

# Monitoring of Animal Movement using Computer Vision

Stephen Nii Adu Tagoe  
Engineering Department  
Ashesi University  
Accra, Ghana

stephen.tagoe@ashesi.edu.gh

Mohammed Elmir  
Engineering Department  
Ashesi University  
Accra, Ghana

mohammed.elmir@ashesi.edu.gh

Nathan Amanquah  
Engineering Department  
Ashesi University  
Accra, Ghana

namanquah@ashesi.edu.gh

**Abstract**— This paper presents a novel approach for identifying individual pigs and monitoring the movement of individual pigs using computer vision, designed explicitly for non-intrusive and real-time tracking in farm environments. Identifying individual pigs is the first step taken to track their movement. Accurately identifying individual pigs enables real-time monitoring of the pigs in order to get essential parameters such as their movement. Monitoring the movement of individual pigs is very important as it allows farmers to track their health. Pig movements such as walking, rolling, and laying down give the farmer a clear indication of how healthy each pig is. In this study, the focus is on tracking the movement of individual pigs when they move in the pen. Leveraging a Raspberry Pi 4 equipped with a Raspberry Pi Camera Module 3, the system integrates multiple deep learning models, including YOLOv8n, MobileNetSSD, and EfficientNet-B0, to achieve high-accuracy breed identification and motion detection. Among these, YOLOv8n emerged as the most effective, reaching 97-98% accuracy. The project involves a centroid-based tracking algorithm that measures movement by calculating Euclidean distances between centroids across frames. The data gathered on the individual pig movements in meters is analysed and visualised through time series graphs, providing valuable insights into the motion of individual pigs. This paper discusses the system's design, implementation, and performance, highlighting its potential applications in precision livestock farming.

**Keywords**—YOLOv8n, centroid, Computer vision, Machine learning, Euclidian distance

## I. INTRODUCTION

In recent years, Africa has seen a significant increase in pork demand and consumption, driven by population growth, urbanisation, and evolving dietary preferences [1]. A study in [2] indicates that the pork market size in sub-Saharan African countries like Nigeria, South Africa, and Ghana is projected to grow by 161% by 2050. However, despite this anticipated rise in pork production, the methods used to monitor pig behaviour on African farms remain primarily traditional, relying heavily on manual labour and basic infrastructure [3]. Conventional practices, such as visual observation to detect unusual behaviours like illness, distress, and irregular feeding habits, are labour-intensive, time-consuming, and prone to human error and subjectivity [3].

Pig movement is a critical indicator of health, with activities such as walking, standing, humping, and lying being closely linked to their mortality rate. Detecting changes in pig movement can allow for early disease detection [4]. The recent advancements in machine learning, sensor technology, and the Internet of Things (IoT) offer promising solutions to the challenges posed by traditional monitoring techniques [3]. Specifically, the rapid development of deep learning-based computer vision has become increasingly popular in monitoring applications. Two primary approaches in computer vision-based monitoring include one-stage and two-stage detection. One-stage detectors like YOLOv8,

MobileNetSSD, and EfficientNet-B0 are ideal for real-time speed and accuracy.

Alternatively, a study in [5] used motion sensors such as inertial sensors, accelerometers, and gyrometers have been employed to monitor animal movement. However, these methods require attaching multiple sensors to pigs, which can be problematic as certain behaviours like rolling and aggression interfere with the sensors' functionality.

Within the domain of pig monitoring, deep neural network architectures such as convolutional neural networks (CNN) have been employed to meticulously analyse data from sensors, images, and video frames to detect vital pig behaviours such as aggression [12] and feeding habits [1]. Other applications, such as detecting pig postures (standing, mounting, ear biting and lying), have been monitored through computer vision algorithms like Visual Geometry Group (VGG) and Faster-Region-based Convolutional Neural Network (Faster RCNN) that are built using deep neural network architectures [12]. The use of deep neural networks has provided a giant leap that can be capitalised on to solve the challenges posed by traditional monitoring of pig behaviour on farms, thus improving farming methods while cutting labour costs and saving time.

