A (0.37) than for Farm B (0.65), pHash shows the opposite behavior (Farm A = 4.11; Farm B = 14.56).

To understand this opposite behavior phenomenon, it is important to consider how these metrics operate. SSIM works in the spatial domain and evaluates how much small regions of the compared images (neighborhood windows) are occupied by similar elements (Wang et al., 2004). Therefore, it is more sensitive to changes in posture and spatial displacement of objects within the scene. In contrast, pHash operates in the frequency domain and is largely invariant to spatial shifts, as it measures the similarity of low-frequency patterns between images (Steinebach, 2012). Based on this, they are interpreted in this study as follows: SSIM indicates how perceptually similar two images are, reflecting the extent to which they may be considered visually identical, and pHash indicates the extent to which images share similar

Table 5
Mean values of the similarity metrics (RMSE, SSIM, and pHash) computed for all comparison settings. The arrows next to the metric names indicate the direction of increasing similarity.

Setting	SSIM↑		RMSE↓		pHash↓	
	Mean	CI95%	Mean	CI95%	Mean	CI95%
General	0.36 ± 0.17	0.36–0.36	7.37 ± 100.36	5.28–9.46	11.77 ± 15.47	11.45-12.10
Intra-animal *All*	0.57 ± 0.13	0.56–0.58	1.45 ± 35.88	−0.31–3.22	11.50 ± 15.38	10.74-12.26
Intra-animal *Farm A*	0.37 ± 0.04	0.37–0.38	4.89 ± 66.25	−1.16–10.94	4.11 ± 10.36	3.17-5.06
Intra-animal *Farm B*	0.65 ± 0.02	0.65–0.65	0.03 ± 0.00	0.03–0.03	14.56 ± 16.06	13.62-15.50
Intra-farm *All*	0.53 ± 0.15	0.53–0.53	2.90 ± 62.84	1.59–4.21	12.96 ± 15.86	12.63-13.29
Intra-farm *Farm A*	0.30 ± 0.05	0.30–0.31	9.17 ± 112.15	5.00–13.35	7.97 ± 13.77	7.45-8.48
Intra-farm *Farm B*	0.63 ± 0.25	0.63–0.63	0.04 ± 0.00	0.04–0.04	15.23 ± 16.23	14.83-15.64
Inter-farm	0.24 ± 0.03	0.24–0.24	6.63 ± 94.76	4.66–8.60	11.85 ± 15.42	11.52-12.17

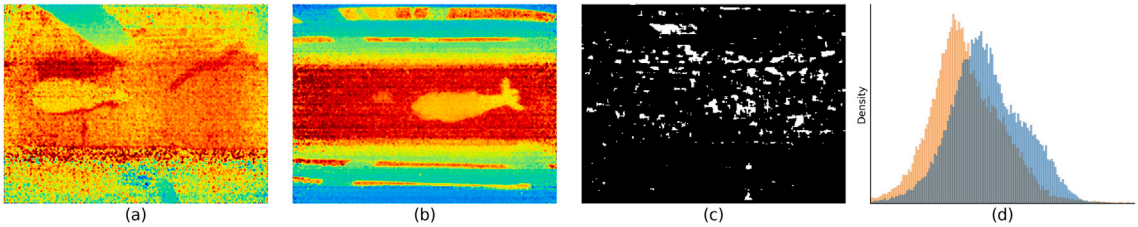


Fig. 8. Illustration of the SSIM and pHash comparison between two depth images. Images **(a)** and **(b)** show animals from Farms A and B, respectively. Image **(c)** shows the binarized SSIM map (threshold = 0.6), where white areas indicate regions considered similar. Image **(d)** presents the frequency density distribution of both images; legends are omitted to emphasize similar frequency patterns.

Table 6
Training and test sets composition after dataset partitioning. **Size** denotes the number of images. **Farm A** and **Farm B** indicate the percentage of images from each farm. **QI** denotes the average image noise level (values on the order of 10^3).

Run	Training sets					Test sets				
	Size	Farm A	Farm B	QI	CI95%	Size	Farm A	Farm B	QI	CI95%
run-1	1578	31.12%	68.88%	4.2 ± 1.4	4.1–4.3	194	32.47%	67.53%	4.3 ± 1.4	4.1–4.5
run-2	1497	30.46%	69.54%	4.2 ± 1.4	4.1–4.2	275	35.64%	64.36%	4.4 ± 1.5	4.1–4.5
run-3	1444	35.11%	64.89%	4.3 ± 1.4	4.2–4.4	328	14.33%	85.67%	3.7 ± 1.0	3.6–3.8
run-4	1550	28.45%	71.55%	4.1 ± 1.3	4.1–4.2	222	50.90%	49.10%	4.8 ± 1.5	4.6–5.0
run-5	1218	–	100.00%	3.3 ± 0.1	3.3–3.3	554	100.00%	–	6.2 ± 0.7	6.2–6.3
run-6	987	–	100.00%	3.3 ± 0.1	3.3–3.3	231	–	100.00%	3.3 ± 1.2	3.3–3.3
run-7	504	100.00%	–	6.2 ± 0.7	6.1–6.3	50	100.00%	–	6.4 ± 0.6	6.2–6.6
run-8	1542	26.50%	73.50%	4.1 ± 1.3	4.0–4.1	230	63.50%	36.50%	5.0 ± 1.4	4.8–5.2

elements or patterns, even when these elements appear in different spatial positions within the images. As a result, two images can be considered very different in the spatial domain but similar in the frequency domain, as illustrated by the images in Fig. 8.

Regarding **intra-farm** and **inter-farm** settings, the values shown in Table 5 also indicate low similarity among images of the same farm, as well as SSIM and pHash exhibit antagonistic behaviors for Farms A and B. However, it should be noticed that, when comparing the metrics calculated in the **inter-farm** setting with others comparison settings that contains images from both farms (**general**, **intra-animal (*All*)**, and **intra-farm (*All*)**) pHash values remain consistent, but SSIM decreases significantly. This finding reinforces the previous observation that the images exhibit structural spatial differences between farms, influenced by the environment, animal, and posture differences, but they share consistent and resilient elements and patterns with each other (Fig. 8).

3.2. Training and test sets configurations

Taking into account the first four executions, which used the entire dataset and partitioning was performed randomly, run-1 has the largest number of images in the training set, while run-3 contains the smallest (Table 6). Regarding the distribution of images between farms, their training sets show a similar pattern, with an average of 31.28% images from Farm A and 68.72% from Farm B. For the test sets, in runs 1 and 2 the distribution is similar to that observed for training sets. On the other hand, there is a predominance of images from Farm B in run-3 and a nearly equal distribution in run-4. In run-8, where the absence of images with weights between 20 and 35 kg was intentional, Farm B images are more prevalent in the training set, whereas Farm A images dominate the test set. Run-7 has the lower number of images in the training set (at most 504).

Regarding image quality, training set noise levels are similar across executions 1 to 4 and for run-8. Even so, run-3 shows the lowest average test set QI (3.7), whereas run-8 and run-4 shows the highest (5.0 and 4.8, respectively). The executions 5 and 6 show the lowest noise degrees for the training sets (3.3); however, they differ in the test sets: in run-5, the noise level is approximately twice that of the training set (QI = 6.2), whereas in run-6, it remains similar (QI = 3.3). Run-7 presents substantially higher incidence of noise than all other executions for both training (QI = 6.2) and test (QI = 6.4) sets.

The distribution of images by weight in the training and test sets are illustrated, respectively, in Figs. 9(a), and 9(b). In general, the training

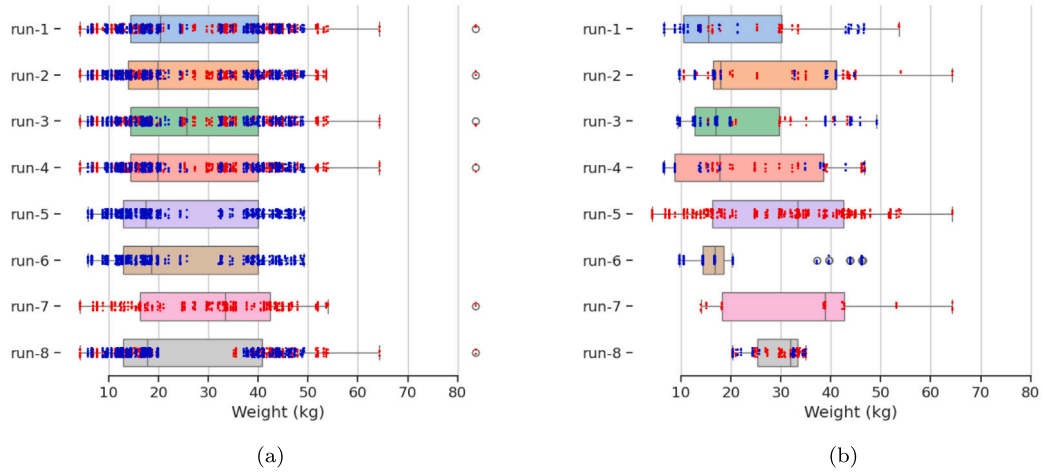


Fig. 9. Boxplots of images by weight for the **(a)** training and **(b)** test sets. Colored dots denote individual images; red and blue correspond to Farms A and B, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

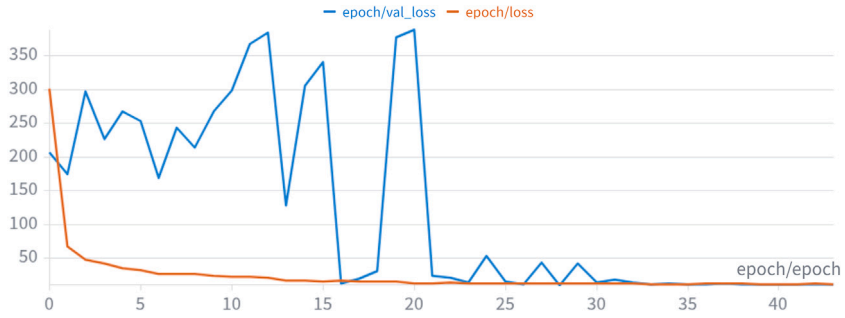


Fig. 10. Training (orange line) and validation (blue line) loss curves over epochs for execution run-1, showing rapid convergence of training loss and stabilization of validation loss after epoch 30. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

sets of all executions show a consistent pattern, with images (colored dots on the chart) evenly distributed within the 5 to 55 kg weight range, some weight gaps remain between 20 and 30 kg, and most of the images of the test set have similar weights in the corresponding training set. However, some deviations from this general behavior were observed: (i) the training set of run-2 does not include images with weight greater than 55 kg, whereas the corresponding test set contains images in this range; (ii) in run-5, the test set includes images at the extremes of the weight range (below 8 kg and above 50 kg) that are not represented in the training set; (iii) there are images corresponding to a breeding ram from Farm A with a weight greater than 80 kg (signed as an outlier on the chart), but they remain in the training set for all executions that used data from this farm, except for run-5.

3.3. Prediction performance

The model exhibited similar fitting patterns across all executions, as illustrated in Fig. 10: rapid convergence in training loss, with slightly slower stabilization on the validation set. Notably, no overfitting was observed in any execution, highlighting the effectiveness of the model architecture and learning strategies adopted in this study, including Dropout, Early stopping, and learning rate scheduling. Fig. 10 shows the loss curves for run-1, while all curves are available in Supplementary Materials A.19. As shown in Table 7, **run-7** — which had the smallest training set and the highest training QI — took longer to converge than the others (57 epochs).

Regarding performance, a substantial divergence between the training and test results was observed in **run-5**, **run-7**, and especially **run-8**, underlining the importance of relying on test-set metrics when assessing model accuracy and generalization. Therefore, considering only test-set metrics, the best results were achieved in run-1, with R^2, MAE, and MAPE equal to 0.95, 2.10 kg and 10.80%, respectively (Table 7). Runs 4 and 6 also achieved R^2 values similar to run-1, but with higher MAE and MAPE. The executions run-5, run-7, and especially run-8 presented the worst results. It can be also noticed that model performance for Farm B was better than for Farm A in all executions (Table 8).

Table 7
Prediction performance in all executions considering the metrics R^2, MAE, and MAPE, measured from the training and test sets.

Execution	Training set				Test set		
	Epoch	R^2	MAE	MAPE	R^2	MAE	MAPE
Run-1	43	0.93	2.47	9.73	0.95	2.10	10.80
Run-2	42	0.91	2.83	11.24	0.91	2.46	9.25
Run-3	34	0.92	2.83	10.88	0.91	2.46	12.81
Run-4	44	0.93	2.46	9.68	0.95	2.28	11.39
Run-5	31	0.91	2.79	10.65	0.66	6.71	37.95
Run-6	37	0.87	3.00	13.44	0.93	2.28	11.19
Run-7	57	0.83	3.64	11.61	0.61	7.22	22.98
Run-8	29	0.93	2.53	9.86	−0.56	5.22	17.47

3.3.1. Measured and estimated weight correlation

A graphical representation of model's goodness of fit in all executions is illustrated in Fig. 11. Runs 1, 4 and 6 achieved better fits in all weight ranges presenting the highest values of correlation (r = 0.98). Data points in these executions lie closer to the best-fit line (red line in the plot) and, in general, the amplitude between the highest and lowest estimated values for images of the same animal (points of the

