Faculty of Science, University of Porto
Faculty of Engineering, University of Porto

Automatic Recognition of Pig Activity in Intensive Production Systems

INESC TEC

Diogo Mendes

Bachelor of Artificial Intelligence and Data Science

Orientadores na empresa: Ricardo Cruz

Co-orientador: Nuno Lavado (ISEC)

July 2, 2024

Contents

Abstract

This report presents the development of an automatic pig activity recognition system utilizing advanced machine learning and computer vision techniques. The primary objective is to create a system capable of identifying and classifying a variety of behaviors in real-time, such as lying, eating, and aggressive behaviors, using a microcontroller and video data from a camera. Initially, the approach involved employing both convolutional neural networks (CNN) and long short-term memory (LSTM) networks, along with tracking algorithms to support individual activity prediction. However, the methodology was refined to focus exclusively on a CNN-based model due to its superior performance in image analysis. The CNN model predicts the location of pigs within video frames and detects their activities, providing a robust solution for real-time monitoring. This system aims to significantly enhance the efficiency of animal monitoring in intensive production systems. By leveraging accessible equipment and advanced image processing techniques, we anticipate this project will substantially contribute to the fields of animal welfare and management. Additionally, the implementation of this system can lead to better resource allocation, early detection of health issues, and overall improved productivity in pig farming operations. The results from our experiments demonstrate the system's accuracy and reliability. The system achieved an accuracy rate of 91% in detecting two behaviors, aggressive and non-aggressive, and around 18% when the intersection of the bounding box with the pig is above 50%, providing a satisfactory prediction of ten objects when tested in the same environment it was trained in. This showcases its potential for widespread adoption in the agricultural industry. However, there are some limitations when the model is tested in different environments, and the predictions are sometimes not very accurate.

Keywords: automatic recognition, CNN, LSTM, pig behaviour, activity detection

Figure 1: Pen housing 10 male pigs.

1 Introduction

In the field of swine production, recognizing individual pig behaviors such as eating, drinking, resting, social interactions, aggression, and exploratory activities is crucial for timely intervention when necessary. Video monitoring facilitates the surveillance of a large number of animals, even in low-light conditions, without disturbing them. Intelligent image analysis systems can contribute to the early detection of abnormal situations, thereby improving animal health and welfare while modernizing the industry.

As interest in activity monitoring grows alongside the evolution of systems designed to facilitate it, animal monitoring has gained significant attention for its potential to automate behavior recognition. This automation can substantially aid farmers and animal producers, reducing the burden of constant manual observation. While most studies on animal behavior recognition focus on mice and cattle, swine monitoring has been relatively neglected. This underexplored area has become increasingly important due to recent regulations prohibiting pig castration, which has led to more aggressive behaviors among swine. Additionally, real-time monitoring of swine behavior poses a significant challenge due to the substantial amount of data required to effectively detect and categorize various behaviors. Addressing these gaps is crucial for improving welfare and management practices in swine production.

Improving animal welfare is a primary concern. Aggressive behaviors resulting from uncastrated pigs can lead to injuries, stress, and reduced overall health. By developing advanced monitoring systems, we can detect early signs of aggression and intervene promptly to mitigate these issues. Enhancing the efficiency and effectiveness of swine farming operations is also vital for economic sustainability. Automated behavior recognition can help farmers optimize resource allocation, improve the accuracy of health monitoring, and reduce labor costs. Finally, complying with regulatory changes and public expectations regarding animal welfare is essential for the agricultural industry's reputation and long-term viability. Addressing these challenges through innovative monitoring solutions not only supports regulatory compliance but also aligns with the growing consumer demand for ethically produced animal products.

In this project, a video was provided by the Agriculture School of the Polytechnic Institute of Coimbra (ESAC) in collaboration with INESC TEC, featuring a pen house with ten pigs that exhibit some similarities, Fig. 1. The project aims to achieve individual pig activity recognition to continuously track the pigs and predict their behavior. Additionally, we aim to construct a model capable of monitoring the pigs in real-time and processing the data using a microcontroller due to its low cost.

The primary goal of this project is to advance the field of real-time animal activity recognition, specifically focusing on swine. Monitoring pig behavior in real-time is challenging due to the significant amount of time pigs spend lying down, necessitating substantial data to effectively detect various behaviors. To address this, our project aims to use a microcontroller and an RGB camera to monitor pig behavior directly

in the pen house. We aim to identify and categorize various behaviors, including individual behaviors like lying and eating, and group behaviors such as biting, as illustrated in Figure 2a. This system will enhance the efficiency of animal monitoring in intensive production systems by leveraging accessible and advanced image processing techniques. Through this approach, we hope to provide a practical and scalable solution for improving animal welfare and operational efficiency in swine farming.

The objectives of this project include:

- Developing a model for the real-time recognition of pig activities in intensive systems.
- Exploring existing works to inform the model development.
- Implementing a CNN-based approach to accurately detect and classify pig behaviors in real-time.

Of the various approaches discussed below, the code for our final approach is available on a GitHub repository

This paper is structured as follows: Section 2 provides an overview of the related work and the context of the application domain. Section 3 presents the methodology, detailing all the necessary steps to start developing each approach of the project and the experimental results of each one. Section 4 presents the experimental results and analysis of our best and final approach, highlighting the performance of the algorithm. Section 5 concludes the paper and presents future work.

2 State of the art

Monitoring animal behavior has become a critical area of research, particularly with advancements in automatic video analysis and activity recognition technologies. While significant progress has been made in human activity recognition, such as in construction [1] and sports [2; 3], animal activity recognition has also seen notable developments. Studies on mice [4] and cattle [5] have laid the groundwork, yet swine activity recognition remains relatively underexplored.

Various devices have been employed to monitor animal behavior, including ear tags [6], accelerometers [6; 7], motion collars [7], and cameras [8]. For a non-invasive and cost-effective approach, this study proposes using microcontrollers connected to cameras to capture relevant data without disturbing the animals.

The models and techniques used in activity recognition range from traditional signal processing methods [6] to sophisticated artificial intelligence approaches. Convolutional Neural Networks (CNNs) are widely used for their capability to extract features from images and videos. However, standard CNNs might not be sufficient for capturing three-dimensional features in videos or movements, making 3D CNNs a more effective alternative for processing such data. Long Short-Term Memory (LSTM) networks are also extensively used to extract temporal features from movements, complementing CNNs' visual analysis. Recently, transformers, originally developed for natural language processing, have been adapted for behavior recognition in videos. These models leverage attention mechanisms to identify specific movements, enhancing the accuracy of activity recognition.