## II. LITERATURE REVIEW

Research has been done to identify various ways to monitor animal locomotion in livestock farming. The manual method is commonly used to identify animal motion [8]. This is done by recording the number of zones that a pig enters in an area. However, these approaches are usually based on approximation, not actual values.

To overcome the challenges of the manual method, sensors were used to track animal movement. However, many of these approaches require sensors to be placed on the animals. Using sensors and tags can have a lot of disadvantages. Some of these include biosecurity risks [6] and the intense pain the animals go through during the process of installing the sensors [7].

Image processing techniques have been used to track animal movements in livestock farming to overcome these drawbacks. These techniques tend to be non-intrusive, reducing the risks of using sensors. A study in [8] introduced a system to track pig movement automatically by administering various doses to seven pigs and measuring the movement over 60-minute intervals. In [9], a real-time automatic cow locomotion monitoring system was designed to track the movement and posture of eight pregnant cows in the 24 hours leading up to calving. The system could accurately classify, on average, 85% as standing and 87% as drinking and eating sessions.

A study in [10] utilized the mean-shift clustering algorithm and computer vision to track the movement of grouped pigs. The YOLOv5 model was used to identify the pigs with an accuracy of 99%. The authors, however, encountered problems when pigs came very close to each

other. Also, the study could only monitor the movement of grouped pigs and not individual pigs.

Another method of identifying animal movement is colour tracking [8]. Colour tracking usually involves placing a coloured mark, which is unique in terms of saturation or hue on the animal. The colour is constantly tracked using computer vision to determine how much the animal moves [13]. This solution overcomes the problem of varying backgrounds because of the distinction of colour in terms of saturation and hue. However, it is a costly method and is usually done commercially.

### III. PROPOSED SOLUTION

The proposed solution leverages a computer vision model to detect the movement of individual pigs in a pen. The system is built to be non-intrusive by ensuring no sensors are attached to the individual pigs.

#### A. Pig Selection and Data collection

Four pigs of different breeds with distinct characteristics in colour were selected to identify different pig breeds. These breeds were Berkshire (Black colour), Pietrain (Spots of black and white), Duroc (Brown colour), and Landrace (White colour). For consistency in the experiment, the breeds selected were of similar weight and age (25 kg and eight weeks (about two months), respectively).

Data was collected for two weeks using a V380 Pro Wi-Fi PTZ CCTV camera with a 1080p HD resolution and 12MP resolution. A second batch of data was collected using the Raspberry Pi Camera Module 3 to ensure our compatibility with the model since it would be used in the final design. Fig 1a, 1b and 1c are images captured from the V380 Pro camera, Raspberry Pi camera and its position, respectively.



Fig 1: Image captured from V380 Pro camera (A) Image captured from Raspberry Pi Camera (B) Position of Camera (C)

#### B. Data processing, model selection & training

Ten thousand six hundred seventy-six images were extracted from the data collected. The images were annotated based on the four class labels (Berkshire, Duroc, Landrace and Pietrain) using the RoboFlow online annotator. The images were annotated by drawing bounding boxes around the pigs observed in the images extracted, as seen in Fig 2.

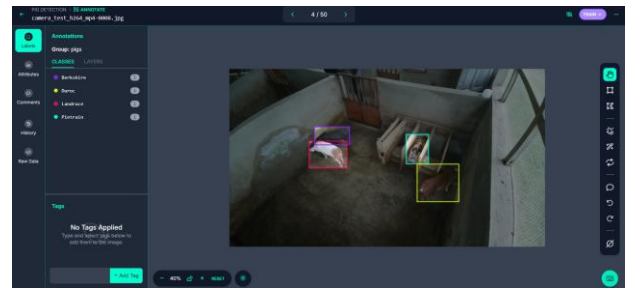


Fig 2: Image annotation using RoboFlow

Seven augmentation techniques were added to introduce specific features that mimic camera occlusions, different pig orientations, and brightness, which are likely to occur on the farm, as seen in Table 1. The images were first resized to 320x320 size then split into 82% training data (8,794 images), 12% testing data (1,254 images) and, 6% validation data (628 images). Table 2 indicates a summary of the preprocessing steps used.