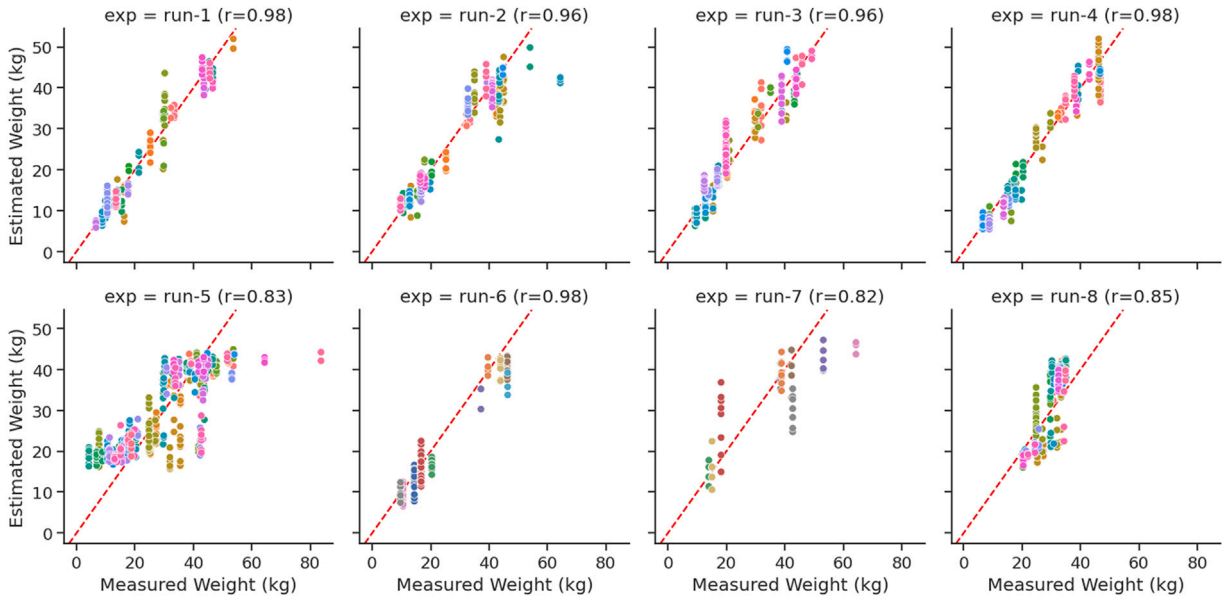


Fig. 11. Correlation plots between measured and estimated weights across all executions. Red dashed lines indicate the best-fit line, and the *r* value above each plot denotes the Pearson correlation. Points of the same color correspond to images of the same animal; colors are not consistent across runs. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

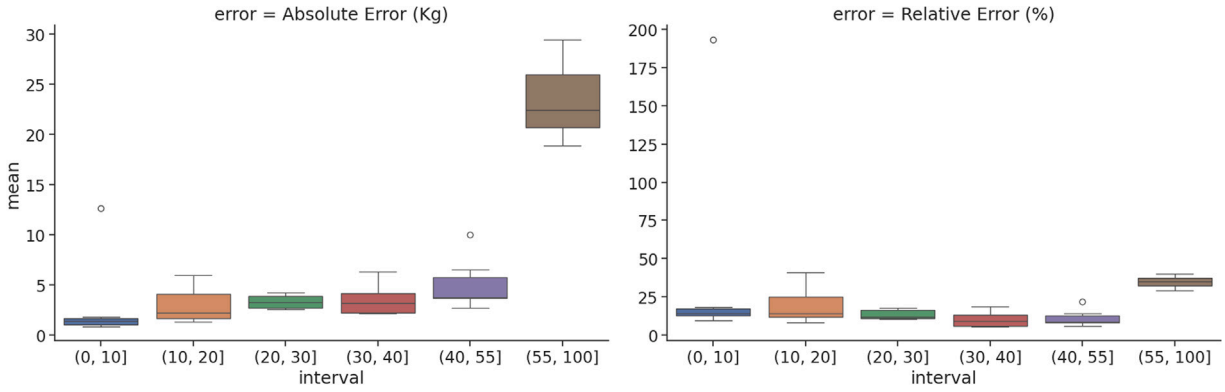


Fig. 12. Distribution of **(a)** absolute and **(b)** relative errors across the weight classes for all executions.

Table 8
Prediction performance for each farm in all executions considering the metrics R^2, MAE, and MAPE, measured from the test set.

Execution	Farm A			Farm B		
	R^2	MAE	MAPE	R^2	MAE	MAPE
Run-1	0.83	3.04	12.54	0.97	1.64	9.96
Run-2	0.79	4.28	12.75	0.97	1.45	7.31
Run-3	0.73	3.77	12.46	0.92	2.24	12.87
Run-4	0.91	2.79	9.99	0.97	1.75	12.84
Run-5	0.66	6.71	37.95	–	–	–
Run-6	–	–	–	0.93	2.28	11.19
Run-7	0.61	7.22	22.98	–	–	–
Run-8	−1.33	5.95	19.69	0.29	3.95	13.62

same color in each plot) is smaller. The models from **run-2**, and **run-7** exhibited a tendency to underestimate the weights of animals weighing more than 40 kg. In **run-5**, the fit is worse in the extremes of the weight range. In addition, most animals had estimated weights for different images both above and below the measured weight for runs 1 and 4.

3.3.2. Prediction error by weight range

Fig. 12 shows the distribution of prediction errors across weight classes and indicates that, as expected from scale effects, absolute error increases with animal weight (Fig. 12a). Conversely, relative error remains broadly consistent across classes, except for (55,99], with the lowest variability in the (30,40] and (40,55] intervals (Fig. 12b). As it removes scale dependence and enables unbiased comparison, subsequent analyses are based exclusively on relative error.

3.3.3. Main effects in prediction error

The three best-performing candidate models used to investigate the factors associated with relative error are presented in Table 9. Model **A**, hereafter referred to as the Error Analysis Model (EAM), showed the best fit based on AIC, BIC, and the estimated variance components of the random (animal) effect. The estimated coefficients and their significance levels computed by EAM to the fixed effects are reported in Table 10.

Representativeness (p-value < 0.1) and Weight (p-value < 0.001) have significant effects on relative error, contributing to its reduction as they increase. In contrast, ImageQI (p-value < 0.1) is positively associated with relative error. A significant interaction between TrainingSize and MeanTrainingSetQI indicates that the effect of the latter varies with the number of training images. As shown in Fig. 13, increases in MeanTrainingSetQI reduce error, with a more pronounced effect in larger training sets.

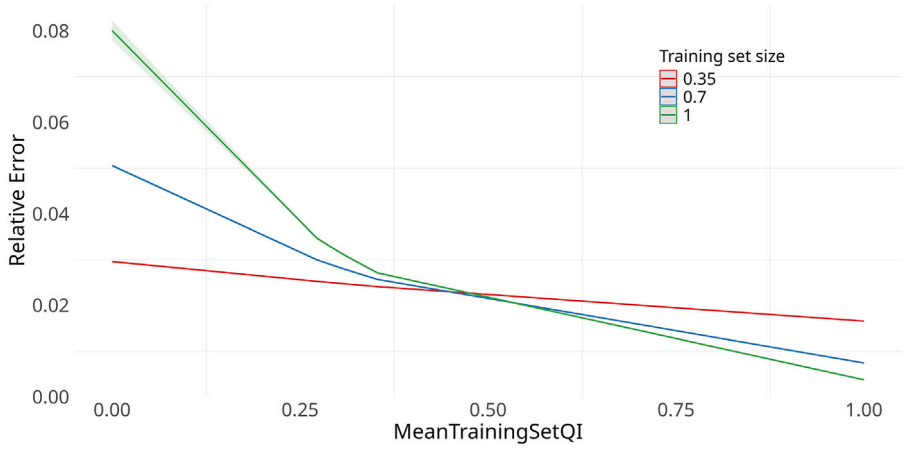


Fig. 13. Effect of MeanTrainingSetQI on relative error across training set sizes. Scales were normalized for model fitting; values of 0.35, 0.7, and 1 in the legend correspond to training sets of up to 550, 1269, and 1578 images, respectively.

Table 9
Summary of the three best-performing candidate EAM, including the error distribution (Family), model selection criteria (AIC and BIC), and the standard errors (SE) for the random effect. The arrows next to the metric names indicate the direction of best fit. Model equations are detailed in the table notes.

Model	Family	AIC↓	BIC↓	Random effect (SE)↓
A	Gamma (link = log)	−4074.0	−4023.7	0.44
B	Gamma (link = log)	−4066.6	−4016.3	0.45
C	Gamma (link = log)	−4066.5	−4016.2	0.46

A = *Erel ~ 1 + TrainingSize * MeanTrainingSetQI + Representativeness + ImageQI + Weight + (1|AnimalID).*
B = *Erel ~ 1 + TrainingSize + MeanTrainingSetQI * Representativeness + ImageQI + Weight + (1|AnimalID).*
C = *Erel ~ 1 + TrainingSize + MeanTrainingSetQI * ImageQI + Representativeness + Weight + (1|AnimalID).*

Table 10
Fixed-effect coefficients of the model **EAM**. The table reports the estimated coefficients, standard errors (SE), t-values, and p-values for each predictor.

Fixed effect	Estimate	SE	T-value	P-value
Intercept	−2.01	0.31	−6.33	<0.00 ***
TrainingSize	1.53	0.48	3.14	<0.00 **
MeanTrainingSetQI	0.76	0.34	2.21	0.02 *
Representativeness	−0.29	0.17	−1.69	0.08 .
ImageQI	0.34	0.21	1.65	0.09 .
Weight	−2.31	0.33	−6.94	<0.00 ***
TrainingSize * MeanTrainingSetQI	−3.84	0.92	−4.15	<0.00 ***

Significance codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1.

3.3.4. Variability of estimated weight intra-animal

Prediction variability across all runs is shown in Table 11. No clear relationship was observed with model performance (Table 7). Runs 1 and 6, despite strong predictive accuracy, exhibited among the highest PVI, lower only than run-7. In contrast, run-5, which presented poor predictive performance, had the lowest variability. Predictions for Farm B consistently exhibited lower variability across runs.

3.3.5. Main effects in prediction variability

The three best-performing candidate models used to investigate the factors associated with prediction variability are presented in Table 12. Model **C**, hereafter referred to as the Variability Analysis Model (VAM), showed the best fit based on AIC, BIC, and the estimated variance components of the random (animal) effect. The estimated coefficients and their significance levels computed by VAM to the fixed effects are reported in Table 13. The predictors MeanTrainingSetQI and Weight showed significant main effects on PVI (p-value < 0.001). Higher Training-set noise and lower Weight were associated with increased prediction instability.

A significant interaction between TrainingSize and ImageQI was observed. As shown in Fig. 14, the influence of ImageQI on variability was stronger in runs with larger and more heterogeneous training sets (runs 1 to 4 and run-8), resulting in a steeper rise in variability as ImageQI increased. For runs 5 and 6, whose training relied on data from a single farm, the effect of ImageQI was weaker but remained in the same direction. In contrast, ImageQI exhibited a negative association with variability in run-7, which had the smallest dataset. Representativeness and UnexpecteBehavior predictors were initially tested but did not show significant effects.

3.3.6. Performance using mean estimated weight

Performance metrics computed using the mean estimated weight show substantial improvements in runs 1, 3, 4, 7, and 8 (Table 14). For example, run-1 improved by 2% in R^2, 20% in MAE, and 37% in MAPE. In contrast, using the mean estimated weight increased errors in executions 2, 5, and 6.

3.4. Model segmentation inspection

Figs. 15–18 show saliency maps and top-95% saliency masks highlighting how precisely the model identified the animal region as the most correlated with body weight across executions. Based on qualitative inspection of representative images and supported by GLMM results, this section presents evaluation scenarios to examine the model's ability to localize the animal and estimate weight without explicit segmentation. The first three scenarios (Figs. 15–17) consider images with the highest relative error in the low-performance runs 5, 7, and 8 that appeared in the test set for multiple runs. The final scenario (Fig. 18) compares the localization ability of run-6 with a execution trained on a multi-farm dataset.

Fig. 15 presents images of a low-weight animal from Farm A (8.8 kg) in runs 4 and 5. Despite relatively high noise levels ($QI > 6.1$), run-4 achieved low relative errors ($E_{rel} < 3.1\%$), whereas run-5 performed poorly ($E_{rel} > 100\%$). In run-4, the most salient pixels are concentrated on the animal, forming a pattern consistent with its shape. In contrast, run-5 exhibits diffuse saliency, with higher relevance assigned to regions outside the animal. Notably, the model appears to focus more on the fence post than on the animal itself (as seen in the top-95% masks of run-5). Considering the GLMM findings (Table 10), the poor performance in run-5 is likely due to the smaller training set and exposure to a different background during training. Additionally, image noise may further hinder localization, as indicated by detached regions in the top-95% masks of run-4, particularly in the third image.

Table 11
Model prediction variability index (PVI) of all executions. Column **All Animals** indicates the PVI calculated from the entire dataset, while Farm A and Farm B denote PVI computed from each farm.

Run	Prediction variability index (PVI)					
	All animals	CI95%	Farm A	CI95%	Farm B	CI95%
run-1	9.79 ± 6.35	7.39–12.21	12.84 ± 7.26	8.44–17.22	7.32 ± 4.08	5.15–9.49
run-2	8.21 ± 5.27	6.16–10.25	9.73 ± 5.86	6.81–12.64	5.48 ± 2.05	4.00–6.94
run-3	8.59 ± 4.89	6.69–10.48	9.22 ± 4.98	5.87–12.56	8.19 ± 4.79	5.72–10.64
run-4	8.11 ± 4.53	6.35–9.87	8.62 ± 4.50	6.30–10.93	7.33 ± 4.47	4.33–10.33
run-5	6.83 ± 5.61	5.66–8.00	6.83 ± 5.61	5.66–8.00	–	–
run-6	9.58 ± 4.56	6.31–12.83	–	–	9.58 ± 4.56	6.31–12.83
run-7	14.32 ± 9.27	7.19–21.44	14.32 ± 9.27	7.19–21.44	–	–
run-8	8.20 ± 7.85	5.54–10.85	9.22 ± 8.53	5.69–12.73	5.88 ± 5.35	2.28–9.47

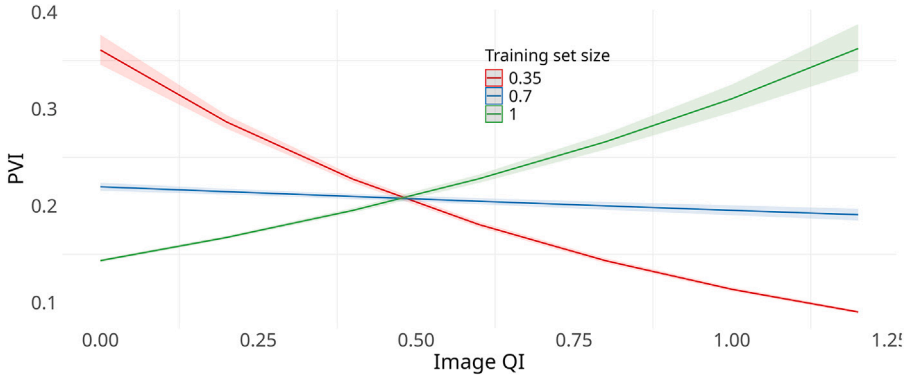


Fig. 14. Effect of ImageQI on prediction variability across training set sizes. Scales were normalized for model fitting; values of 0.35, 0.7, and 1 in the legend correspond to training sets of up to 550, 1269, and 1578 images, respectively.