In addition to these models, methods such as optical flow are employed to capture motion patterns in videos, further improving activity recognition accuracy. Combining different models has proven effective, such as integrating CNNs with LSTMs to capture both visual and temporal features, or combining 3D CNNs with optical flow to merge spatial and temporal information for comprehensive movement analysis.

This project utilized a variety of libraries, including:

- **PyTorch:**[1] A deep learning framework used for constructing the neural networks. It provides tools for automatic differentiation and GPU acceleration, making it suitable for training complex models.

[1] https://pytorch.org/

- **OpenCV (cv2):**[2] An open-source computer vision library that helps visualize the results of the models and read videos. It provides functionalities for image processing and manipulation.
- **Imageio:**[3] A library for reading and writing images in various formats. It is used to effectively extract frames from videos.
- **LabelMe:**[4] A graphical image annotation tool that assists in labeling images to feed the model. It is essential for creating annotated datasets required for supervised learning.
- **Ultralytics YOLO:**[5] A library for implementing pre-trained YOLO (You Only Look Once) models for object detection and tracking. YOLO models are used to detect all the animals in the video frames accurately.

Inputs for pig activity recognition systems primarily consist of video data [9], but may also include sensor data from devices like accelerometers [7]. These inputs are processed to identify behavior patterns such as feeding, resting, walking, and social interactions. The system's outputs are classifications of the behaviors or activities occurring in real time.

Automatic activity recognition is a research area in constant evolution, driven by the need to monitor and improve animal welfare in production environments. Over the past years, a variety of approaches have been explored to detect and classify behaviors in different areas such as construction, human, and animal behavior, as shown in Table 1.

Table 1: Behavior Analysis

Models	Authors	Input	Output
3D CNN + OF*	Alvaro et al., 2020 [8]	350 videos, 12 min each	Cattle activity classification
3D CNN + OF*	Kaifeng et al., 2020 [10]	Dataset with 1000 videos of about 6s each	Pig activity classification
Transformer + 3D CNN	Meng et al., 2023 [1]	1595 videos of 1 to 9 seconds	Predicting human behavior in construction
Various models	Guangle et al., 2019 [2]	Datasets HMDB51, UCF101, and Sports-1M	Detecting human activity in the temporal space
CNN + LSTM - review	Qiumei et al., 2020 [11]	Collection of videos	Pig activity classification
CNN + LSTM	Chen et al., 2020 [12]	2400 episodes with 2 seconds each	Detecting aggressive behavior in pigs
Transformer + OF*	Kaidong et al., 2022 [13]	Datasets Youtube-VOS and DAVIS	Video inpainting
OF* + CNN	Bo et al., 2020 [5]	1080 videos	Detecting symptoms of illness in cattle
OF* + CNN	Zhenyang et al., 2018 [14]	Datasets UCF101, HMDB51, and THUMOS13 localization	Predicting behavior in human activities

Activity recognition systems can be implemented on a range of devices, from high-performance computers to embedded devices or wearable sensors. The choice of device depends on the specific application needs, considering factors such as real-time processing requirements, cost, and energy efficiency. The ultimate goal is to develop a model that can be executed on a microcontroller, as microcontroller prices are decreasing over time, making them a more desirable option for cost-effective and scalable solutions.

The performance of activity recognition systems is typically evaluated using metrics like accuracy, precision, and recall. These metrics are calculated based on specific datasets, which can vary in size, quality, and recording conditions. Comparing different evaluation methods helps identify the most effective approaches. For our project, the feasibility of implementation on a microcontroller, which has processing limitations, is crucial. Therefore, in addition to traditional metrics, the complexity of each model must be considered. Choosing low-complexity models is essential to ensure adequate performance in a resource-constrained context. Table 2 illustrates the advantages and disadvantages of each model explored.

The metrics used in the project were from a library named TorchMetrics[6] from PyTorch, including:

- **AUC (Area Under the Curve):** Used to measure the model's ability to distinguish between classes.

[2] https://opencv.org/
[3] https://pypi.org/project/imageio/
[4] https://blog.roboflow.com/labelme/
[5] https://docs.ultralytics.com/
[6] https://lightning.ai/docs/torchmetrics/stable/

- **F1-Score:** Provides a balance between precision and recall, especially useful in multi-behavior detection where accuracy alone might be insufficient.
- **mAP (mean Average Precision):** Used for bounding box detection to evaluate the precision and recall of the detected objects.

One of the approaches used Explainable AI (XAI), specifically Grad-CAM [15] from the Captum library[7], to detect which features of the images were contributing for the model's decision-making the most.

The field of pig activity recognition is poised for significant advancements with ongoing research in AI algorithms and sensor technology. The integration of more sophisticated models and the development of non-invasive, cost-effective monitoring solutions will enhance the accuracy and reliability of these systems. This research aims to contribute to this evolving field by proposing and validating new methodologies for monitoring pig behavior.

Table 2: Models and their Characteristics

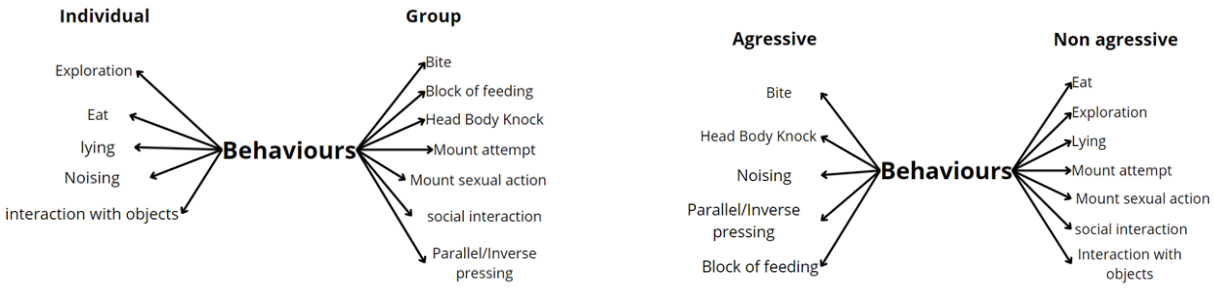
Models	**Advantages**	**Disadvantages**
CNN	Good with images	Requires a lot of data
RNN	Good with video sequences	Memory
LSTM	Good for retaining long-term dependencies	Memory
Transformer	Good with sequential data	Computational complexity
3D CNN	Good with pattern recognition	Computational complexity

3 Methodology

This section will explain the data acquisition process, including how the data was collected, as well as the various approaches and methodologies employed throughout the project. Each approach will be detailed, describing its development, constitution, and results. Multiple strategies were considered to ensure the final product met all client requirements. These objectives included individual monitoring of the animals, behavior prediction, and real-time monitoring. Additionally, an extra objective was to develop a model capable of being used in an implementation with a microcontroller.