Table 1: Augmentation techniques used with justification

Augmentation Technique	Reasoning
Flip, Rotation	Capture different orientations of the pig.
Crop	Improves object localisation by breaking down image
Grayscale, Saturation & Brightness	Suitable for situations where lightning conditions may be a problem
Blur, Noise	Occlusion

Table 2: Data processing stages

Data processing	Results
Tool used	RoboFlow Annotator
Images Extracted & Annotated	10,676
Image resizing	320x320
Augmentations added	7
Splitting Rule	82:12:6

YOLOV8n, MobileNetSSD and EfficientNet-B0 models were selected for the computer vision task. The reason was that they were pre-trained and well-known for their speed and accuracy in real-time operations. Also, all three models are one-stage detectors, which means the bounding boxes and class probabilities are predicted in a single network pass. This method ensures the models are lightweight, efficient and fast as they do not require separate region proposal stages like two-stage detectors such as R-CNN, Faster R-CNN, and SPPNet. The EfficientNet-B0 model is the base line model of the different EfficientNet variants with over five million parameters. The selection of the B0 variant was as a result of it being light weight and fast making it suitable for embedding on constrained devices. The feature extraction layers of the three pre-trained models were frozen before training on the

dataset was processed to maintain the performance of the models selected. All three pre-trained models were loaded and trained using the Google Collaboratory Notebooks. The YOLOv8n model was trained for six hours and 350 epochs. EfficientNet was trained for three hours and 150 epochs, while the MobileNetSSD model was trained for eight hours and 300 epochs. Table 3 shows the hyperparameters used in training the three models. The Adaptive Moment Estimation (Adam) optimizer was selected for adjusting the weights and biases during training to minimise the loss function.

Table 3: Hyperparameters used in training models

YOLOV8n	MobileNetSSD	EfficientNet
Epochs: 350	Epochs: 300	Epochs: 150
Optimizer: SDG	Optimizer: Adam	Optimizer: Adam
Learning rate: 0.01	Learning rate: 0.001	Learning rate: 0.01

### C. Model training results

The results for the three pre-trained models varied, which can be attributed to their training for different epochs. Also, the architecture of all three models differs, affecting the speed and ability to learn features from the dataset. After training, the YOLOV8n model performed well in detecting the different class breeds with an overall accuracy of 97%. This was followed closely by the MobilenetSSD model, which had an overall accuracy of 94.6%. The EfficientNet-B0 model had a lower accuracy when compared to MobileNetSSD and YOLOV8n, with an accuracy of 70.0%. Table 4 shows the results from the training of the three models in detecting the different pig breeds. From the results obtained after training, the YOLOV8n model was selected to monitor the movement of the individual pigs. The EfficientNet-B0 model had very low accuracy due to the number of training epochs assigned during training (150 epochs). Subsequently, the learning rate parameter was set to 0.01 which meant the model may not be able to learn features effectively while updates through backpropagation are going to be more drastic thus increasing model losses while reducing its accuracy.

Table 4: Table showing results of models in detecting different pig breeds

Pig Breed	YOLO V8n	MobileNetSSD	EfficientNet-B0
Berkshire	0.96	0.89	0.65
Duroc	0.98	0.98	0.71
Landrace	0.97	0.95	0.75
Pietrain	0.97	0.96	0.68
Overall	0.97	0.95	0.70

### D. Deployment of model on Raspberry Pi 4

The model was deployed on a Raspberry Pi model 4B. The Raspberry Pi model 4B is a microprocessor which features a Quad-Core ARM Cortex-A72 chip which runs at 1.5GHz and an 8GB ram. It was selected as our microprocessor for the machine learning models because of its availability and its processing speed. Other substitutes to the Raspberry Pi Model 4B include the NVIDIA Jetson Nano and Odroid-XU4, in which the availability of a dedicated GPU on these

boards make them far more capable for computer vision applications than the Raspberry Pi 4B. However, the Pi 4B was selected over the others because of its relatively cheaper cost.