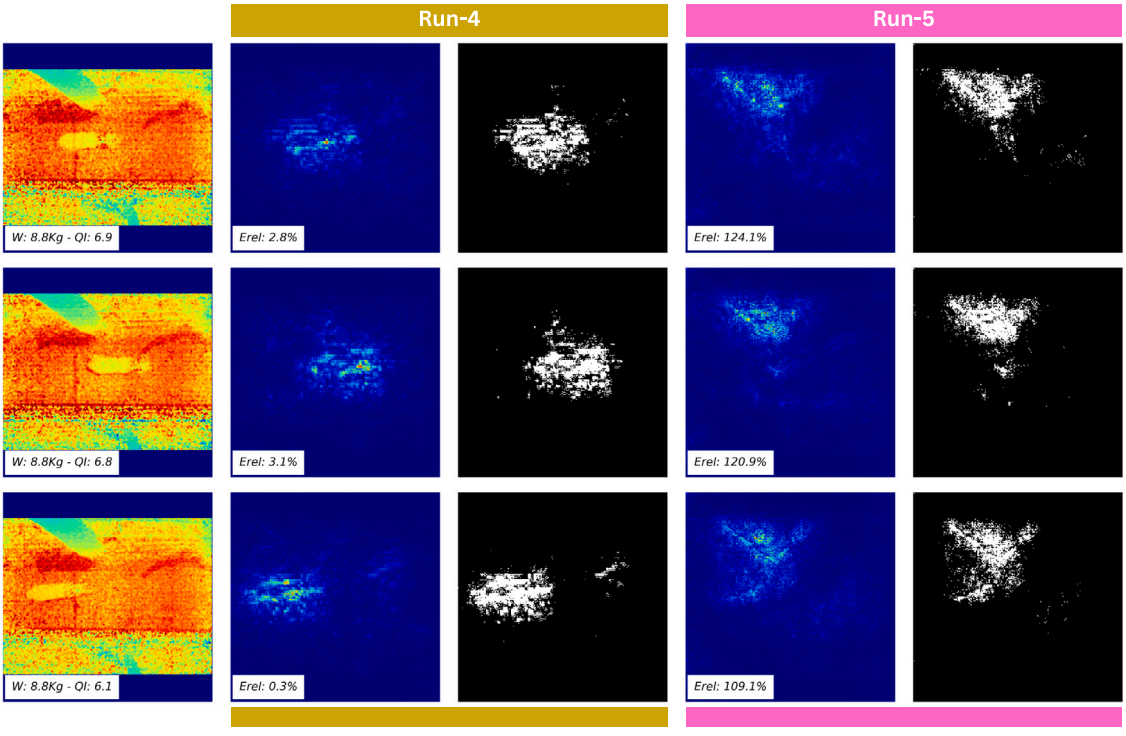


Fig. 15. Saliency maps (blue, with bright regions) and top-95% masks (black and white) for images of an animal of Farm A in runs 4 and 5. **W** and **QI** denote the animal's weight and the image quality index, respectively. **Erel** denotes the relative error for each image in the corresponding run. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 16 shows images of two animals from Farm A (15.1 kg and 38.9 kg) in runs 4, 5, and 7. Run-4 achieved the lowest relative error for all images and exhibited the most concentrated saliency, with top-95% masks that closely matching the animal's shape. Run-5 again emphasized the fence post in images 1 and 2, but achieved better localization and lower error for image 3. This aligns with GLMM findings, indicating that animal weight also influenced on model performance. Run-7, although showing better pixel concentration than run-5 did not

Table 12
Summary of the three best-performing candidate VAM, including the error distribution (Family), model selection criteria (AIC and BIC), and the standard errors (SE) for the random effect. The arrows next to the metric names indicate the direction of best fit. Model equations are detailed in the table notes.

Model	Family	AIC↓	BIC↓	Random effect (SE)↓
A	Gamma (link = log)	−4520.0	−4480.5	1.01
B	Gamma (link = log)	−4683.9	−4638.7	1.02
C	Gamma (link = log)	−4713.1	−4668.0	1.01

A = *PVI ~ 1 + TrainingSize + MeanTrainingSetQI + ImageQI + Weight + (1|AnimalID).*
B = *PVI ~ 1 + TrainingSize * MeanTrainingSetQI + ImageQI + Weight + (1|AnimalID).*
C = *PVI ~ 1 + TrainingSize * ImageQI + Weight + MeanTrainingSetQI + (1|AnimalID).*

Table 13
Fixed-effect coefficients of **VAM**. The table reports the estimated coefficients, standard errors (SE), t-values, and p-values for each predictor.

Fixed effect	Estimate	SE	T-value	P-value
Intercept	−0.10	0.15	−0.67	0.49
TrainingSize	−1.42	0.09	−15.32	<0.00 ***
Image QI	−2.19	0.19	−11.12	<0.00 ***
Weight	−2.20	0.34	−6.47	<0.00 ***
MeanTrainingSetQI	0.88	0.05	16.22	<0.00 ***
TrainingSize * Image QI	2.96	0.20	14.42	<0.00 ***

Significance codes: '***' 0.001 '**' 0.01 '*' 0.05.

Table 14
Performance metrics recalculated using the mean estimated weight. Percent values indicate comparison with results using the full dataset. Arrows denote the direction of improvement: higher R^2 and lower MAE and MAPE indicate better performance.

Run	R^2 ↑	MAE↓	MAPE↓
run-1	0.97 (+2%)	1.68 (−20%)	6.74 (−37%)
run-2	0.86 (−5%)	3.18 (+29%)	9.48 (+2%)
run-3	0.94 (+3%)	2.33 (−6%)	9.38 (−26%)
run-4	0.95 (+0%)	2.12 (−7%)	9.64 (−15%)
run-5	0.64 (−3%)	6.57 (−2%)	29.61 (−22%)
run-6	0.91 (−2%)	2.98 (+30%)	9.18 (−18%)
run-7	0.71 (+16%)	6.18 (−14%)	15.90 (−30%)
run-8	−0.13 (+23%)	4.65 (−10%)	15.74 (−10%)

outperform run-4, even when trained solely on data from the same farm (Farm A). Therefore, the model roughly localizes the animal, but the combination of limited samples and high QI in both training and test sets (Table 6) was insufficient for accurate weight estimation.

Fig. 17 presents images of an animal from Farm A (30.2 kg) with the highest relative error in run-8. Although no training images were available in the same weight range for run-8, the model still successfully localized the animal. Overall predictive performance was poor across executions, particularly in runs 5 and 8. Run-1 again exhibited a higher concentration of relevant pixels. Finally, Fig. 18 shows images of an animal from Farm B (16.7 kg) across runs 2 and 6. Localization and segmentation in run-6, trained only on Farm B data, are comparable to run-2, but with greater pixel concentration in run-2, highlighting the benefit of a larger and more diverse dataset. Notably, the animal appears inverted relative to the corridor direction and exhibits a twisted body posture in the images; nevertheless, the model successfully localizes it. Undoubtedly these scenarios are not exhaustive; therefore, all saliency maps and top-95% masks are available online as supplementary material in Sousa et al. (2026a).

4. Discussion

This study evaluates the accuracy and generalization of an end-to-end computer vision method for predicting sheep body weight from dorsal depth images, and investigates the factors that influenced prediction performance. The method, based on the EfficientNet-B3 architecture, automatically localizes the animal and estimates its weight without explicit segmentation or annotation. The dataset was collected from two livestock farms with animals in motion. Prediction accuracy was assessed using R^2, MAE and MAPE. Generalization across coat colors, farms, and weight ranges was evaluated through multiple training–test splits. Key sources of variation (experimental, animal, and image-related) were analyzed using a GLMM-based approach. Finally, saliency maps were inspected to verify whether the model accurately localizes animals and captures relevant traits.

4.1. Prediction performance

Given the consistent fitting patterns (Fig. 10; Supplementary Fig. A.19) and the achieved prediction performance (Tables 7 and 14), the proposed method can be considered capable of effectively identifying the animal region as most relevant to body weight, capturing its non-linear relationships, producing accurate estimates, and generalizing to unseen animals, provided it is exposed to data of sufficient quality and quantity. As shown in Figs. 15–18, even in cases with higher relative error or unusual postures (e.g., twisted body), the model primarily focuses on animal pixels, although salient regions extend beyond the animal in runs with a low amount of training data (notably runs 5 and 7). Therefore, the results suggest that the model captures true physical correlates of weight rather than noise.

The high accuracy achieved in runs 1 and 4 is consistent with recent ConvNet-based approaches for cattle and pigs and outperforms prior work on sheep (Table 15), advancing the state of the art in computer vision-based sheep weight estimation. Comparisons were limited to R^2 and MAPE, as these metrics are scale-independent and suitable across studies with different weight ranges. In addition, the variability in predictions for images of the same animal is low to moderate (Gomez and Gomez, 1984), indicating stable and reliable estimates. No comparison with related work was performed for this aspect, as prediction variability is rarely reported. These promising results are expected, given that ConvNet can automatically learn hierarchical feature representations — often outperforming manually designed pipelines (LeCun et al., 2015) — and that depth images precisely capture animal shape. Unlike RGB, depth images encode the distance between the camera and the surfaces in the scene, producing a geometric representation of the animal's body. As a result, structural characteristics such as body contour, volume distribution, and overall morphology can be preserved across images, enabling the detection of similar patterns even when spatial variations, postural changes, or environmental differences are present, such as demonstrated in Fig. 8.

4.1.1. Main effects in prediction performance and variability

The GLMM-based approach used to assess the effects of experimental (TrainingSize, Representativeness, MeanTrainingSetQI), animal (Weight, UnexpectedBehavior), and image-related (ImageQI) factors on relative error and prediction variability helped explain the substantial differences in model performance across runs (Table 7). Representativeness — defined as the number of training images within the same weight range as the predicted sample — proved to be an important factor in improving prediction accuracy, as relative error decreases with increasing representativeness (Table 10). However, it did not significantly affect prediction variability. These findings suggest that the training set should include images with geometric patterns (i.e., shape) similar to those of the target animals. Animal weight also influences model performance, with larger animals associated with lower relative error and reduced variability. This indicates that the model has greater difficulty segmenting and estimating the weight of smaller animals, particularly under high-noise conditions and when training data are limited.

Nevertheless, the factor with the greatest impact on performance across runs was image noise, which in this study is associated with sunlight incidence during image acquisition (Table 4). The mean training-set QI had a significant effect on prediction stability, increasing variability as it rises, which explains the higher variability observed in

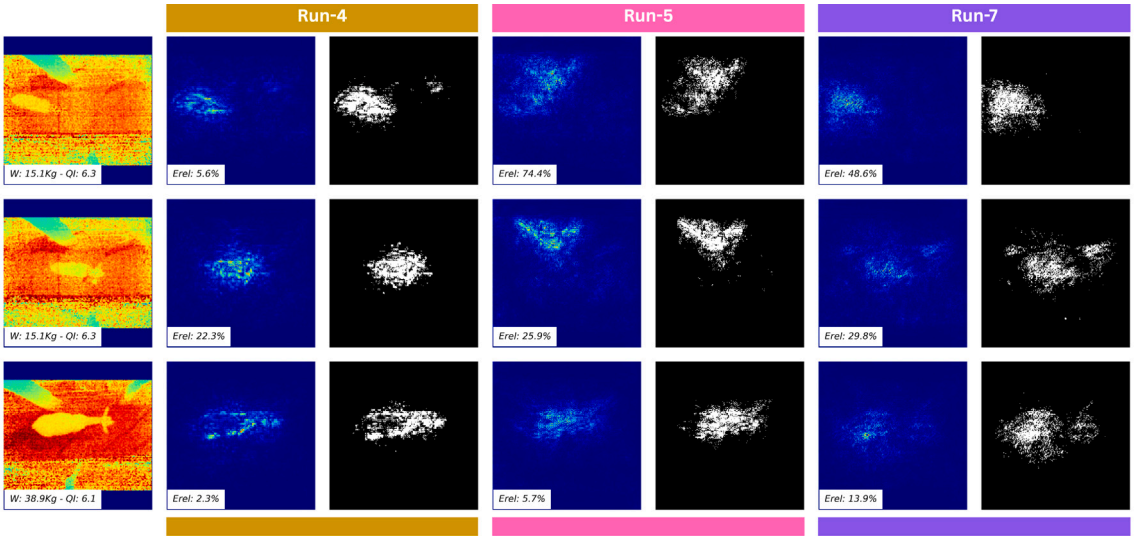


Fig. 16. Saliency maps (blue, with bright regions) and their corresponding top-95% saliency mask (black and white) for three images of an animal of Farm A in runs 4 (gold bar), 5 (pink bar), and 7 (purple bar). In the depth images, **W** and **QI** denote the animal's weight and the image quality index, respectively. In the saliency maps, **Erel** denotes the relative error for the image in the corresponding execution. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

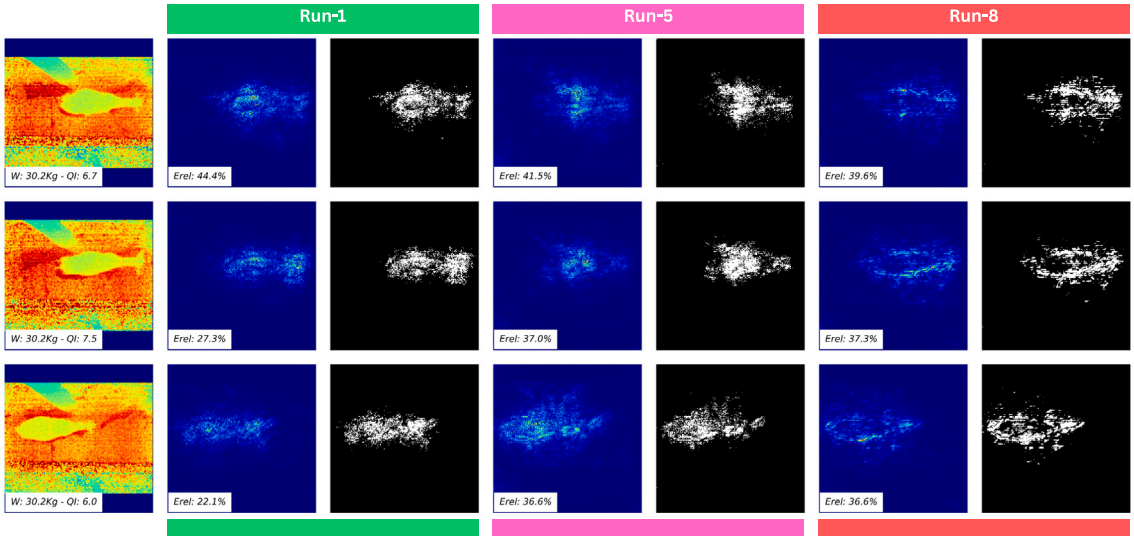


Fig. 17. Saliency maps (blue, with bright regions) and top-95% masks (black and white) for images of an animal of Farm A in runs 1, 5, and 8. **W** and **QI** denote the animal's weight and the image quality index, respectively. **Erel** denotes the relative error for each image in the corresponding run. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 15
Comparison of studies applying ConvNets to predict animal body weight from images. The **Cross-farm** column indicates whether multiple farms are used. Arrows next to R^2 and MAPE denote the direction improvement.