Throughout the project, we explored several approaches to develop a robust system for pig activity recognition. The primary focus was to create a solution capable of identifying and classifying pig behaviors in real-time. The methodologies varied from leveraging entire video datasets to implementing specific models for behavior prediction and tracking.

- **Initial Approach:** This approach involved using the entire video dataset to predict all observed behaviors, which can be of two types: individual or group behaviors (Figure 2a). Additionally, the prediction of two specific behaviors, aggressive and non-aggressive (Figure 2b), was tested. The goal was to determine if comprehensive behavior prediction was feasible using the available data.
- **Tracking Model:** The second approach focused on identifying or developing a model or tool that could effectively track each pig. This tracking capability was crucial for predicting the behavior of individual pigs accurately over time.
- **Final Implementation:** The final approach aimed to predict bounding boxes around the pigs and classify behaviors into two main categories: aggressive and non-aggressive. These categories encompassed a series of specific behaviors, providing a practical solution for monitoring swine activity.



(a) individual and group behaviors

(b) aggressive and non-aggressive behaviours

Figure 2: Behavior classification groupings

The models were executed on two different platforms:

- **Desktop:** An Intel(R) Core(TM) i7-14700KF CPU @ 3.40 GHz with 32 GB of RAM, running Microsoft Windows 11 Home, equipped with an NVIDIA GeForce RTX 4070 Ti GPU with 16 GB of VRAM.
- **Server:** Provided by INESCTEC, featuring a NVIDIA A100 GPU with 40 GB of VRAM.

3.1 Data Acquisition

For this project, video data was utilized to study swine behavior. The data collection was provided by the Agriculture School of the Polytechnic Institute of Coimbra (ESAC). Over the course of one morning, approximately three hours of video footage were recorded within one of the pen houses containing 10 pigs.

The video footage was captured using a static camera strategically positioned to cover the entire area of the pen house, ensuring comprehensive visibility of the pigs. The recordings were made at approximately 30 frames per second (fps).

Accompanying the video data were detailed annotations of swine behaviors at each moment, meticulously recorded by an animal behavior specialist trained in video data analysis. These annotations were created using BORIS (Behavior Observation Research Interactive Software)[8], an open-source tool designed for the observation and analysis of behavior in scientific studies. BORIS enables researchers to mark specific behavioral events in videos, facilitating detailed and systematic analysis.

An essential aspect of using BORIS was the creation of an ethogram, a catalog of all observed behaviors during the study. The ethogram provided a standardized framework for labeling behaviors, ensuring consistency and accuracy in data analysis (as shown in Figure 3).

This combination of video data and behavioral annotations provided a robust dataset for training and validating our models. The detailed labeling by the specialist ensured that our dataset was rich in information, enabling a thorough analysis of swine behavior patterns.

3.2 First approach

As our first approach, we replicated the model used in the article "Recognition of Aggressive Episodes of Pigs Based on Convolutional Neural Network and Long Short-Term Memory" [12]. This study aimed to develop a deep learning method combining a convolutional neural network (CNN) and long short-term

[7] https://captum.ai/
[8] https://www.boris.unito.it/

Behavior code	Behavior type	Description	Key
LEVER	Point event	Levering at one part of the body as 1- another part considered as 2	L
NOSING	Point event	Anal/Genital nosing	N
FLEE	Point event	Animal actually fleeing away	F
BLOCK OF FEEDING	Point event	Blocks the access to feed the other animals using his body or shoving with body - NOT WITH HEAD	Q
CHASE	Point event	Chasing one another	C
EAT	State event	Eating from feeder	R
HEAD BODY KNOCK	Point event	Hitting with head - Count each head knock even if in same place	K
ABNORMAL BEHAVIOUR	Point event	Lick or bite - wall, chão, tail	W
LYING	State event	Lying down in any form lateral or sternal	Y
MOUNT ATTEMPT	Point event	Mounting attempt with no pelvic thrust or penile extrusion and in any position	A
MOUNT SEXUAL ACTION	Point event	Mounting with pelvic thrust and penile extrusion and in any position	M
BITE	Point event	Mouth open and really biting the other animal	B
THREAT	Point event	No actual contact - Attempt to bite or head knock or lunging	T
PRESSING PARALLEL/INVERSE	Point event	Pressing can be inverse or parallel on head or on body	P
EXPLORATION	Point event	Sniffing pen, wall, drinker, feeder, ground	E
SOCIAL INTERACTION	Point event	Sniffing, Gentlly touching and grooming other individual	S
INTERACTION with objects	Point event	Touching objects with snout	O
FLEE ATTEMPT	Point event	Tries to flee but still has the other animal on his back	G
INACTIVE	State event	When all 10 pigs are lying down at the same time - wait for last pig to lye down	I

Figure 3: Ethogram of pig behaviors from BORIS software.

memory (LSTM) to recognize aggressive episodes in pigs. The authors utilized video data of nursery pigs and manually selected 600 two-second aggressive episodes, which were then augmented to 2400 episodes through various transformations. These episodes, along with 2400 non-aggressive episodes, were used to train and validate the model, achieving an accuracy of 97.2% in recognizing aggressive behavior.

In our replication, we used all the videos and data provided in the original study. We divided the dataset into 80% for training and 20% for testing, with the test set consisting of the last videos. The following sections will elaborate on the preprocessing steps, the models used, and the results obtained from these models:

- **Preprocessing:** This subsection will detail the steps taken to prepare the video data for training and testing, including data augmentation techniques and the division of the dataset.
- **Models:** Here, we will discuss the architecture of the CNN and LSTM models used, including any modifications made to the original model and the rationale behind these changes.
- **Results:** This section will present the performance metrics of the models, comparing them with the results reported in the original study, and discussing any discrepancies or improvements observed.

By following the methodology outlined in the original article, we aim to validate their findings and potentially enhance the model's performance through further experimentation and refinement.

3.2.1 Preprocessing

In the initial stage of our approach, we performed preprocessing to extract all relevant information from the video data and the BORIS annotations. This step was crucial to ensure that we could utilize the data effectively for our analysis and model training.

Firstly, we quantified the number of behaviours present in the video data during the observation period. These behaviors were categorized into two types:

- **State Event:** These are behaviours that have a duration, starting at one point in time and ending at another. Examples include eating, lying, exploring, and social interactions.
- **Point Event:** These are behaviours that occur at a specific moment in time. Examples include biting, head and body knocking, mounting attempts, and threats.