### E. Implementation of monitoring pig movement and Open CV.

OpenCV is an open-source computer vision and machine learning software library. After developing the computer vision model, OpenCV was used to detect individual pigs. The environment used to implement OpenCV was an HP desktop PC with an 8th-generation Intel Core i7 processor. After testing the OpenCV on the desktop, it was later transferred to the Raspberry Pi 4 for deployment. The IDE used was PyCharm for the desktop PC and Thonny on the Raspberry Pi 4. The programming language used on both platforms was Python.

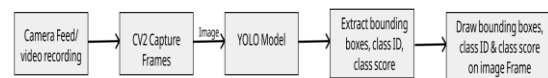


Fig 3: Steps used in the detection of individual pigs

Fig 3 describes the steps to detect individual pigs based on the breed. Images from the video frames are initially extracted using the cv2 library. After extraction, the images are passed to the YOLOv8n model for inferencing. The results are stored in a variable that contains an object with keys holding the detection information, such as the bounding box coordinates and class IDs with the corresponding scores. The results are looped through to extract the bounding box coordinates and the class data.

The bounding box coordinates are in the x1, x2, y1 and y2 variables. These variables represent the pixel position of the detected object in the image. The class data contains the class ID, which determines the pig breed and the class score, which is the degree expressed as a probability that the model is confident about its detection. The height and width of the bounding box are then calculated.

Fig 4 shows sample detections of individual pigs based on their breed using OpenCV and YOLOV8n model.

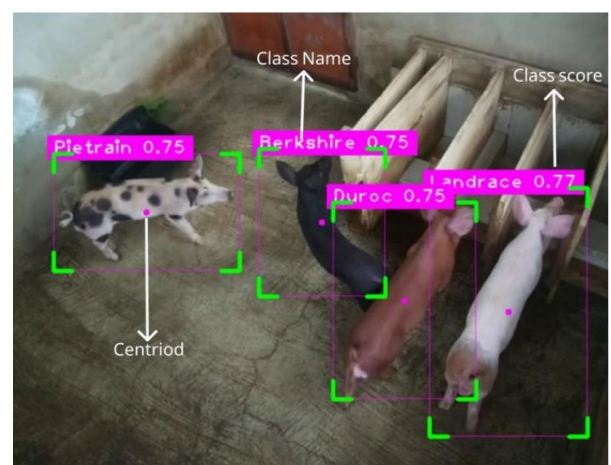


Fig 4: Sample detection with YOLOV8n and OpenCV

Centroid points must be drawn at the bounding boxes' midpoint. It is ideal to have the centroid point as it would play

a key role in tracking the motion of the individual pigs. The centroid point is obtained through the calculation:

$$C_x = x_1 + \frac{w}{2}, C_y = y_1 + \frac{h}{2}$$

#### IV. MONITORING INDIVIDUAL PIG MOVEMENT

In tracking the movement of individual pigs, the calculation of the Euclidean distance to set a threshold for centroid movement across different frames is used. The centroid as seen in fig 4 represents the center of the box bounding the object detected (pig). The centroid plays an integral role in knowing whether the pig has moved or not with changes in its pixel value ( $C_x$ ,  $C_y$ ) across frames indicates movement. By measuring the Euclidean distance across two frames, we can quantify the object's (pig) displacement. The threshold set through Euclidean distance calculations allows the system to accurately determine instances where there is active movement in body from one region in the pen to another (or when pig detected is stationary or moving). If the calculated Euclidean distance exceeds the threshold, the object is considered in motion otherwise stationary.