Related work	Device	Image type	$R^2\uparrow$	MAPE↓	Cross-farm	Species
Meckbach et al. (2021)	RealSense D435i	Depth	0.97	3.9%	✘	Pig
Hou et al. (2023)	IFM O3D303	Point Cloud	–	3.2%	✘	Cattle
Afridi et al. (2024)	RealSense D415	RGB/Depth	0.97	3.79%	✓	Cattle
Liao et al. (2025)	RealSense D435	Depth	0.99	2.34%	✓	Calf
Marques et al. (2025)	Kinect V2	RGB	0.92		✓	Sheep
This study (**run-1**) *mean estimated weight*	MS-A075V 3D	Depth	0.97	6.74%	✓	Sheep
This study (**run-4**) *mean estimated weight*	MS-A075V 3D	Depth	0.95	9.64%	✓	Sheep
This study (**run-1**) *all images*	MS-A075V 3D	Depth	0.95	10.80%	✓	Sheep
This study (**run-4**) *all images*	MS-A075V 3D	Depth	0.95	11.39%	✓	Sheep

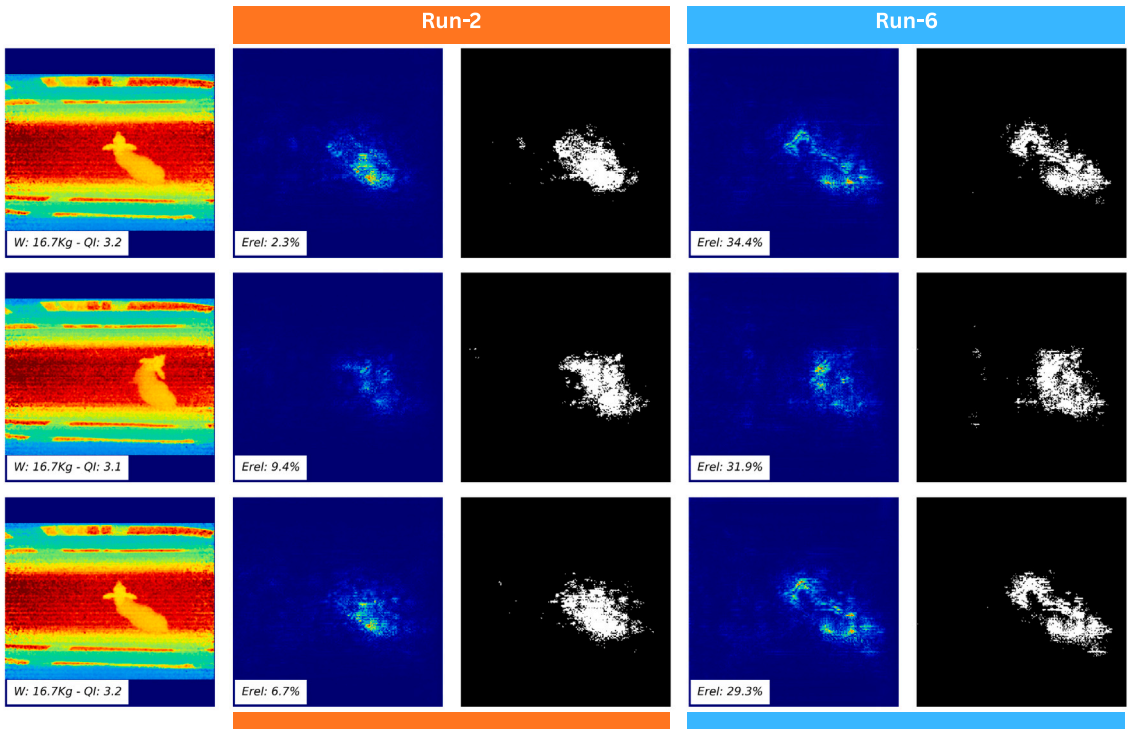


Fig. 18. Saliency maps (blue, with bright regions) and top-95% masks (black and white) for images of an animal of Farm B in runs 2 and 6. **W** and **QI** denote the animal's weight and the image quality index, respectively. **Erel** denotes the relative error for each image in the corresponding run. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

runs trained on data from both farms (runs 1–4 and 8). Regarding predictive performance, its interaction with TrainingSize is noteworthy. Fig. 13 shows that relative error decreases as MeanTrainingSetQI increases, with a stronger effect for larger training sets. This effect is interpreted not as a consequence of noise — despite what the variable name suggests — but primarily as a result of increased diversity in environments and animals. Accordingly, performance improves as data from both farms are incorporated into the training set, despite the higher noise levels (runs 1–4 and 8). Similarly, the noise level of the predicted image also contributed to higher relative error (Table 10) and prediction variability (Table 13). Regarding variability, its interaction with TrainingSize is also noteworthy, indicating that ImageQI increases prediction instability more strongly in larger training sets (Fig. 14).

Hence, training set size is intrinsically important: it has an additive effect with representativeness, mitigates the impact of image noise on variability, and amplifies the effect of environmental and animal diversity on predictive performance. Notably, unexpected behaviors (i.e., attempts to turn back or run) showed no significant effect on relative error, variability, or image noise.

4.1.2. Prediction performance between farms

Performance and variability were worse for Farm A than for Farm B mainly due to lower image quality, as Farm A exhibits higher noise levels (Table 2). However, as noted earlier, performance for Farm A animals improved when the training set included data from both farms, compared to runs trained on a single farm (**run-5** and **run-7**). This suggests that incorporating data from Farm B enhanced the model's predictive capability, even for Farm A animals. Conversely, performance for Farm B remained relatively stable across all executions, regardless of whether the training data included images from both farms or only from Farm B, indicating that lower-quality data from Farm A did not adversely affect accuracy for Farm B.

4.1.3. Mean estimated weight as a strategy to improve performance

Using the mean estimated weight across all images of a sheep improved model performance, particularly in runs 1 and 4 (Table 14). These findings suggest that leveraging multiple images capturing postural variation can be as effective as using a single image of a stabilized animal. Moreover, this approach is less invasive, as images are acquired while animals are in motion, facilitating practical adoption in livestock farms. However, averaging was effective only in runs where the model produced both over- and underestimates for the same animal, indicating that its adoption should be carefully evaluated.

4.2. Model generalization analysis

The results also suggest that the model learned meaningful patterns from the training set and generalizes to previously unseen samples. However, the sources of variation considered in this study should be analyzed separately. Based on the performance of models trained on the full dataset and the effectiveness of depth images in representing animal shape — regardless of color — it can be inferred that the model generalizes across different coat colors and patterns. This is supported by the dataset composition, which includes mixed-breed animals from multiple farms, acquired in open environments and in varied postures. Regarding generalization to unseen weight ranges, the model fails to extrapolate beyond the training distribution. In **run-2**, where the training set contains no images in the 60–70 kg range while the test set does (Fig. 9(b)), errors are high, indicating poor performance on unseen patterns. This suggests the method generalizes primarily to animals with shapes similar to those observed during training. A similar pattern is observed in **run-5**, which performs substantially worse at the extremes, where training data are absent; **run-8** shows similarly poor performance. These results (Fig. 11) highlight a limitation of end-to-end approaches compared to methods that decouple body measurement and weight estimation. For example, Liao et al. (2025) show that ConvNet-based body measurements from early images can predict weight at later stages. Overall, the poor performance on unseen weight ranges indicates limited extrapolation capacity beyond the training data. Finally, the model performed poorly when generalizing across farms (e.g., run-5), indicating that exposure to both animal weight patterns and farm-specific environmental conditions is necessary.

4.3. Practical considerations for real-world deployment

The present findings indicate that knowledge learned from one farm can be transferred to another, even when the target farm includes animals with different coat colors or mixed breeds. However, strong predictive performance in a new farm depends on two main factors: (i) the presence of animals with shapes similar to those observed during training, and (ii) a calibration stage in which samples from the new herd are incorporated, allowing the model to adapt to its specific characteristics. Incorporating more patterns and environments into a single model tends to create a network effect, whereby all users benefit as new data are added. As the dataset grows, the number of samples required for each new deployment is expected to decrease, reducing time and effort — particularly for small herds — by eliminating the need to build a fully dedicated dataset for each location. However, based on the data available in this study, it is not possible to determine whether calibration will always be required for each new farm or whether the model could eventually be deployed without this step. In any case, a key advantage of this end-to-end approach is that, even when calibration is needed, the required effort is reduced, as no manual image annotation is necessary and segmentation is performed directly by the model.

Additionally, the influence of sunlight on image quality limits the method's applicability in outdoor environments, particularly in regions with high temperatures or variable illumination. As a result, its use may require infrastructure adaptations — especially in small farms — to prevent direct exposure of the device to sunlight. Therefore, evaluating illumination correction methods, such as those proposed in Li and Teng (2022), is necessary.

4.4. Limitations

The main limitation of this study is the dataset size. Although the images exhibited low spatial similarity and depth data accurately captured animal shape, the limited sample size restricted the inclusion of additional predictors in the GLMMs — such as body condition score and wool level — and prevented convergence of three-way interactions. Moreover, although the dataset includes images of mixed-breed and multi-colored sheep, their phenotypic characteristics are predominantly similar to Dorper and/or Santa Inês hair breeds. Consequently, it does not include wool breeds such as Merino, Rambouillet, or Corriedale, nor does it account for variation related to shearing status. Incorporating more phenotypically diverse data could improve model generalization and the reliability of the results.

Image noise — particularly in Farm A — introduced variability into the analyses and affected both segmentation and estimation performance. Therefore, environmental factors during image acquisition — such as lighting, shading, and farm-specific conditions — should be investigated under controlled settings to better assess their impact on depth image quality. The qualitative visual assessment used to remove inadequate images from the dataset is another important limitation. As the dataset grows, manual verification becomes impractical and necessitates automated procedures (e.g., a frame selection script). Finally, reliance on top-view depth images makes model performance highly sensitive to unexpected behaviors such as jumping and running.

4.5. Future works

Future work will evaluate the impact of data enhancement techniques — particularly depth image restoration — and the integration of attention mechanisms on prediction performance and the accuracy of animal localization. Future studies should also incorporate data from breeds with varying degrees of wool coverage to better assess model robustness and adaptability to broader phenotypic diversity. In addition, the model's computational performance will be evaluated on lightweight, low-power devices to support the development of solutions suitable for real-world adoption by rural producers.

5. Conclusions

Computer vision-based techniques for predicting animal body weight provide an indirect and non-invasive alternative for estimating livestock weight that reduces the labor and time required for weighing and supports more frequent monitoring of animals. However, methods for predicting the weight of sheep from images are still focused on traditional image processing techniques and suffer from low accuracy and generalization. This study proposes an efficient ConvNet method based on the EfficientNet-B3 baseline network model and evaluates, quantitatively and qualitatively, its performance in predicting the body weight of sheep from images. The results are encouraging and show that the proposed model, operating in an end-to-end approach and combined with cross-farm dorsal depth images, achieved an accuracy higher than that of the current studies with sheep (R^2 = 0.97 and MAPE = 6.74%) and a promising generalization capability. Despite the rigorous evaluation of generalization, the dataset should include more phenotypic and environmental diversity to further improve the robustness of the findings. In future works, some novel data enhancement techniques and adjustments in the model architecture will be evaluated, as well as the computational performance of the method will be assessed on lightweight and low-power devices, with the goal of enabling real-world products that can be effectively adopted by livestock farmers. Overall, this work advances our understanding on developing more efficient and non-invasive methods for animal weighing in livestock farms and provides a basis for practical application and further research.

CRediT authorship contribution statement

Ewerton Costa Sousa: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ricardo de Andrade Lira Rabelo:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization. **José Lindenberg Rocha Sarmento:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Formal analysis, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT (OpenAI) to improve the clarity and language of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded by Brazilian National Council for Scientific and Technological Development (CNPq), Brazil under Grant No. 307967/2022-0 (“Development of Intelligent Architecture for Applications on Internet of Robotic Things, Brazil”), No. 407274/2021-9 (”Strategic Solutions for Aerial-Terrestrial IoT Network Architectures, Brazil“), and No. 406734/2022-4 (“Meat Production Chain”), and by Ministério da Integração e do Desenvolvimento Regional (MIDR) - Coordenação-Geral de Sistemas Produtivos Inovadores (in Portuguese), Brazil under Grant Termo de Execução Descentralizada No. 942052/2023 - 2023NS001587 (“Desenvolvimento e Difusão de Tecnologias

para Melhoria da Produção de Carne nos Rebanhos Caprino e Ovino na Chapada Vale do Itaim"). The authors thank Colégio Técnico de Teresina (CTT/UFPI), represented by Antonio de Sousa Junior, PhD, the owners of livestock farms for allowing the acquisition of images used in this experiment, and all the friends who helped during the collection of the images, especially Divino Rodrigues, João Batista Rodrigues, Igor Tawan, Jaquebede Batista, Bruno Felipe, and Jarbson Santana.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.smallrumres.2026.107799.

References

Abdelhady, A.S., Hassanien, A.E., Awad, Y.M., El-Gayar, M., Fahmy, A., 2019. Automatic sheep weight estimation based on K-means clustering and multiple linear regression. In: Hassanien, A.E., Tolba, M.F., Shaalan, K., Azar, A.T. (Eds.), Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2018. Springer, Cham, pp. 546–555.

Afridi, H., Ullah, M., Nordbø, Ø., Hoff, S.C., Furre, S., Larsgard, A.G., Cheikh, F.A., 2024. Analyzing data modalities for cattle weight estimation using deep learning models. J. Imaging 10 (3), 72. http://dx.doi.org/10.3390/jimaging10030072.