To ensure the quality and usability of the data for training our machine learning models, we segmented the videos into short clips, each lasting a few seconds, corresponding to the observed behaviours. This method helped in isolating individual behaviours and ensured an even distribution of behavior types in the dataset. This approach is particularly important because some behaviors are more frequent than others, and an even distribution prevents the model from being biased towards more frequent ones.

The short video clips were extracted at varying frame rates of 3, 15 and 30 frames per second (fps). These frame rates were chosen not only for comparison with reference scientific papers but also because the animals were often inactive in the video data. Additionally, we included approximately 5 seconds of footage prior to the observed behavior to provide context and a sense of movement leading up to the behavior. This inclusion helps the model predict the beginning of a behavior more accurately.

By segmenting the videos and carefully selecting frame rates and durations, we ensured that the dataset was well-prepared for the subsequent stages of analysis and model training. This preprocessing step was essential for maximizing the effectiveness of our machine learning models in recognizing and predicting swine behaviors.

3.2.2 Models

In this approach, we implemented the same models as in the referenced study to facilitate a direct comparison when training and evaluating the performance of our machine learning architecture. The architecture comprised two types of neural networks:

- **Convolutional Neural Network (CNN):** CNNs are designed to automatically and adaptively learn spatial hierarchies of features from input images. They work by applying a series of convolutional layers, each of which uses filters to detect specific features such as edges, textures, and shapes. In our model, we used a VGG (Visual Geometry Group) network pre-trained on ImageNet to extract visual features from each frame of the video. The VGG network transforms the original images from the videos into feature vectors with enhanced discriminatory power, reducing the feature dimensions and optimizing the feature extraction process. This resulted in a dimensional vector of 25088 (7 x 7 x 512).
- **Long Short-Term Memory (LSTM):** LSTMs are a type of recurrent neural network (RNN) capable of learning long-term dependencies, making them well-suited for sequence prediction problems. They work by maintaining a cell state that can capture temporal information over long sequences of data. In our model, the LSTM was built with a single hidden layer fully connected, followed by a sigmoid activation function. This design aimed to predict behaviors in real-time by capturing the temporal dynamics of the video sequences.

We trained the models for all behaviors using a sigmoid activation function, which is appropriate for multi-label classification where each behavior is predicted independently. For the detection of two specific behaviors (aggressive and non-aggressive), we used a softmax activation function to classify each instance into one of these two categories. The pseudocode is shown in Algorithm 1.

Algorithm 1 Model Class

Input: x - Input tensor with shape (N, T, C, H, W)
Output: Predicted classes for each input sequence

Initialize VGG16 model pre-trained on ImageNet without the classifier part
Initialize LSTM with input size $7 \times 7 \times 512$, hidden size 128, and 2 layers
Initialize Dense layer with output size equal to number of classes
procedure FORWARD(x)
    Reshape x to $(N \times T, C, H, W)$
    $x \leftarrow \text{VGG}(x)$ ▷ Pass through VGG network
    Reshape x to $(N, T, 7 \times 7 \times 512)$
    $x \leftarrow \text{Flatten}(x, 2)$ ▷ Flatten feature maps
    $x, \_ \leftarrow \text{LSTM}(x)$ ▷ Pass through LSTM network
    $x \leftarrow \text{Dense}(x[:, -1, :])$ ▷ Dense layer on the last LSTM output
    if multi-label classification **then**
        $x \leftarrow \text{Sigmoid}(x)$
    else
        $x \leftarrow \text{Softmax}(x)$
    return x

After training the models, we implemented various explainable artificial intelligence (XAI) techniques to interpret the model's decisions. One of the key techniques used was Grad-CAM (Gradient-weighted Class Activation Mapping)[15] from the Captum library. Grad-CAM helps visualize which parts of the input image are most relevant for the classification decision made by the neural network. This technique was applied to the last convolutional layer of the CNN to highlight the areas of the image that were most influential in predicting specific behaviors.

Throughout the training process, we adjusted the model parameters several times to optimize performance. These adjustments, along with the corresponding results, will be detailed and analyzed in the results section of this approach.

3.2.3 Results

The analysis of the behavior recorded during the observation morning revealed significant insights into the activities carried out in the pen house. The data were synthesized into Table 3, which summarizes the total and median duration of each behavior in seconds, along with the number of occurrences for each type of behavior. This data were collected by analyzing all the videos, with durations counted in two different ways: state events were counted from start to finish, while point events were counted by multiplying the number of occurrences by 5 seconds. The 5-second multiplier was chosen to provide a consistent time frame for each point event, simulating a brief but relevant period for behavioral analysis.

Table 3: Behaviours analysis

Index	Behaviour	Duration Behaviour (s)	Average Duration (s)	Total Behaviours
0	EAT	5732.142	573.2142	57
1	EXPLORATION	4312.429	431.2429	98
2	SOCIAL INTERACTION	3537.689	353.7689	109
3	HEAD BODY KNOCK	400.000	40.0000	80
4	LYING	82689.911	8268.9911	52
5	NOSING	160.000	16.0000	32
6	BLOCK OF FEEDING	55.000	5.5000	11
7	INTERACTION WITH OBJECTS	15.000	1.5000	3
8	MOUNT ATTEMPT	80.000	8.0000	16
9	PRESSING PARALLEL / INVERSE	30.000	3.0000	3
10	MOUNT SEXUAL ACTION	5.000	0.5000	1
11	BITE	55.000	5.5000	11

The results presented in Table 3 are obtained by analyzing the data provided by Boris. We observed that certain behaviors, such as social interaction and exploration, were the most frequent, while lying down was the behavior with the longest duration. This indicates that the swine were inactive for a significant portion of the time. The median duration per occurrence offers additional insights into the nature of the observed behaviors, highlighting the consistency or variation in the time spent on each activity. These results provide a solid foundation for a more profound understanding of the dynamic behavior patterns, which are crucial for model development.

To evaluate the performance of the trained models, we used various metrics available in the PyTorch library, specifically TorchMetrics. Among the metrics, we highlighted the F1-score and AUC:

- **F1-Score:** This metric provides a comprehensive view of the model's precision (the ability to correctly identify positive instances) and recall (the ability to capture all positive instances).
- **AUC:** This metric measures the model's ability to discriminate between positive and negative samples, indicating its overall performance in classification tasks.

All metrics were evaluated using the median value of each epoch, and the results for all different configurations are presented in Table 4. The models were trained for 10 epochs each.