The movement of the individual pigs is monitored by tracking the centroid position of the bounding boxes across the video frames. Fig 5 illustrates the tracking of the centroid position of a pig in two overlapping frames. The centroid positions across the two frames are stored and the euclidean distance between them are calculated to determine if there was motion. To illustrate this, consider the centroid position the previous frame ( $C_{X_1}, C_{Y_1}$ ) and tposition in the next frame ( $C_{X_2}, C_{Y_2}$ ). The Euclidean distance between the two positions is calculated by the equation:

$$D = \sqrt{(C_{X_1} - C_{X_2})^2 + (C_{Y_1} - C_{Y_2})^2}$$

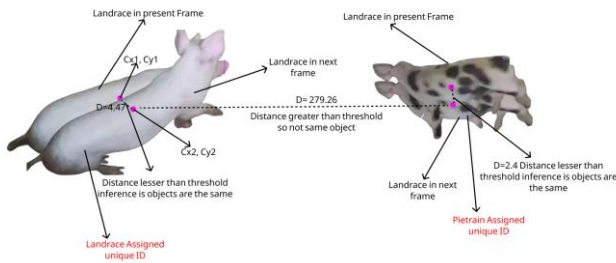


Fig 5: Illustration of tracking individual pig motion

A threshold distance is set as a baseline. This baseline is chosen by checking for the Euclidean distance at which pigs have very low movement activities (laying down, sleeping, standing) and when their activities are heightened (walking). A threshold value of 10 pixels was used in this study. When the distance calculated between two frames is greater than the baseline threshold, ten, the pig is not actively moving; however, when it exceeds, it is classified as moving. The number of times the distance exceeds the threshold is counted and displayed using OpenCV as seen in Fig 6. The motion of each pig is captured and posted to a Firebase database with its associated timestamps.

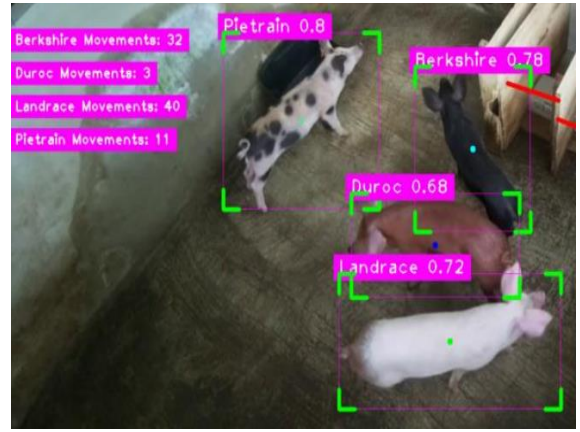


Fig 6: Tracking of individual pig motion

#### V. RESULTS

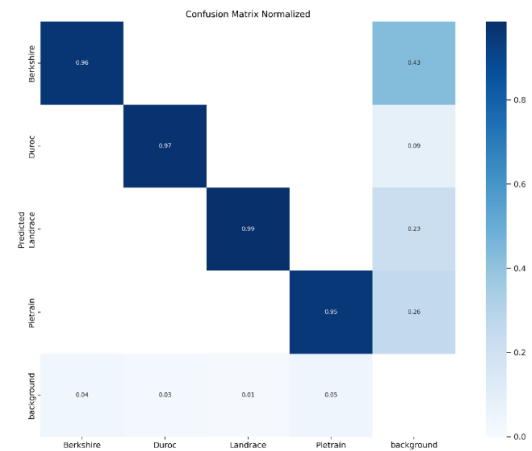


Fig 7: Confusion matrix for YOLO model

Fig 7 above is the confusion matrix for the YOLO model. The confusion matrix's vertical axis represents the data points' actual classes. It is also referred to as the true labels. The horizontal axis is the classes predicted by the model (also called the predicted labels). The diagonal values represent the correctly classified instance for each class. The off-diagonal values represent the misclassifications. The YOLO model performed well across all breeds with high accuracy. The information from the confusion matrix is given Table 5 below.

Table 5: Table of confusion matrix values for YOLO model

YOLO model	Diagonal values	Off diagonal values
Berkshire	0.96	0.43
Duroc	0.97	0.09
Landrace	0.99	0.23
Pietrain	0.95	0.26

One of the reasons for the possible misclassifications of the YOLOv8 model was the model performance under varying light conditions during the day. It was observed that during bright light conditions (sunlight), the model often misclassified the Landrace breed as a Duroc. This was

because of the change in skin color of the Landrace breed in high sunlight conditions. Since the model learnt the breeds using their colors, it mistook the Landrace as a Duroc due to the brown shades on the Landrace created by sunlight.