Atkinson, L., Doyle, R.E., Jongman, E.C., 2024. Temperament behaviours in individually tested sheep are not related to behaviours expressed in the presence of conspecifics. Animals 14 (1), 155. http://dx.doi.org/10.3390/ani14010155.

Bates, D., Mächler, M., Bolker, B., Walker, S., 2015. Fitting linear mixed-effects models using lme4. J. Stat. Softw. 67 (1), 1–48. http://dx.doi.org/10.18637/jss.v067.i01.

Bo, H.X., Huong, N.T., Hue, D.T., Luc, D.D., 2023. Modeling growth curves to estimate the suitable slaughter age for the Vietnamese tap Na pig. Livest. Sci. 272, 105233. http://dx.doi.org/10.1016/j.livsci.2023.105233.

Buchner, J., 2020. ImageHash: Perceptual image hashing for python. URL: https://github.com/JohannesBuchner/imagehash. (Accessed 2026).

Cerqueira, V., Santos, M.R.d., Roque, L., Baghoussi, Y., Soares, C., 2024. Online data augmentation for forecasting with deep learning. http://dx.doi.org/10.48550/arXiv.2404.16918, arXiv preprint arXiv:2404.16918.

Chen, M., Li, H., Zhang, Z., Ren, R., Wang, Z., Feng, J., Cao, R., Hu, G., Liu, Z., 2025. Real-time pig weight assessment and carbon footprint monitoring based on computer vision. Livest. Sci. 15 (17), http://dx.doi.org/10.3390/ani15172611.

Dohmen, R., Catal, C., Liu, Q., 2022. Computer vision-based weight estimation of livestock: a systematic literature review. N. Z. J. Agric. Res. 65 (2–3), 227–247. http://dx.doi.org/10.1080/00288233.2021.1876107.

Gomez, K.A., Gomez, A.A., 1984. Statistical Procedures for Agricultural Research, second ed. John Wiley & Sons, New York.

Gonzalez, R.C., Woods, R.E., 2018. Digital Image Processing, Global Edition, fourth ed. Pearson, Harlow; Essex; England.

Goodfellow, I., Bengiccco, Y., Courville, A., 2016. Deep Learning. MIT Press, http://www.deeplearningbook.org.

Gurgel, A., Difante, G., Emerenciano Neto, J., Santana, J., Dantas, J., Roberto, F., Campos, N., Costa, A., 2021. Use of biometrics in the prediction of body weight in crossbred lambs. Arq. Bras. Med. Vet. Zootec. 73 (1), 261–264. http://dx.doi.org/10.1590/1678-4162-12087.

Hamming, R.W., 1950. Error detecting and error correcting codes. Bell Syst. Tech. J. 29 (2), 147–160. http://dx.doi.org/10.1002/j.1538-7305.1950.tb00463.x.

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., et al., 2020. Array programming with NumPy. Nature 585, 357–362. http://dx.doi.org/10.1038/s41586-020-2649-2.

Hou, Z., Huang, L., Zhang, Q., Miao, Y., 2023. Body weight estimation of beef cattle with 3D deep learning model: PointNet++. Comput. Electron. Agric. 213, 108184. http://dx.doi.org/10.1016/j.compag.2023.108184.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444. http://dx.doi.org/10.1038/nature14539.

Li, K., Teng, G., 2022. Study on body size measurement method of goat and cattle under different background based on deep learning. Electronics 11 (7), 993. http://dx.doi.org/10.3390/electronics11070993.

Liao, M., Morota, G., Bi, Y., Cockrum, R.R., 2025. Predicting dairy calf body weight from depth images using deep learning (YOLOv8) and threshold segmentation with cross-validation and longitudinal analysis. Animals 15 (6), 868. http://dx.doi.org/10.3390/ani15060868.

Marques, J.V.M., Santos, W.S., Araújo, R.L., Sarmento, J.L.R., Silva, R.R.V., 2025. Non-invasive and low-cost approach for estimating sheep weight using computer vision. Eng. Rep. 7 (8), e70329. http://dx.doi.org/10.1002/eng2.70329.

McCulloch, C.E., Searle, S.R., 2001. Generalized, Linear, and Mixed Models. Wiley.

Meckbach, C., Tiesmeyer, V., Traulsen, I., 2021. A promising approach towards precise animal weight monitoring using convolutional neural networks. Comput. Electron. Agric. 183, 106056. http://dx.doi.org/10.1016/j.compag.2021.106056.

OECD/FAO, 2025. OECD-FAO Agricultural Outlook 2025–2034. Food and Agriculture Organization of the United Nations and Organisation for Economic Co-operation and Development, Rome and Paris, http://dx.doi.org/10.1787/601276cd-en.

Prechelt, L., 2012. Early stopping — But when? In: Montavon, G., Orr, G.B., Müller, K.-R. (Eds.), Neural Networks: Tricks of the Trade: Second Edition. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 53–67. http://dx.doi.org/10.1007/978-3-642-35289-8_5.

Sant'Ana, D.A., Pache, M.C.B., Martins, J., Soares, W.P., de Melo, S.L.N., Garcia, V., de Moares Weber, V.A., da Silva Heimbach, N., Mateus, R.G., Pistori, H., 2021. Weighing live sheep using computer vision techniques and regression machine learning. Mach. Learn. Appl. 5, 100076. http://dx.doi.org/10.1016/j.mlwa.2021.100076.

Simonyan, K., Vedaldi, A., Zisserman, A., 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. http://dx.doi.org/10.48550/arXiv.1312.6034, arXiv preprint arXiv:1312.6034.

Sipeed, 2022. MaixSense-A075V: a 3D TOF camera module with RGB. URL: https://wiki.sipeed.com/hardware/en/maixsense/maixsense-a075v/maixsense-a075v.html. (Accessed 17 May 2025).

Sousa, E.C., Rabelo, R., Sarmento, J., 2026a. A computer vision approach for predicting sheep body weight in livestock farms: Supplementary files v1.0.0 (version 1.0.0) [image]. Zenodo. http://dx.doi.org/10.5281/zenodo.19372960.

Sousa, E.C., Rabelo, R., Sarmento, J., 2026b. A computer vision approach for predicting sheep body weight in livestock farms: Supplementary files v1.0.0 (version 1.0.0) [software]. zenodo. http://dx.doi.org/10.5281/zenodo.19372131.

Sousa, E.C., Rabelo, R., Sarmento, J., 2026c. A computer vision approach for predicting sheep body weight in livestock farms: Supplementary files v1.0.0 (version 1.0.0) [software]. zenodo. http://dx.doi.org/10.5281/zenodo.19372263.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.

Steinebach, M., 2012. Robust hashing for efficient forensic analysis of image sets. In: Gladyshev, P., Rogers, M.K. (Eds.), Digital Forensics and Cyber Crime. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 180–187.

Tan, M., Le, Q.V., 2020. EfficientNet: Rethinking model scaling for convolutional neural networks. http://dx.doi.org/10.48550/arXiv.1905.11946, arXiv preprint arXiv:1905.11946.

Tu, G.J., Jørgensen, E., 2023. Vision analysis and prediction for estimation of pig weight in slaughter pens. Expert Syst. Appl. 220, 119684. http://dx.doi.org/10.1016/j.eswa.2023.119684.

van der Walt, S.J., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T., 2014. Scikit-image: Image processing in python. PeerJ 2, e453. http://dx.doi.org/10.7717/peerj.453.

Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process. 13 (4), 600–612. http://dx.doi.org/10.1109/TIP.2003.819861.

Zhang, H., Zhang, Y., Niu, K., He, Z., 2025. Neural network-based method for contactless estimation of carcass weight from live beef images. Comput. Electron. Agric. 229, 109830. http://dx.doi.org/10.1016/j.compag.2024.109830.

Zhao, Y., Xiao, Q., Li, J., Tian, K., Yang, L., Shan, P., Lv, X., Li, L., Zhan, Z., 2023. Review on image-based animals weight weighing. Comput. Electron. Agric. 215, 108456. http://dx.doi.org/10.1016/j.compag.2023.108456.

Zuur, A.F., Ieno, E.N., Walker, N.J., Saveliev, A.A., Smith, G.M., 2009. Mixed Effects Models and Extensions in Ecology with R. Springer, New York, pp. 127–129. http://dx.doi.org/10.1007/978-0-387-87458-6.

5 A MULTI-VARIANT EVALUATION OF ENERGY AND COMPUTATIONAL EFFICIENCY OF A CONVNET-BASED AIOT EDGE COMPUTING SYSTEM FOR SHEEP WEIGHING

A Multi-variant Evaluation of Energy and Computational Efficiency of a ConvNet-based AIoT Edge Computing System for Sheep Weighing

Ewerton Costa Sousa[a,c,*,1], Gabriel L. Bastos[b], Ricardo de Andrade Lira Rabelo[b] and José Lindenberg Rocha Sarmento[a]

[a]Department of Animal Science, Federal University of Piauí, Campus Petrônio Portela, Teresina, 64049-550, Piauí, Brazil
[b]Department of Computer Science, Federal University of Piauí, Campus Petrônio Portela, Teresina, 64049-550, Piauí, Brazil
[c]Department of Computer Science, Federal Institute of Piauí, Campus Angical, Angical, 64410-000, Piauí, Brazil

ARTICLE INFO

Keywords:
Artificial intelligence of things (AIoT)
Edge computing
Precision livestock farming (PLF)
Multi-Agent System (MAS)
Weight prediction
Sheep

ABSTRACT

Recent research on vision-based animal weighing systems, a trending topic in Precision Livestock Farming (PLF), has substantially improved prediction accuracy, particularly with Convolutional Neural Network (ConvNet) approaches. However, deploying these systems on farms remains challenging due to technological limitations in rural environments. Edge Computing has the potential to become an effective solution to these challenges, enabling applications within the Artificial Intelligence of Things (AIoT) paradigm. Nevertheless, despite continuous hardware advances, AIoT edge devices still face resource constraints that may be incompatible with the high computational demands of recent ConvNet-based weight estimation methods. Therefore, this study assesses the deployment feasibility, computational efficiency, and energy consumption of CVbSWP, an edge AIoT computer vision application for automated sheep weighing in smart farming environments. Four CVbSWP variants were developed by combining single-stream and batch-stream deployment strategies with sequential and parallel multiagent-based paradigms. Performance was evaluated using the metrics System throughput, Response latency, Inference time ratio, Top 95th mean CPU utilization, Top 95th mean RAM utilization, Power consumption total, and Effective FPS. The results demonstrate the feasibility of deploying the system on a Raspberry Pi 5, representing a worst-case deployment scenario, achieving a System throughput of 276 animals per hour in the parallel variants.

1. Introduction

Precision Livestock Farming (PLF) applications enable real-time monitoring of animals, equipping them with sensors, and provide data-driven decision support that enhances livestock output, reduces costs, improves product quality, and mitigates environmental impacts [1]. In this field, vision-based animal weighing systems can be considered a trend research topic, since through images it is possible to measure traits that can be used to estimate animal weight. Most recent research on vision-based animal weight prediction methods has achieving better accuracy over the years, especially those using Convolutional Neural Network (ConvNet) approaches, such as Sousa et al. [2], Li and Teng [3], Hou et al. [4], Afridi et al. [5], Liao et al. [6], Zhang et al. [7], Meckbach et al. [8], Marques et al. [9].

However, deploying these systems on farms remains a challenge, as the rural environment still presents obstacles such as the lack of power or connectivity across the farm [10]. In many cases, transmitting the collected images for processing on more powerful computers or even in a cloud environment can lead to high latency problems or even be unfeasible, since images are more dimensional data than information collected by traditional sensors. A promising alternative is the adoption of Edge Computing [11], where data is processed on local devices rather than transmitted over a network, enabling solutions within the Artificial Intelligence of Things (AIoT) paradigm. AIoT integrates Artificial Intelligence (AI) models into Internet of Things (IoT) infrastructures and is particularly beneficial for PLF, as it enables on-field autonomous decision-making, local processing of high-dimensional data such as images, and mitigation of latency and connectivity issues associated with cloud-based solutions in rural environments [12].

Nevertheless, AIoT edge devices, although increasingly powerful, still have limitations related to processing, storage, power availability, and network connectivity that may be incompatible with the potentially high computational costs of weight monitoring ConvNet-based methods. Moreover, despite advances in improving the accuracy of image-based weight prediction methods, their computational efficiency remains largely overlooked. Therefore, this study aims to assess the deployment feasibility, computational efficiency, and energy consumption of CVbSWP, an edge AIoT ConvNet-based system for automated sheep weight monitoring, in PLF environments.

The main contributions of this work are: i) the proposal and evaluation of a complete edge AIoT computer vision system for automated sheep weighing, whereas most prior studies focus primarily on prediction accuracy; ii) a comparative analysis of single-stream and batch-stream deployment strategies and sequential and multiagent-based processing paradigms; and iii) the demonstration of system feasibility on Raspberry Pi 5 hardware without dedicated AI acceleration.

The next sections are organized as follows: The materials and procedures used in this study are detailed in Section

*Corresponding author. E-mail address: ewertoncosta@ifpi.edu.br
ORCID(s): 0000-0003-4695-454X (E.C. Sousa); 0009-0007-4232-9620 (G.L. Bastos); 0000-0003-1482-6404 (R.d.A.L. Rabelo); 0000-0002-4215-1515 (J.L.R. Sarmento)

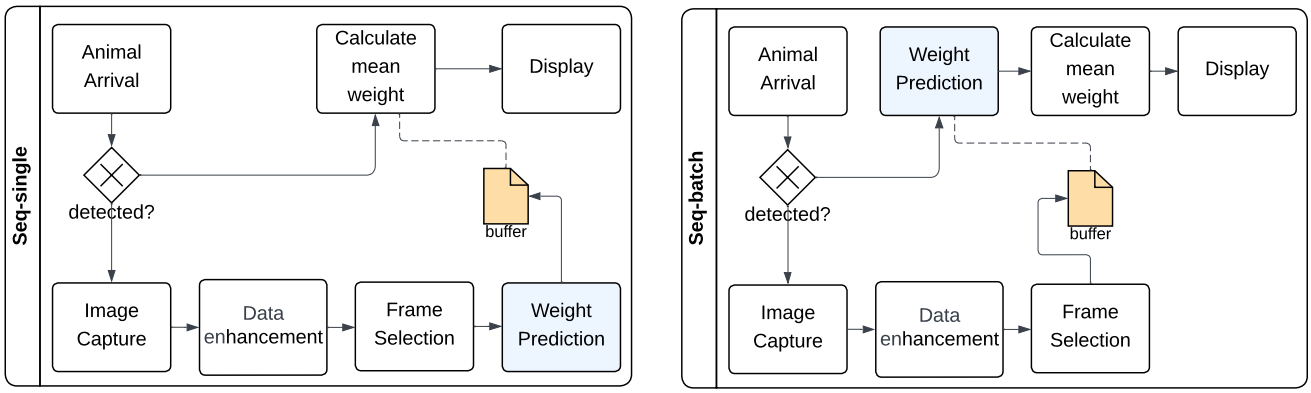


Figure 1: Main elements of the CVbSWP variants developed following the sequential paradigm for (a) single-stream and (b) batch-stream strategies.