Table 4: Metrics results in the trainning set

Model	FPS	Seconds Before	Avg accuracy	F1 Score	AUC score
1	3	5	0.92778	0.13082	0.29283
2	15	2	0.93904	0.13290	0.35660
3	15	2	0.92034	0.15485	0.37829
4	30	2	0.91899	0.45795	0.50194

- **First model:** Trained using 3 frames per second and including 5 seconds of prior footage. Initially, this approach seemed promising, but the results were not satisfactory. The accuracy was high, but this was due to correctly predicting the activities that were not present, rather than the correct ones. This discrepancy was evident in the results of the other metrics.
- **Second model:** Trained using 15 frames per second and including only 2 seconds of prior footage. This model aimed to improve upon the first, but the results remained similar. It achieved excellent accuracy but poor F1-score and AUC, indicating issues in correctly identifying the behaviors.
- **Third model:** After several unsuccessful attempts, we opted to predict only two behaviors by altering the CNN's classification function from sigmoid to softmax, focusing on predicting the

Figure 4: Grad-cam results

majority action: non-aggressive and aggressive behaviors. This model was trained using 15 frames per second with 2 seconds of prior footage, linking each frame to one pig's activity. The selected pig had a varied set of behaviors. While accuracy was still high, the other metrics did not show significant improvement.

- **Fourth model:** Similar to the third model, but focusing on all behaviors from only one pig to better recognize the occurring activity. This adjustment aimed to improve the model's ability to identify specific behaviors accurately.

Overall, while accuracy was high across models due to the class imbalance, the F1-score and AUC indicated that further refinement and different strategies are needed to improve the model's performance in behavior classification. After evaluating feature importance using Grad-CAM in Captum, the results, presented in Figure 4, highlighted specific areas for improvement.

In the image, it is noticeable that the color remains uniform, indicating that the model is not correctly identifying the proximity of the pigs as important. This area should be highlighted as a significant feature because it is where their activity is most visible.

Given the poor results in metrics and Captum analysis, we explored alternative approaches to solve the problem. One solution was to shift from providing the model with the same image containing all the behaviors of each pig at a given moment. Instead, we proposed giving the model smaller images, focusing on a small area around each pig. These images are labeled with the specific behavior, which could better help the model detect all behaviors in each frame of the video. To implement this, we introduced our second approach, which involved testing existing tracking devices.

By focusing on smaller, behavior-specific areas, we aim to enhance the model's ability to accurately classify behaviors. Additionally, utilizing tracking devices can provide more precise data, potentially improving the model's overall performance in real-time behavior classification.

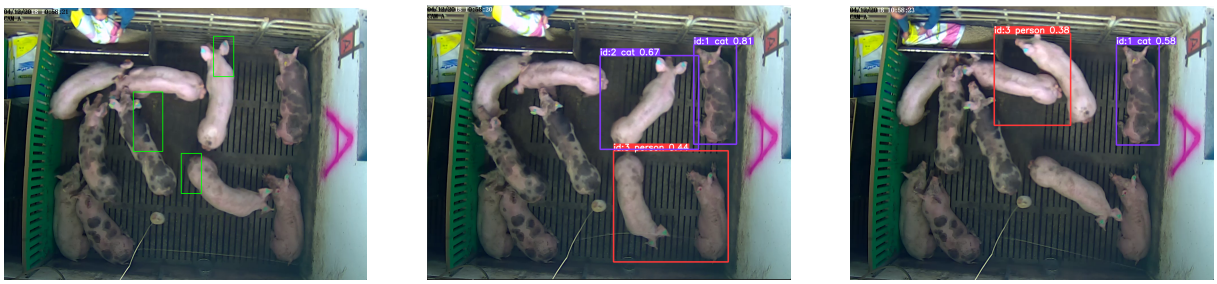
3.3 Second approach

Given the poor results observed in the previous approach, we attempted to refine our method by detecting individual pigs and extracting images centered around each pig. The goal was to feed the neural network with these focused images rather than using the original image containing all the pigs at once. This

approach aimed to reduce redundancy and mitigate the problem of predominantly predicting the same four most common behaviors: lying, social interaction, exploration, and eating.

Initially, we explored using optical flow, a technique that estimates the motion of objects between consecutive frames of a video. Optical flow can help identify moving objects, but it proved ineffective in our case due to the inactivity of the pigs. Minimal movements, such as those caused by light changes or external objects, resulted in false detections, making this method unsuitable for our needs.

Recognizing the limitations of optical flow, we turned to object tracking using the cv2.HOGDescriptor(), Figure 5a from the OpenCV library. This tracking model is trained to detect wild boars and pigs. However, the results were unsatisfactory as the model failed to detect all the animals in the frame and often mistook external object movements for pig movements.



(a) Hog descriptor (b) Yolo model m (c) Yolo model x

Figure 5: Tracking models results

Our final attempt with existing tracking models involved using YOLO (You Only Look Once) algorithms from Ultralytics, specifically yolov8m.pt and yolov8x. YOLO models are renowned for their real-time object detection capabilities. The yolov8m.pt is a medium-sized model balancing accuracy and speed, while yolov8x is an extended model with higher accuracy but more computationally intensive. Despite their advanced capabilities, these models also failed to detect most of the pigs accurately. When detections did occur, the pigs were often clustered together, as shown in Figures 5b, 5c. The detected class was not an issue since it could be adjusted in the code.

As these results were still not as satisfactory as we desired, we proceeded to our final approach, which involved experimenting with new tools and techniques beyond traditional models. This approach aimed to address the shortcomings of the previous methods and improve the detection and behavior analysis of the pigs.

3.4 Final approach

After numerous attempts to develop a model that could accurately and effectively detect all the pigs in each frame, we arrived at our final approach. In this approach, we integrate the most successful elements of the previous attempts and introduce additional implementations to support our goals. Specifically, we aim to predict two types of behavior: aggressive and non-aggressive, while also attempting to detect each pig individually. Unlike previous methods that involved tracking, this approach focuses on predicting the location of each pig in every frame.

The methodology of this approach is divided into three key parts:

1. **Preprocessing:** Extracting and preparing the data to ensure its quality and relevance for model training.
2. **Model Composition:** Building and configuring the model to leverage the strengths of the previous approaches and introduce new capabilities.

3. **Results:** Evaluating the performance of the model and analyzing its effectiveness in different scenarios.