The system was tested on a pig farm for a day using a Pig. Data on its motion was taken throughout the day. The data collected was used in a time series graph, as seen in Fig 7. The graph shows the movement of the Pietrain at specific time stamps, which allows the pig to be tracked throughout the day. The vertical axis indicates the steps the pig took in meters, while the horizontal axis represents the time stamps recorded for each step. The time series graph reveals the activity pattern of the Pietrain over time. It shows the periods during the day when the pig is more active or dormant. Instances of peaks on the graph show periods where the Pietrain was actively moving while the troughs represent periods of inactivity. Inferences from the time series graph could be used to understand other key behavioral patterns behaviors of the pig such as the frequency of visitations to the feeding area, instances of gestation period, stress levels and health conditions. The time series graph indicated heightened level of movements between the times 16:00pm and 18:00pm which were the dedicated feeding period of the pigs in the farm during the evening, and very little movements during early mornings and midnight periods which we the resting periods.

The YOLOV8n model trained performed very well in detecting individual pigs based on their breed; this was because of their distinctive skin colours, the Berkshire having a dark colour, Duroc with brown skin, Pietrain white and black patches and the Landrace with white skin and few brown patches.

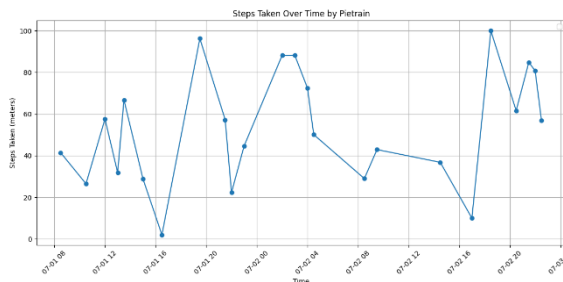


Fig 7: Time series graph showing the pig movement throughout the day

The recorded data can be used in other calculations to determine other animal behaviour closely related to movement behaviour, such as when on heat, sick, or pregnant.

## VI. LIMITATIONS AND CONCLUSIONS

In this study, four different pig breeds were selected. MobileNetSSD, EfficientNet-B0, and YOLOv8n were chosen as the machine learning models for the project. Among the three models, YOLOv8n performed the best, with an overall accuracy of 97%. Although, EfficientNet-B0 is known for its high efficiency in terms of speed and accuracy as compared to the many Convolutional Neural Network (CNN) models, its performance in this use case was low as compared to MobileNetSSD and YOLOV8n. The very low performance can be attributed to the hyperparameters assigned during training as seen in Table 1. OpenCV was used to draw the bounding boxes around the pigs detected. The centroids obtained from the bounding boxes were used to track the movement of the individual pigs by determining their distance

across video frames. The motions of each pig were captured over time, posted to a database, and analyzed using a time series graph.

Tracking of individual pigs using machine learning is a very innovative approach to tracking animal welfare in large scale farming. Integrating this method of monitoring pigs to conventional farm management systems can lead to improved health monitoring as the system can alert farmers when a pig's movement deviates from the norm, which may be an indication of sickness. It can also lead to feed optimization since the system can track the individual pigs feeding area movement habits. Combining this information which current farming practices can lead to a reduction in feed waste.

Some of the project's limitations include the Raspberry Pi's slow speed during real-time operation. This was tackled by taking shorter, regular-timed pictures instead of continuously taking videos. The YOLO model trained can detect pigs of different breeds and faces challenges when there are pigs of the same breed. A tracking algorithm or RFID sensors would be ideal for tracking pigs of the same breed.

For future works, integrating the machine learning approach with sensors such as accelerometers and RFID tags will increase the robustness and accuracy of the system in determining individual pig's motion. Additionally, the overall system should be optimized for tracking larger pig populations in bigger farms. In addition to this, the EfficientNet-B0 model's performance can be drastically improved by tuning its hyperparameters such as training epochs and learning rate.