2. The results are presented in Section 3, and the practical implications of the investigation are discussed in Section 4. The main conclusions of the study are presented in Section 5.

2. Material and Methods

2.1. System design and variants

The main modules of the CVbSWP reference architecture are shown in Fig. 1. The weighing process starts when module **Animal Arrival** detects an animal and notifies **Image Capture** to begin image acquisition. Each captured frame is processed by **Data Enhancement** for image enhancement or preprocessing and then forwarded to **Frame Selection** to assess its suitability for prediction (e.g. complete animal visibility and acceptable noise levels). These steps are repeated while the animal remains exposed to the camera. After the animal completes its passage, the average weight estimated from all collected frames is computed and sent to **Display**.

Based on the reference architecture, four variants of CVbSWP were developed and evaluated. These variants differ in deployment strategy (single-stream and batch-stream) and in module processing and communication paradigm (sequential and parallel multiagent-based).

2.1.1. Deployment strategies

Deployment strategies are inspired by Reddi et al. [13] and define how images of the same animal are processed by the Weight Prediction module. Two strategies were implemented: **single-stream** and **batch-stream**. In the single-stream strategy, images are individually sent to the model as soon as they are captured, and the average weight is computed after all predictions are completed (Fig. 1a). In the batch-stream strategy, captured images are temporarily stored in a buffer and, after the animal completes its passage, all images are simultaneously sent to the model for prediction, followed by average weight computation. The differences between both strategies are further illustrated in pseudo-algorithms 1 and 2.

2.1.2. Module processing and communication paradigms

In sequential paradigm, each system module is executed only after completion of the previous module. As an illustrative example, in the single-stream strategy (Fig. 1a), after animal detection, the Image Capture, Data Enhancement, Frame Selection, and Weight Prediction modules are sequentially executed in a loop. When Animal Arrival indicates that the animal is no longer exposed to the camera, the average weight is computed and forwarded to Display to complete the weighing process. In this arrangement, communication between modules occurs through direct method calls between system modules. Considering that sequential pipelines leads to under utilization of the available hardware, the parallel multiagent-based paradigm, hereafter referenced by Multi-Agent System (MAS), employs the non-blocking principle and run in an asynchronous design, enabling the system to running more than one module simultaneously. This paradigm relies on event-driven reactive agents [14], which respond either to the capture of a new frame or to messages from other agents indicating that a frame is ready for processing.

MAS variants were developed using the PADE framework [15], chosen for being built in Python, which provides access to an extensive ecosystem of AI libraries, native support for FIPA-ACL (Foundation for Intelligent Physical Agents - Agent Communication Language) messaging, and its foundation on the Twisted event-driven framework. Additionally, although the application runs on a single device, the use of these frameworks naturally supports future evolution toward distributed computing, as message routing is determined by the destination IP address and port.

The Adapter design pattern [16] was adopted in the implementation to promote decoupling and to ensure that the same underlying processing modules, covering capture, enhancement, and inference, are shared between the sequential and the asynchronous strategies. Therefore, each agent corresponds to a distinct stage of the pipeline: Image Capture Agent, Data Enhancement Agent, Frame Selection Agent, and Weight Prediction Agent. All agents are initialized within the Single Board Computer (SBC), each assigned to

Algorithm 1 Evaluate model performance using a single-stream strategy

Require: $herd_size, images_per_animal, interval$
Require: $frame_selection_time, frame_selection_ratio$
Require: $deployment_scenario$

$model \leftarrow$ LOADMODEL(path) ▷ Load model to memory
$n \leftarrow 0$
while $n < herd_size$ **do**
  $n \leftarrow n + 1$
  $W \leftarrow []$ ▷ Initialize an empty list W
  $i \leftarrow 0$
  while $i < imgs_per_animal$ **do**
    $i \leftarrow i + 1$
    $img \leftarrow$ LOADIMG(img)
    $is_adequate \leftarrow$ FRAMESELECT(i)
    if $is_adequate$ **then**
      $w_i \leftarrow model.predict(img)$
      Append w_i to W
    else
      $continue$
    end if
  end while
  $w_{mean} \leftarrow avg(W)$
  $display(w_mean)$
  $sleep(interval)$ ▷ wait for next animal
end while
function FRAMESELECT($image_index$)
  $qtt \leftarrow images_per_animal * ratio$
  $mean \leftarrow images_per_animal/2$
  if $image_index >= qtt - mean$ & $image_index <= qtt + mean$ **then return** true
  elsereturn false
  end if
  $sleep(frame_selection_time)$
end function

Algorithm 2 Evaluate model performance using a batch-stream strategy

Require: $herd_size, images_per_animal, interval$
Require: $frame_selection_time, frame_selection_ratio$
Require: $deployment_scenario$

$model \leftarrow$ LOADMODEL(path) ▷ Load model to memory
$n \leftarrow 0$
while $n < herd_size$ **do**
  $n \leftarrow n + 1$
  $I \leftarrow []$ ▷ Initialize an empty image list I
  $i \leftarrow 0$
  while $i < imgs_per_animal$ **do**
    $i \leftarrow i + 1$
    $img \leftarrow$ LOADIMG(img)
    $is_adequate \leftarrow$ FRAMESELECT(i)
    if $is_adequate$ **then**
      Append img to I
    else
      $continue$
    end if
  end while
  $W \leftarrow model.predict(I)$
  $w_{mean} \leftarrow avg(W)$
  $display(w_mean)$
  $sleep(interval)$ ▷ wait for next animal
end while
function FRAMESELECT($image_index$)
  $qtt \leftarrow images_per_animal * ratio$
  $mean \leftarrow images_per_animal/2$
  if $image_index >= qtt - mean$ & $image_index <= qtt + mean$ **then return** true
  elsereturn false
  end if
  $sleep(frame_selection_time)$
end function

a dedicated port. Frame acquisition begins once the Image Capture Agent receives a confirmation message indicating that both the Frame Selection Agent and the Weight Prediction Agent have loaded the model into memory. From that point, the capture and processing pipeline runs continuously. The overall application flow mirrors that of the sequential strategy, with the key difference that instead of a synchronous while loop, agents communicate through ACL messages using the INFORM performative to notify the next stage that a new frame is available. Each message carries a frame_id and an animal_id, allowing the prediction stage to correctly associate each image with its corresponding animal batch.

Since all processing occurs locally on the SBC with no inter-device network traffic, embedding full frames directly in ACL messages would introduce unnecessary serialization and deserialization overhead. To avoid this, an in-memory frame buffer was adopted, implemented as a Python dictionary that stores each frame's image data (numpy.ndarray) indexed by its frame_id. Agents communicate solely by passing frame identifiers, reading from and writing to a shared global FRAME_BUFFER. A frame enters the buffer when captured by the Image Capture Agent and is released either by the Frame Selection Agent upon rejection, or by the Weight Prediction Agent upon completion of inference, in either single or batch strategy. The message flow between

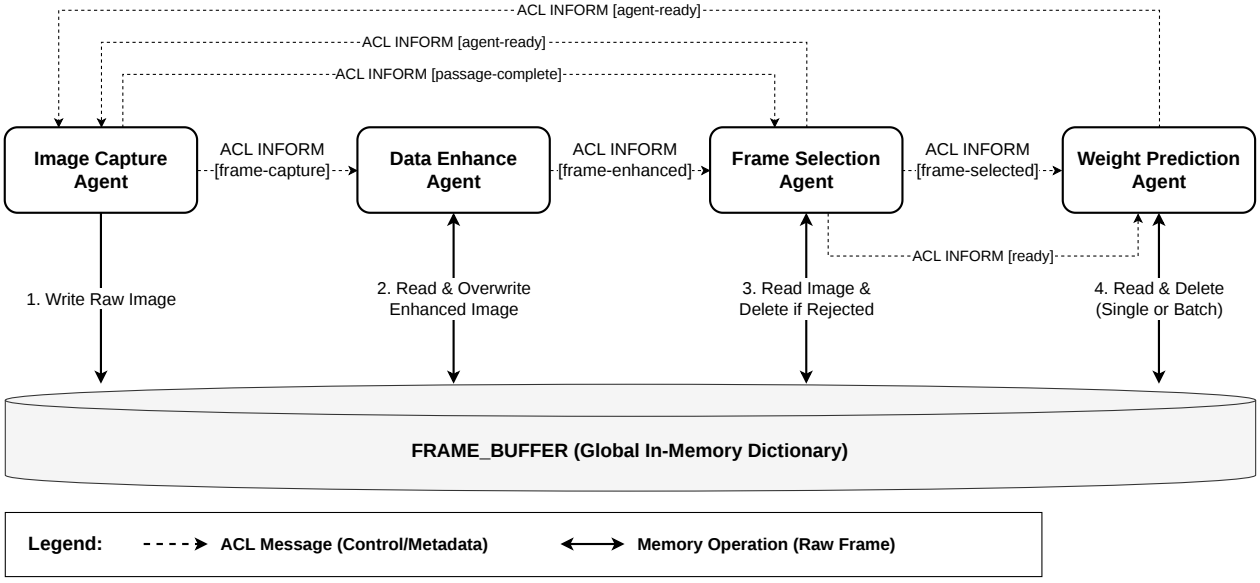


Figure 2: Main elements of the CVbSWP variants developed following the MAS paradigm.

agents and the corresponding read/write operations on the FRAME_BUFFER are illustrated in Fig. 2.

A challenge inherent to the MAS paradigm concerns both deployment strategies: the Weight Prediction Agent must know precisely when all selected frames for a given animal have been forwarded by the Frame Selection Agent, so that the final weight can be computed over the complete set. To address this, an event-counting synchronization mechanism was implemented:

1. Image Capture Agent notifies the Frame Selection Agent of the total number of frames captured for that animal.
2. The Frame Selection Agent tracks each processed frame, regardless of whether it was accepted or discarded.
3. Only when the count of processed frames reaches the expected total and all selected frame identifiers have been forwarded to the Weight Prediction Agent does the Frame Selection Agent send a control message signaling that no further frames will arrive for that animal.

Upon receiving this signal, the Weight Prediction Agent computes the final weight estimate as the arithmetic mean of all individual predictions for that animal. In batch strategy, inference is performed over the complete set of selected frames at once, with the mean computed afterward. In single strategy, predictions are performed individually as frames arrive and accumulated in memory, with the mean computed only upon receiving the synchronization signal. This design ensures statistical parity between both strategies, as the final estimate always reflects the complete set of selected frames regardless of the deployment strategy.

2.2. Evaluation scenario

The four variants were evaluated based on a simulated baseline scenario that resembles a real-world data collection environment where farmers weigh a group of animals (a flock or the entire herd) sequentially. This scenario took into account behaviors observed during previous weighings, which are represented in the constraints shown in Table 1. Since the method adopted in this work weighs one animal at a time, it is assumed that farmers must exercise control to ensure that only one animal is exposed to the camera. No real camera was used in the simulation, therefore the Image Capture module was implemented by loading an image stored on the device. Every time the module was executed, the image was loaded from storage, making the processing as costly as possible.

Modules Frame Selection and Weight Prediction were implemented to simulate a pipeline in which both adopt ConvNet-based methods. The Weight Prediction module uses a model originally proposed and trained in Sousa et al. [2] and its output corresponds to the actual prediction generated by the model. The same model was also employed in module Frame Selection; however, its output was based only on the behavior described in Table 1 for variable Frame Selection Window. Therefore, the model was used solely to ensure that processing time and resource consumption accurately reproduced the computational patterns of ConvNet-based models. Regarding module Data Enhancement, the same processing was performed as in the original work [2]: (i) noise removal, (ii) normalization, (iii) channel replication, and (iv) image resize.

Constraint	Meaning	Value
Herd Size	Number of animals to be weighed.	500
Passage Time	The time the animal is exposed to the camera.	10 seg
Arrival Time	Time between the end of weighing one animal and the detection of the next.	3 seg
Frame Selection Window	Time interval in which the animal is fully represented in the images. Only images captured within this interval are labeled as suitable by the Frame Selection module.	3 seg
FPS	The capture rate was set to 1.0 FPS based on the number of frames processed by the sequential strategies, which captured approximately 9 to 11 frames per animal within the defined passage time. At 1.0 FPS, the agent-based architecture captures exactly 10 frames per animal during the 10-second passage window, ensuring a comparable number of frames across all evaluated architectures and enabling a fair assessment of the deployment strategies.	1.0

Table 1
Parameters that define a simulated evaluation scenario. The column **Value** denotes the values used to assess the performance of the four system configurations, named baseline scenario.

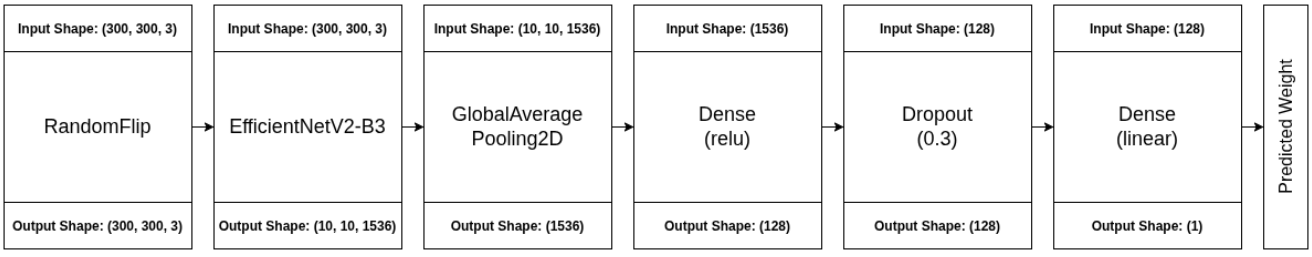


Figure 3: The architecture of the proposed model. It is composed of all layers of the pre-trained EfficientNet-B3 network, except the output layer, augmented with four new layers that aim to estimate the regression output: Global Average Pooling 2D, Dense (128 units and relu activation function, Dropout(0.3) and Dense (1 unit and linear activation function). The input to the network consists of images with a resolution of (300, 300, 3). This Figure was originally published in Sousa et al. [2].