3.4.1 Preprocessing

In our final approach, we aimed to build our own tracking model to address the tracking issues encountered earlier and provide more detailed data to the model. To begin, we developed a script to extract frames from the videos. Since the pigs were mostly inactive, we initially extracted frames at 2-minute intervals. However, this interval still resulted in many frames with inactive pigs. To create a more comprehensive dataset, we decided to extract frames every 30 seconds from the videos with the most movement. This method yielded 86 images: 69 images extracted at 2-minute intervals and 17 images extracted at 30-second intervals. The pseudocode for this extraction process is presented in Algorithm 2.

Algorithm 2 Extract Frames from Videos and Save as Images

Input: Directory of videos, Output path for frames **Output**: Extracted frames saved as images

```
List all video files in the input directory
Initialize counters for total frames and frame number
for each video in the list of videos do
    for each frame in the video do
        Increment total frame counter
        if total frame counter modulo 900 equals 0 or is the first frame then
            Increment frame number counter
            Save the current frame as an image in the output directory
    Update the starting frame index for the next video
```

After extracting the frames, we used a labeling tool called LabelMe to manually annotate the correct positions of all the pigs. LabelMe is an open-source graphical image annotation tool that allows users to create annotations in various formats for machine learning purposes. During the annotation process, we used bounding boxes to mark the location of each pig. Each pig was assigned a unique identifier, labeled as P1, P2, and so on up to P10. This method ensured that we could effectively track the activity of each pig and associate it with the corresponding bounding box. An example of an annotated image can be seen in Figures 6a, 6b.

By manually annotating the images, we ensured that the tracking model would have accurate data on the positions and identities of the pigs, allowing for a more precise and reliable analysis of their behaviors.

3.4.2 Model

In this approach, we employed a significantly different model architecture compared to our previous attempts. Instead of using a combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, we opted for a single CNN model. Specifically, we utilized a pre-trained ResNet50 model from the ImageNet dataset. This choice leverages the powerful feature extraction capabilities of ResNet50, a well-established deep learning architecture renowned for its performance in image classification tasks.

The model architecture is built as follows on Algorithm 3.

(a) LabelMe Image (b) Pigs position

Figure 6: LabelMe Example

Algorithm 3 Detector Model for Bounding Box and Behavior Prediction

Input: x - Input image/frame **Output**: Bounding box coordinates and behavior predictions for each pig

Initialize ResNet50 backbone network without the final layer
Initialize linear layers for bounding box predictions for each pig
Initialize linear layers for behavior predictions for each pig
$x \leftarrow$ Extract features using the backbone network
$x \leftarrow$ Flatten the extracted features
$bboxes \leftarrow$ [Sigmoid activation on bounding box predictors(x) for each pig]
$acts \leftarrow$ [Sigmoid activation on behavior predictors(x) for each pig]
return $bboxes, acts$

- **Backbone:** We utilized ResNet50, excluding its final classification layer. The backbone is composed of all layers up to the penultimate layer, which effectively extracts high-level features from the input images.
- **Behavior Predictions:** Similarly, for each pig, we used a linear layer to predict the behavior (aggressive or non-aggressive). This output is also passed through a sigmoid activation function to yield probabilities.

The model receives an input frame and processes it through the ResNet50 backbone to extract feature representations. These features are then fed into separate linear layers to predict the bounding boxes and behaviors for each pig in the frame. The model outputs the predicted coordinates for each pig's bounding box and a behavior classification (aggressive or non-aggressive).

This architecture simplifies the detection and behavior classification task by using a powerful pre-trained network for feature extraction, allowing us to build on a robust foundation and tailor the final layers to our specific needs.

3.4.3 Results

In this approach, we divided the dataset into 80% for training and 20% for testing, with the testing set comprising the most recent images collected. The model was trained for varying numbers of epochs, as shown in Table 5.

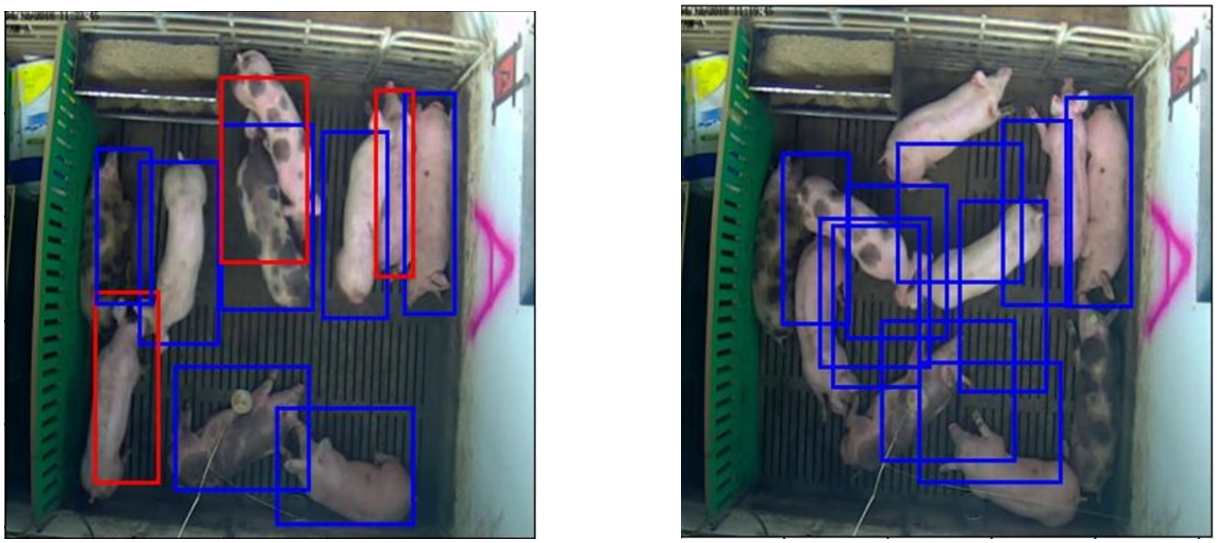
Table 5: Tracking results in the test set

Epochs	mAP	mAP@50	mAP@75	mAP per pig									
				1	2	3	4	5	6	7	8	9	10
200	0.0221	0.0735	0.0182	0.1444	0.2111	0.1722	0.0556	0.0333	0.1278	0.0056	0.0389	0.0333	0.0000
500	0.0224	0.0816	0.0091	0.1111	0.2333	0.1278	0.0667	0.0667	0.1167	0.0333	0.0778	0.0611	0.0222
1000	0.0732	0.1779	0.0389	0.1222	0.4333	0.1333	0.0778	0.1278	0.1722	0.0667	0.0833	0.1056	0.0722
5000	0.0810	0.1756	0.0611	0.1611	0.4444	0.1556	0.1167	0.1611	0.1944	0.0722	0.1111	0.1444	0.1444
20000	0.1439	0.2886	0.1475	0.1944	0.6167	0.3667	0.1444	0.1667	0.2000	0.1444	0.2944	0.1556	0.1722

During testing, we observed consistent performance improvements with an increasing number of epochs. However, it became evident that the model was overfitting to the training images due to the relatively small size of the training set. Despite this, we used the model trained for 1000 epochs for evaluation, as it provided the best performance in terms of convergence and overall accuracy.