## REFERENCES

- [1] W. Yu and J. D. Jensen, "Sustainability implications of rising global pork demand," *Animal Frontiers*, vol. 12, no. 6, pp. 56–60, Dec. 2022, doi: <https://doi.org/10.1093/af/vfac070>.
- [2] M.M. Erdaw, "Contribution, prospects and trends of livestock production in sub-Saharan Africa: a review," *International Journal of Agricultural Sustainability*, vol. 21, no. 1, Aug. 2023, doi: <https://doi.org/10.1080/14735903.2023.2247776>.
- [3] S. Wang, H. Jiang, Y. Qiao, S. Jiang, H. Lin, and Q. Sun, "The Research Progress of VisionBased Artificial Intelligence in Smart Pig Farming," *Sensors*, vol. 22, no. 17, p. 6541, Aug. 2022, doi: <https://doi.org/10.3390/s22176541>.
- [4] J. Xu, J. Ye, S. Zhou, and A. Xu, "Automatic quantification and assessment of grouped pig movement using the XGBoost and YOLOv5s models," *Biosystems Engineering*, vol. 230, no. 1537–5110, pp. 145–158, Jun. 2023, doi: <https://doi.org/10.1016/j.biosystemseng.2023.04.010>.
- [5] F. Marin, "Human and Animal Motion Tracking Using Inertial Sensors," *Sensors*, vol. 20, no. 21, p. 6074, Oct. 2020, doi: <https://doi.org/10.3390/s20216074>.
- [6] M. Hernández-Jover, N. Schembri, J. -a. L. M. L. Toribio, and P. K. Holyoake, "Biosecurity risks associated with current identification practices of producers trading live pigs at livestock sales," *Animal*, vol. 2, no. 11, pp. 1692–1699, Jan. 2008, doi: <https://doi.org/10.1017/s1751731108003066>.
- [7] E. Leslie, M. Hernández-Jover, R. Newman, and P. Holyoake, "Assessment of acute pain experienced by piglets from ear tagging, ear notching and intraperitoneal injectable transponders," *Applied Animal Behaviour Science*, vol. 127, no. 3–4, pp. 86–95, Nov. 2010, doi: <https://doi.org/10.1016/j.applanim.2010.09.006>.
- [8] N. M. Lind, M. Vinther, R. P. Hemmingsen, and A. K. Hansen, "Validation of a digital video tracking system for recording pig locomotor behaviour," *Journal of Neuroscience Methods*, vol. 143, no. 2, pp. 123–132, Apr. 2005, doi: <https://doi.org/10.1016/j.jneumeth.2004.09.019>.
- [9] Ö. Cingar et al., "Automatic real-time monitoring of locomotion and posture behaviour of pregnant cows prior to calving using online image analysis," *Computers and Electronics in Agriculture*, vol. 64, no. 1, pp. 53–60, Nov. 2008, doi: <https://doi.org/10.1016/j.compag.2008.05.014>.
- [10] J. Xu, J. Ye, S. Zhou, and A. Xu, "Automatic quantification and assessment of grouped pig movement using the XGBoost and

- YOLOv5s models,” *Biosystems Engineering*, vol. 230, no. 1537–5110, pp. 145–158, Jun. 2023, doi: <https://doi.org/10.1016/j.biosystemseng.2023.04.010>
- [11] C. Mishra and D. Gupta, “Deep Machine Learning and Neural Networks: An Overview,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 6, p. 66, Jun. 2017, doi: 10.11591/ijai.v6.i2.pp66-73.
- [12] J. Wei, X. Tang, J. Liu, and Z. Zhang, “Detection of pig movement and aggression using deep learning approaches,” *Animals*, vol. 13, no. 19, p. 3074, Sep. 2023, doi: 10.3390/ani13193074.
- [13] L. P. J. J. Noldus, A. J. Spink, and R. a. J. Tegelenbosch, “EthoVision: A versatile video tracking system for automation of behavioral experiments,” *Behavior Research Methods Instruments & Computers*, vol. 33, no. 3, pp. 398–414, Aug. 2001, doi: 10.3758/bf03195394.