2.3. Experimental setup

2.3.1. Model architecture

The weight prediction model, originally proposed in Sousa et al. [2], achieved a promising accuracy and generalization, with an R^2 of 0.97, MAE of 1.68 kg, and MAPE of 6.74%, and consists of a straightforward and efficient 2D ConvNet model based on the EfficientNet-B3 pre-trained baseline network with parameters established from the ImageNet data set. Its architecture is composed of all layers of the EfficientNet-B3 network, except the output layer, augmented with four new layers that aim to estimate the regression output (Fig. 3).

2.3.2. Device

The experiments were conducted on a Raspberry Pi 5 (16 GB RAM), running Raspberry Pi OS Lite to reduce resource overhead during execution. This board was selected for having sufficient processing capacity to run model inference on CPU, while also representing one of the best available options in terms of cost-effectiveness and regional availability in the Brazilian market. Among single-board computers that meet the minimum processing requirements for this task, the Raspberry Pi 5 sits at the lower end of the performance spectrum, as more capable alternatives equipped with dedicated AI accelerators and GPUs, such as the NVIDIA Jetson series, which offer greater computational performance but at significantly higher acquisition costs and with limited availability in the region. This makes the Raspberry Pi 5 a representative worst-case deployment scenario: if the proposed architecture operates efficiently on this hardware, performance gains are expected on more capable devices.

2.3.3. CPU and memory utilization measurement

CPU and memory metrics were collected at a one-second sampling interval using the psutil library (v7.2.2). To ensure the integrity of the performance evaluation, the monitoring routines were executed in independent threads, and therefore did not compete for resources with the core application workload, where capture, selection, and inference take place. The same monitoring mechanism was applied uniformly across all evaluated variants, allowing for an unbiased comparison of resource consumption between the sequential and the MAS paradigm.

2.3.4. Energy consumption measurement

Energy consumption was measured using a TC66C USB Type-C Tester (firmware v1.18), which monitors voltage, current, and power in real time. Data acquisition was automated through a Python script adapted from [17], which dispatches a read command to the device via serial port at a one-second polling interval. The evaluation was conducted with both PWR and PD switches enabled, and the TC66C was powered independently through its serial connection to avoid interference with the measurements. Instantaneous power readings (P_i, in Watts) were extracted directly from the device. Total energy consumption (E, in Watt-hours) was then computed by numerically integrating the power over the experiment duration, according to Eq. 1.

$$E = \sum_{i=1}^{N} P_i \cdot \Delta t_i \tag{1}$$

where:

- Δt_i represents the time interval in hours between consecutive measurements.

2.4. Performance evaluation

The deployment feasibility and the computational and energy efficiency of the system were evaluated across variants using the following metrics. Response latency and System throughput are more higher level metrics and can be interpreted as business metrics that closer to the user understanding. Others metrics are also meaningful, but more technical and focused in measuring the computational efficiency and energy consumption. It should be noted that the device was restarted before each execution.

- **Response latency (RLT)**: denotes the mean time required for the system to produce the weight estimation of an animal. It was measured as the mean elapsed time between the Image Capture and Display modules considering the last image captured from each animal, which approximates the moment the animal exits the corridor. This metric reflects the system's real-time capability and determines the minimum interval between consecutive animals during operation.
- **System throughput (STH)**: represents the average number of animals weighed per hour. This metric can be used to assess return on investment (ROI) and compare the proposed system with other weighing strategies.
- **Inference time ratio (ITR)**: denotes the fraction of total execution time occupied by model inference. The inference time is calculated as the sum of the processing times of the Weight Prediction module of all animals.
- **Inference throughput (ITH)**: represents the number of images processed by the Weight Prediction module per unit of time.
- **Top 95th mean CPU utilization ($\mathbf{CPU}_{95th}$)**: denotes the mean CPU utilization for values above the 95th percentile. As the Raspberry Pi has four processing cores, CPU utilization at a given moment corresponds to the mean utilization across all cores. This metric indicates CPU consumption during peak usage periods. Higher values suggest lower computational efficiency and an increased risk of system slowdown or instability.
- **Top 95th mean RAM utilization ($\mathbf{RAM}_{95th}$)**: indicates the mean RAM utilization–the percentage of memory in use at a given moment– for values above the 95th percentile. This metric indicates RAM consumption during peak usage periods. Higher values suggest lower computational efficiency and an increased risk of system slowdown or instability.
- **Power consumption total (PWT)**: represents the total energy consumed during system operation, measured in watts/hour. Lower energy consumption increases the feasibility of using off-grid and mobile power solutions, such as power banks and solar panels, for weighing operations in remote environments without access to the electrical grid.
- **Effective FPS (FPS)**: denotes the effective image processing rate of the system, measured in frames per second (FPS). This metric reflects the actual image processing capacity achieved during operation and can be used to assess the system's capability to process image streams in real time. Higher FPS values indicate greater processing efficiency and support higher animal flow rates during weighing operations.

3. Results

Evaluation metrics computed for each variant are shown in Table 2. Variants are named as SEQ-S and SEQ-B for the sequential processing with single- and batch-stream deployment, respectively. The same notation applies to MAS-S and MAS-B. Sequential variants took longer to complete the weighing of all animals and, consequently, achieved a lower system throughput (SEQ-S=258 and SEQ-B=246) compared to the MAS variants (both equals to 276). Regarding response latency, SEQ-S achieved substantially lower response time than the other variants. However, opposite behaviors were observed when comparing variants with the same processing paradigm. In the MAS variant, MAS-S required a slightly more time to estimate animal weight than MAS-B. In contrast, among the sequential variants, SEQ-B took approximately 2.3× longer than SEQ-S.

SEQ-S yielded the lowest number of suitable frames per animal (2.0), a consequence of the direct forwarding of captured frames to the Weight Prediction module, which reduces the number of images in which the animal is fully visible in the scene. SEQ-B, in turn, captured a higher number of suitable frames per animal (3.1), though at the cost of higher resource consumption peaks (CPU, RAM, and

Config.	Images	STH	Dur.	RLT (s)	ITR (%)	ITH	CPU_{95th} (%)	RAM_{95th} (%)	PWT (Wh)	FPS
SEQ-S	4496 (2.0)	258	01:56:01	1.01	12.5	1.14	39.5 ± 0.8	6.3 ± 0.1	10.12	0.65
SEQ-B	5133 (3.1)	246	02:01:46	2.32	9.1	2.31	53.5 ± 3.2	10.1 ± 0.6	10.87	0.70
MAS-S	5000 (3.0)	276	01:48:32	2.71	87.5	0.26	47.5 ± 0.5	19.9 ± 0.1	10.55	0.76
MAS-B	5000 (3.0)	276	01:48:25	2.29	9.7	2.35	52.8 ± 2.5	30.0 ± 0.7	10.00	0.76

Table 2
Evaluation metrics computed for all variants. Column **Dur.** represent the experiment duration. For the **Images** column, the first value and that inside parentheses denotes, respectively, the total and the suited number of frames captured.

energy) and the longest overall execution time (02:01:46). MAS-S and MAS-B presented similar results in terms of processed frames (3.0), execution duration (≈01h:48min), and FPS (0.76). Notably, the Effective FPS of the MAS variants fell below the configured value (1.0, as shown in Table 1), since frame capture only occurs during the animal passage time, with the system remaining idle during the inter-arrival intervals.

The batch variants achieved higher inference throughput (ITR) than their single-image counterparts. For instance, MAS-B attained a 9× higher inference rate than MAS-S. However, this advantage did not improve overall system performance, as both variants achieved identical FPS and STH values. Instead, batching increased resource consumption, especially RAM usage, and to a lesser extent CPU utilization. Notably, inference accounted for 87.5% of the total execution time in SEQ-S, indicating that Weight Prediction module was actively running for nearly the entire execution duration.

The sequential variants exhibited higher CPU consumption, with SEQ-B, showing the highest peak (CPU_{95th} = 53.5), but lower memory usage compared to the MAS variants. Regarding energy consumption, a general tendency between execution duration and energy usage can be observed, with SEQ-B showing both the longest duration and the highest absolute energy consumption (10.87 Wh). However, this relationship is not linear across all scenarios, as SEQ-S, despite being the second longest in duration, presented the lowest energy consumption (10.12 Wh).

Figs. 4, 5, and 6 illustrate the computational efficiency (CPU and RAM utilization) and energy consumption (W) of each variant over time. For improved visualization and noise reduction, the metrics were grouped into 30-sample windows, and each point represents the **max** value within the corresponding interval. Analysis of Fig. 4 shows that CPU utilization throughout the experiment followed the same ordering presented in Table 1. The sequential variants (SEQ-S and MAS-S) exhibited lower CPU variability, whereas SEQ-B showed both the highest mean CPU utilization and the largest peak-to-peak amplitude. These peaks corresponded to animals with four captured images, while the remaining animals had only three, suggesting a strong relationship between the number of suitable images and CPU utilization. A recurrent cyclic pattern can also be observed in MAS-B, where CPU utilization gradually increases until reaching a peak, followed by a sharp reduction and subsequent repetition of the same cycle, suggesting periodic fluctuations associated with the batch-based processing workflow.

Regarding RAM consumption (Fig. 5), all variants exhibited low variability throughout the experiment, with MAS-B showing the highest memory consumption. Concerning energy consumption (Fig. 6), all variants exhibited similar power consumption ranges (6.5–9.0 W), except for SEQ-S, which consumed less power on average (6.8 W) and showed lower variability. In MAS-B, a recurring cyclic pattern was observed, similar to that shown in Fig. 4 for CPU utilization, with power consumption gradually rising to a peak before dropping sharply and repeating.

4. Discussion

This study aims to assess the deployment feasibility, computational efficiency, and energy consumption of CVb-SWP, an edge IoT CV-based application for automated sheep weight estimation in smart farming environments. Four variants of the system were developed combining single-stream and batch-stream deployment strategies and sequential and MAS module processing and communication paradigms. The variants performance were evaluated through metrics System throughput, Response latency, Inference time ratio, Inference throughput, Top 95th mean CPU utilization, Top 95th mean RAM utilization, Power consumption total, and Effective FPS.

Based on the evaluated variants, the proposed system proved feasible on the target device, as all executions completed successfully with RAM utilization remained below 30% and energy consumption limited to approximately 10 Wh over nearly 2 hours of operation. Nonetheless, CPU utilization warrants attention, particularly in SEQ-B, where it peaked at nearly 70%. Notably, since the Raspberry Pi 5 lacks a dedicated inference accelerator, all model processing is performed on the CPU, which naturally increases its utilization. On devices equipped with dedicated Edge AI hardware, CPU usage is expected to be significantly lower, further supporting performance gains on more capable devices.

The sequential variants reached their processing limit under the evaluated conditions, as their synchronous architecture forces each stage to wait for the previous one, thereby limiting the number of frames captured while animal move through the corridor. In contrast, the MAS variants, evaluated at 1.0 FPS (Table 1), retained spare computational

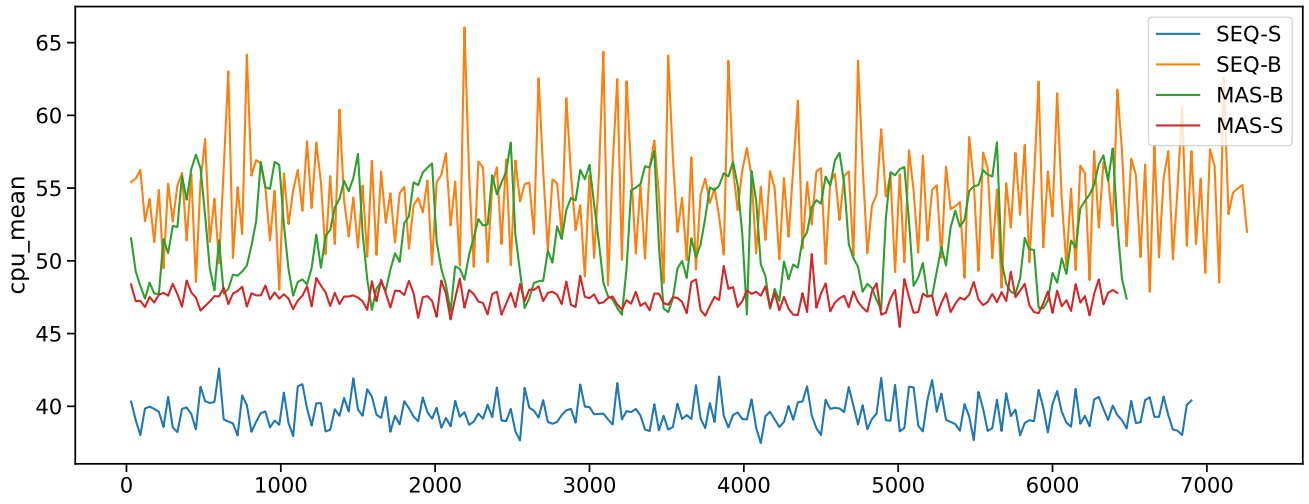


Figure 4: CPU utilization over time for all system variants (window size = 30).