To evaluate the effectiveness of our implementation, we divided the results into two distinct phases: the Tracking Phase and the Detection Phase.

Images 7a and 7b illustrate the best and worst detections with both phases implemented. Image 7a shows an example where the model accurately detected and classified the bounding boxes and behaviors, while Image 7b highlights a scenario where the model struggled, providing insights into areas for further improvement.



(a) Best detection

(b) Worst detection

Figure 7: Final approach results

- **Tracking phase:** In the Tracking Phase, we focused on assessing the accuracy and effectiveness of our model in detecting and tracking pigs in the test images. The key metric used for this evaluation was the Mean Average Precision (mAP), which is commonly used in object detection tasks.

  This metric evaluates the accuracy of the bounding box predictions by measuring the overlap between the predicted bounding boxes and the ground truth boxes. The overlap is measured using Intersection over Union (IoU), which calculates the area of overlap between the predicted and true bounding boxes divided by the area of their union.

  - **mAP@50:** This measures the percentage of bounding boxes that have an IoU greater than 50% with the ground truth boxes.
  - **mAP@75:** This measures the percentage of bounding boxes that have an IoU greater than 75% with the ground truth boxes.

These metrics provide insights into how well the model is detecting the pigs and how accurately it is localizing them within the images. In Table 5, we present the results of these metrics, showing the performance of our model on the test set.

- **Detection phase:** The Detection Phase aimed to classify the behaviors of the detected pigs as either aggressive or non-aggressive. This phase was crucial for understanding the context of the detected behaviors and for validating the model's ability to distinguish between different types of interactions among pigs.

  In this phase, we modified the colors of the bounding boxes to visually differentiate between aggressive and non-aggressive behaviors. Aggressive behaviors were marked with one color (e.g., red), while non-aggressive behaviors were marked with another color (e.g., blue). This visual differentiation allowed for an intuitive and immediate understanding of the behavior classification results. One example of this separation can be seen in Figure 7a.

  To evaluate this part accuracy was utilized and were obtain a value of 0.911 which is pretty good but this result is not fidedigne as we want once most of the behaviour that append on the videos were non agressive which can lead to overfitting with non-agressive behaviours.

Overall, the results from both phases indicate that our model is capable of detecting and tracking pigs with reasonable accuracy, as well as distinguishing between aggressive and non-aggressive behaviors. However, there are still challenges to be addressed, particularly in improving the precision of the bounding box predictions and the robustness of behavior classification.

Further enhancements to the model and additional training data may help in overcoming these challenges and improving the overall performance of our tracking and detection system.

4 Final Product

Among all the attempts made, our final approach demonstrated the most promising results, showing significant potential for further improvement. This approach involved selecting a model to test on new videos to predict the individual behavior of each pig, as outlined in the project proposal.

To thoroughly evaluate the model's performance, we tested it on five different videos. These videos were chosen based on various criteria to ensure a comprehensive assessment of the model:

- **Training Video:** A video used in training to verify model consistency
- **Same Pen House, Different Time:** Two videos from the same pen house (Pen House A) but recorded in the afternoon, to assess temporal generalization.
- **Different Pen House:** Two videos from a different pen house (Pen House D) with distinct environmental features, to test spatial generalization and adaptability.

The evaluation aimed to examine how well the model predicts individual pig behaviors across different settings and times, providing insights into its robustness and areas for further enhancement. It's important to note that we cannot showcase the videos used for evaluation due to intellectual property constraints.

1. **Pig activity - train video:**
   - **Characteristics:** This video was part of the training set.
   - **Results:** In the training video, the model's bounding boxes did not achieve 100% accuracy but were able to detect some aggressive behaviors. For example, when one pig approached another and knocked its head against the other, this was classified correctly as aggressive behavior.

2. **Pig activity - Pen A:**

    - **Characteristics:** These two videos were from the same pen house used in the training set (referred to as Pen House A) but filmed in the afternoon.
    - **Results:** In these videos, which share the same environment as the training set, the model's bounding box predictions were also imperfect. The model struggled to detect aggressive behaviors, even though some incidents appeared to occur during the video.

3. **Pig activity - Pen D:**

    - **Characteristics:** These two videos were from a different pen house (referred to as Pen House D), which has significantly different characteristics compared to Pen House A. Pen House D offers a larger space for the pigs to move around, has two feeders and drinking spots, and features pigs that were completely unknown to the model.
    - **Results:** In these videos, the environment was completely different from what the model had seen during training. As expected, the bounding box predictions were poor. This occurred because the model was trained exclusively with data from Pen House A, which has a specific size and layout, leading to difficulties when applied to a different environment. Despite this, the model showed some capability in predicting activity. At the start of the video, some pigs were fighting, which the model did not detect. However, it managed to identify some subtle aggressive episodes when one pig's head approached another. The limited detection could be attributed to the few aggressive episodes and their specific nature in the training videos/images.

After reviewing the results, it is evident that the model is not yet accurate enough for deployment, as it makes mistakes in detection and has issues classifying certain evident behaviors. Despite this, a brief meeting with the client indicated that most of the objectives were met with this model approach. Individual detection and behavior prediction were achieved, though not with very accurate results, indicating a starting point for future improvements. Additionally, the preparation for real-time detection was completed. However, our approach lacks the ability to be used on a microcontroller, as this was not tested. The results were not entirely satisfactory for the client's expectations, particularly in accurately detecting aggressive behaviors, which is crucial for preventing potential damage among pigs.

To address these shortcomings, we identified key areas for improvement. The problems may stem from the quality or quantity of the training data. To overcome these challenges, we propose the following changes:

- **Diverse Training Data:** Train the model with a more diverse dataset, including videos from multiple pen houses with varying environments. This will help the model generalize better across different settings and effectively function in other pen houses.
- **Diverse Action Training Data:** Incorporate a more varied action dataset, with detailed images of different behaviors, especially focusing on increasing the quantity of aggressive behavior instances. This will enhance the model's ability to detect these critical behaviors accurately.
- **Model Refinement:** Refine the model architecture to improve its generalization across different conditions and pig behaviors. This might involve tweaking the layers, activation functions, or other components of the neural network.
- **Incorporate Movement:** Implement a Long Short-Term Memory (LSTM) network or another sequential model to provide the model with a sequence of images, simulating movement. This will help the model understand the progression of actions more effectively.