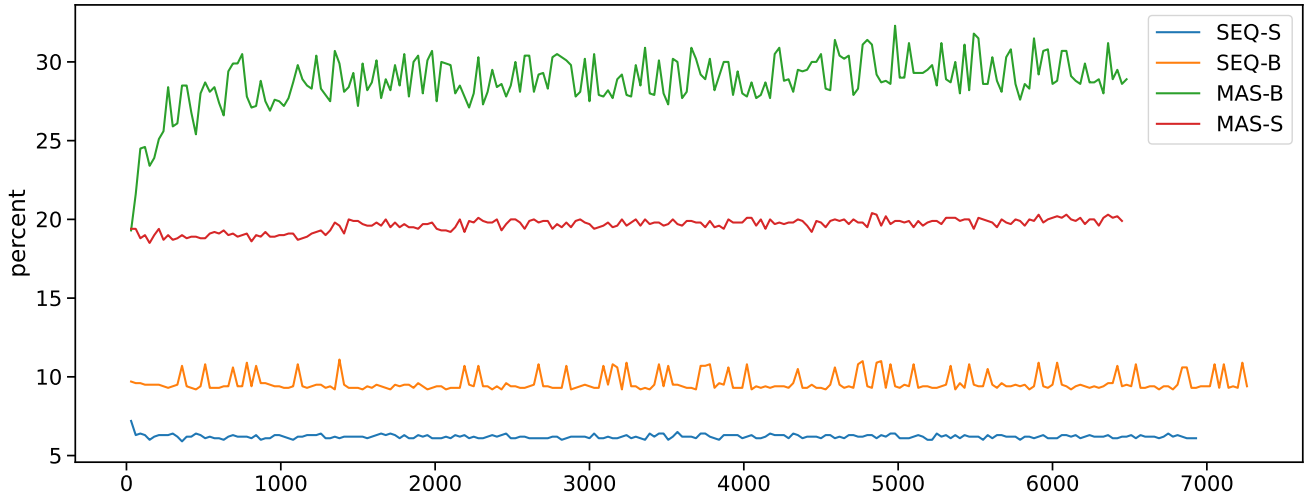


Figure 5: RAM utilization over time for all system variants (window size = 30).

capacity, suggesting that higher capture rates could be supported at the expense of greater resource consumption and latency. For the weighing method adopted in this study, a higher FPS is desirable because images are captured while animals are in motion. Averaging multiple predictions effectively mitigates variability caused by differences in animal posture [2]. Thus, MAS variants are inherently more flexible than sequential ones and are therefore expected to perform better in real-world scenarios, where animal exposure time to the camera may vary considerably. Under such conditions, sequential architectures may fail to capture sufficient suitable frames when animals move rapidly through the corridor.

In the sequential variants, the interval between consecutive animals (Arrival Time in Table 1) starts only after inference on the previous animal is completed, effectively extending the configured arrival time by the inference duration. In contrast, the MAS variants perform final weight estimation during the arrival interval itself, introducing no additional delay between animals. As a result, the MAS variants achieved a throughput of 276 animals per hour, compared to 246 for SEQ-B and 258 for SEQ-S, representing a gain of 30 and 18 animals per hour, respectively. Other aspect that stands out is the significantly higher inference time ratio observed in MAS-S (87.5%) compared to MAS-B (9.7%) and the sequential variants. This behavior can be attributed to the concurrent inference characteristic inherent to the MAS paradigm: since agents operate autonomously, the Weight Prediction Agent begins inference on each received frame immediately, regardless of whether previous inferences are still running. This results in multiple simultaneous inference calls competing for CPU resources,

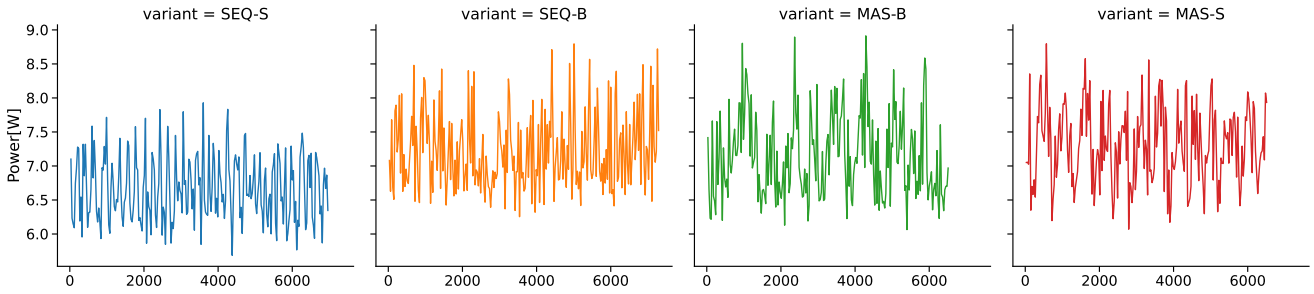


Figure 6: Power utilization over time for all system variants (window size = 30).

which progressively degrades individual inference times and consequently increases both the inference throughput and response latency.

Nonetheless, considering the observed relationship between the number of suitable images and CPU and RAM consumption (Table 2), the MAS-S variant emerges as the lower-risk alternative. Because inference is performed on individual images, resource utilization remains relatively stable even when the number of processed images increases, reducing the likelihood of resource spikes that could lead to system instability or crashes. Furthermore, its throughput and FPS were not significantly different from those of MAS-B. This does not preclude the use of batch-stream deployment. However, additional safeguards should be implemented to prevent system failures, such as limiting the maximum number of images submitted for inference.

Finally, system throughput reached 4.6 animals per minute, approximately eleven times the throughput reported by Sant'Ana et al. [18] for manual weighing in small-scale production (0.4 animals per minute), while reducing environmental disturbance and animal stress. In addition to improving operational efficiency, the proposed system reduces labor requirements, an important advantage in rural livestock systems that increasingly face labor and skills shortages [19]. Furthermore, the low energy consumption suggests that the system can be deployed in remote settings using off-grid power sources.

4.1. Limitations

The experiments were conducted without a physical camera connected to the device, loading frames from local storage instead, and no animal recognition routine (e.g., based on RFID or biometric identification) was implemented in the Animal Arrival. As a result, the energy consumption associated with real-time frame acquisition and animal recognition were not measured. Since small variations in execution time and number of frames processed were observed across strategies, slightly different impacts on total power consumption would be expected in a real deployment scenario.

4.2. Future Research

Building on the discussions and limitations presented, several directions for future research can be identified. First, evaluating the proposed architecture on devices equipped with dedicated Edge AI accelerators, such as the NVIDIA Jetson series, would provide a more comprehensive understanding of the performance and energy efficiency gains achievable beyond the worst-case scenario represented by the Raspberry Pi 5. Second, investigating the impact of higher frame capture rates on both resource consumption and model inference accuracy would allow for a more informed analysis of the trade-off between computational cost and prediction quality in the agent-based architecture. Third, conducting experiments with a physical camera connected to the device and capturing frames in real time would enable a more complete assessment of the total energy consumption of the solution, including the cost associated with frame acquisition. Finally, integrating an animal identification and recognition routine into the pipeline represents a relevant next step toward a fully operational system, and its impact on computational and energy costs should be evaluated.

5. Conclusions

Edge Computing has the potential to become an effective solution for overcoming the challenges associated with deploying AIoT ConvNet-based weighing systems on farms. However, edge devices, although increasingly powerful, still have limitations related to processing, storage, power availability, and network connectivity that may be incompatible with the potentially high computational costs of most recent weight monitoring ConvNet-based methods. This study assess the deployment feasibility, computational efficiency, and energy consumption of four variants of the CVbSWP, an edge AIoT ConvNet-based system for automated sheep weight monitoring, in PLF environments. These variants differ in deployment strategy (single-stream and batch-stream) and in module processing and communication paradigm (sequential and MAS). The results achieved are encouraging and show that the system operates feasible on a Raspberry Pi 5, a representative worst-case deployment scenario, achieving a system throughput of 276 animals weighed per hour in MAS variants. Nonetheless, the asynchronous architecture of MAS has room to scale, allowing higher capture rates with potential benefits to model inference accuracy, at the cost of acceptable, but increased resource consumption and latency.

Despite the results, the experiments were conducted without a physical camera connected to the device and no animal recognition routine, therefore the energy consumption associated with these tasks were not measured. In future works, it will be investigated the impact of higher frame capture rates on both resource consumption and model inference accuracy for MAS variants. Overall, this work advances our understanding on developing more computational and energy efficient AIoT methods for animal weighing in livestock farms and provides a basis for practical application and further research.

Acknowledgements

This research was funded by Brazilian National Council for Research and Development (CNPq) via Grants No. 307967/2022-0 (Development of Intelligent Architecture for Applications on Internet of Robotic Things) and No. 407274/2021-9 (”Strategic Solutions for Aerial-Terrestrial IoT Network Architectures”), and by Ministerio da Integração e do Desenvolvimento Regional (MIDR) - Coordenação-Geral de Sistemas Produtivos Inovadores via Grant Termo de Execução Descentralizada (TED) Nº 942052/2023 - 2023NS001587 (”Desenvolvimento e Difusão de Tecnologias para Melhoria da Produção de Carne nos Rebanhos Caprino e Ovino na Chapada Vale do Itaim”).

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used Claude Sonnet 4.6 (Anthropic) to improve the clarity and language of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] P. Niloofar, D. P. Francis, S. Lazarova-Molnar, A. Vulpe, M.-C. Vochin, G. Suciu, M. Balanescu, V. Anestis, T. Bartzanas, Data-driven decision support in livestock farming for improved animal health, welfare and greenhouse gas emissions: Overview and challenges, Computers and Electronics in Agriculture 190 (2021) 106406.

[2] E. C. Sousa, R. de Andrade Lira Rabelo, J. L. R. Sarmento, A computer vision end-to-end method for predicting sheep body weight from depth images, Small Ruminant Research 261 (2026) 107799.

[3] K. Li, G. Teng, Study on body size measurement method of goat and cattle under different background based on deep learning, Electronics 11 (2022) 993.

[4] Z. Hou, L. Huang, Q. Zhang, Y. Miao, Body weight estimation of beef cattle with 3d deep learning model: Pointnet++, Computers and Electronics in Agriculture 213 (2023) 108184.

[5] H. Afridi, M. Ullah, Ø. Nordbø, S. C. Hoff, S. Furre, A. G. Larsgard, F. A. Cheikh, Analyzing data modalities for cattle weight estimation using deep learning models, Journal of Imaging 10 (2024) 72.

[6] M. Liao, G. Morota, Y. Bi, R. R. Cockrum, Predicting dairy calf body weight from depth images using deep learning (yolov8) and threshold segmentation with cross-validation and longitudinal analysis, Animals 15 (2025) 868.

[7] H. Zhang, Y. Zhang, K. Niu, Z. He, Neural network-based method for contactless estimation of carcass weight from live beef images, Computers and Electronics in Agriculture 229 (2025) 109830.

[8] C. Meckbach, V. Tiesmeyer, I. Traulsen, A promising approach towards precise animal weight monitoring using convolutional neural networks, Computers and Electronics in Agriculture 183 (2021) 106056.

[9] J. V. M. Marques, W. S. Santos, R. L. Araújo, J. L. R. Sarmento, R. R. V. Silva, Non-invasive and low-cost approach for estimating sheep weight using computer vision, Engineering Reports 7 (2025) e70329.

[10] J. Xu, B. Gu, G. Tian, Review of agricultural iot technology, Artificial Intelligence in Agriculture 6 (2022) 10–22.

[11] J. Pan, J. McElhannon, Future edge cloud and edge computing for internet of things applications, IEEE Internet of Things Journal 5 (2018) 439–449.

[12] D. Muhammed, E. Ahvar, S. Ahvar, M. Trocan, M.-J. Montpetit, R. Ehsani, Artificial intelligence of things (aiot) for smart agriculture: A review of architectures, technologies and solutions, Journal of Network and Computer Applications 228 (2024) 103905.

[13] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, C.-J. Wu, B. Anderson, M. Breughe, M. Charlebois, W. Chou, R. Chukka, C. Coleman, S. Davis, P. Deng, G. Diamos, J. Duke, D. Fick, J. S. Gardner, I. Hubara, S. Idgunji, T. B. Jablin, J. Jiao, T. S. John, P. Kanwar, D. Lee, J. Liao, A. Lokhmotov, F. Massa, P. Meng, P. Micikevicius, C. Osborne, G. Pekhimenko, A. T. R. Rajan, D. Sequeira, A. Sirasao, F. Sun, H. Tang, M. Thomson, F. Wei, E. Wu, L. Xu, K. Yamada, B. Yu, G. Yuan, A. Zhong, P. Zhang, Y. Zhou, Mlperf inference benchmark, 2020. URL: https://arxiv.org/abs/1911.02549. arXiv:1911.02549.

[14] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Global Edition, Pearson Education, 2021.

[15] L. S. Melo, R. F. Sampaio, R. P. S. Leão, G. C. Barroso, J. R. Bezerra, Python-based multi-agent platform for application on power grids, International transactions on electrical energy systems 29 (2019) e12012.

[16] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994.

[17] TheHWcave, TC66C Python Interface, https://github.com/TheHWcave/TC66/blob/main/TC66C.py, 2026. Accessed: 2026-04-30.

[18] D. A. Sant'Ana, M. C. B. Pache, J. Martins, W. P. Soares, S. L. N. de Melo, V. Garcia, V. A. de Moares Weber, N. da Silva Heimbach, R. G. Mateus, H. Pistori, Weighing live sheep using computer vision techniques and regression machine learning, Machine Learning with Applications 5 (2021) 100076.

[19] M. Ryan, Labour and Skills Shortages in the Agro-food Sector, Technical Report 189, OECD Publishing, Paris, France, 2023. doi:https://doi.org/10.1787/ed758aab-en.

6 CONCLUDING REMARKS

This thesis evaluated the accuracy, generalization capability, and computational efficiency of an end-to-end Computer Vision method for sheep body weight estimation based on the EfficientNet-B3 ConvNet architecture and depth images captured from a dorsal (*top-down*) perspective.

Overall, based on the experiments conducted and the analysis of the results discussed above, it can be concluded that all the research hypotheses established in this thesis were validated: (i) the proposed method was able to estimate sheep body weight with high accuracy, without requiring explicit segmentation or manual feature extraction steps (H1); (ii) top-view depth images contributed to the model's ability to generalize across animals with different coat colors and from different farms (H2); and (iii) the method operated efficiently on resource-constrained devices, demonstrating compatibility with practical precision livestock farming applications (H3).

Nevertheless, further experiments using a larger and more diverse image dataset are required, encompassing a broader range of production environments, breeds, and phenotypes. Further investigation is also needed into strategies capable of preventing or mitigating issues related to the quality of images acquired in outdoor environments. In addition, the results obtained from the Generalized Linear Mixed Models analysis provided important evidence regarding the factors influencing the performance and behavior of the proposed method. These findings may support further improvements to the model in future research, including, for example, the incorporation of attention mechanisms to enhance the identification and extraction of relevant features from images of younger and, consequently, smaller animals.

Finally, the results achieved represent a step toward bridging the gap between scientific research and the practical reality of sheep farming. By contributing to the development of more accurate, accessible, and practical non-invasive weighing technologies for use under real production conditions, this thesis seeks to foster the adoption of digital tools capable of improving flock productivity, promoting animal welfare, supporting the sustainability of livestock production systems, and delivering tangible benefits to producers, particularly small- and medium-scale farmers.