- **Regular Updates:** Regularly update the model with new data to continuously improve its accuracy and adaptability. This will ensure the model stays relevant and accurate as new behaviors and environmental conditions are introduced.

By implementing these changes, we aim to enhance the model's performance, particularly in detecting aggressive behaviors, and ensure it meets the client's requirements more effectively.

5 Conclusion and future work

This project aimed to develop a model capable of predicting the individual behavior of pigs in various environments, particularly focusing on detecting aggressive behavior due to the problems it causes to others in the same environment. Multiple approaches were attempted, but their results were not fully satisfactory. The final approach was selected due to its potential for improvement and its better performance compared to the others. Through our iterative approach and testing on different video datasets, several key conclusions can be drawn:

1. **Model Performance in Known Environments:**

    - The model performed moderately well in environments similar to those it was trained on, such as Pen House A. While not perfect, it was able to detect some aggressive behaviors, indicating a baseline level of functionality.
    - In the training video, the model correctly identified aggressive interactions, though it did not achieve complete accuracy in bounding box predictions, possibly due to the similarity between pigs.

2. **Challenges in New Environments:**

    - The model struggled significantly when applied to Pen House D, an environment vastly different from the training conditions. The poor performance in detecting bounding boxes and aggressive behaviors highlights the model's limited generalizability.
    - This discrepancy underscores the importance of training with diverse datasets to improve the model's robustness and adaptability.

3. **Detection Limitations:**

    - Even within familiar environments, the model's detection of aggressive behavior was inconsistent. This suggests that the current model architecture and training data may not be fully capturing the nuances of pig behavior. The training set included subtle aggressive behaviors, leading to missed detections of more evident behaviors.

To further enhance the effectiveness and applicability of our model in detecting and predicting pig behavior, several key aspects need to be addressed in future work. These recommendations aim to improve the model's generalizability, accuracy, and robustness across different environments and scenarios:

1. **Diversification of Training Data:**

    - Incorporating videos and images from various pen houses with different layouts, sizes, and pig populations can help the model learn a wider range of environmental and behavioral patterns. This will likely improve its performance across different settings and environments.

2. **Model Architecture Improvements:**

    - Exploring advanced model architectures, such as those utilizing more complex layers in convolutional neural networks (CNNs) or incorporating recurrent neural networks (RNNs or LSTMs) to capture temporal dependencies, could enhance the model's predictive capabilities.
    - Additionally, integrating techniques like data augmentation and transfer learning can help improve the model's ability to generalize to new environments. It might also be beneficial to implement a method to count the number of aggressive behaviors each pig provokes to assess its placement in the pen house.

3. **Evident Behaviors in Training Data:**

    - The training data should include a variety of behaviors, both evident and subtle, to enable the model to predict accurately when a behavior occurs.

4. **Regular Model Updates:**

    - Continuously updating the model with new data and retraining it periodically can help maintain and improve its accuracy. This practice ensures that the model evolves alongside changes in pig behavior and environmental conditions.

5. **Behavioral Analysis Enhancements:**

    - Future work could focus on refining the criteria for classifying behaviors, possibly integrating expert knowledge from animal behaviorists. This can lead to more accurate and meaningful predictions. Additionally, upgrading the model to detect a broader range of behaviors, such as eating, biting, and lying down, without generalizing, would be beneficial.

In conclusion, while the current model shows promise in detecting pig behavior, significant improvements are needed to enhance its robustness and accuracy across varied environments. By addressing the outlined recommendations, future iterations of the model can better meet the project's goals and provide valuable insights for managing pig behavior in diverse settings.

References

[1] Yang, M., Wu, C., Guo, Y., Jiang, R., Zhou, F., Zhang, J., Yang, Z.: Transformer-based deep learning model and video dataset for unsafe action identification in construction projects. Automation in Construction 146, 104703 (2023), https://www.sciencedirect.com/science/article/pii/S0926580522005738

[2] Yao, G., Lei, T., Zhong, J.: A review of convolutional-neural-network-based action recognition. Pattern Recognition Letters 118, 14–22 (2019), https://www.sciencedirect.com/science/article/pii/S0167865518302058, cooperative and Social Robots: Understanding Human Activities and Intentions

[3] Varol, G., Salah, A.A.: Efficient large-scale action recognition in videos using extreme learning machines. Expert Systems with Applications 42(21), 8274–8282 (2015), https://www.sciencedirect.com/science/article/pii/S0957417415004078

[4] Segalin, C., Williams, J., Karigo, T., Hui, M., Zelikowsky, M., Sun, J.J., Perona, P., Anderson, D.J., Kennedy, A.: The mouse action recognition system (mars) software pipeline for automated analysis of social behaviors in mice. eLife 10, e63720 (nov 2021), https://doi.org/10.7554/eLife.63720

[5] Jiang, B., Yin, X., Song, H.: Single-stream long-term optical flow convolution network for action recognition of lameness dairy cow. Computers and Electronics in Agriculture 175, 105536 (2020), https://www.sciencedirect.com/science/article/pii/S0168169920311170

[6] Simanungkalit, G., Barwick, J., Cowley, F., Dawson, B., Dobos, R., Hegarty, R.: Use of an ear-tag accelerometer and a radio-frequency identification (rfid) system for monitoring the licking behaviour in grazing cattle. Applied Animal Behaviour Science 244, 105491 (2021), https://www.sciencedirect.com/science/article/pii/S0168159121002781

[7] Meunier, B., Pradel, P., Sloth, K.H., Cirié, C., Delval, E., Mialon, M.M., Veissier, I.: Image analysis to refine measurements of dairy cow behaviour from a real-time location system. Biosystems Engineering 173, 32–44 (2018), https://www.sciencedirect.com/science/article/pii/S1537511017302179, advances in the Engineering of Sensor-based Monitoring and Management Systems for Precision Livestock Farming

[8] Fuentes, A., Yoon, S., Park, J., Park, D.S.: Deep learning-based hierarchical cattle behavior recognition with spatio-temporal information. Computers and Electronics in Agriculture 177, 105627 (2020), https://www.sciencedirect.com/science/article/pii/S0168169920307110

[9] Zhu, X., Chen, C., Zheng, B., Yang, X., Gan, H., Zheng, C., Yang, A., Mao, L., Xue, Y.: Automatic recognition of lactating sow postures by refined two-stream rgb-d faster r-cnn. Biosystems Engineering 189, 116–132 (2020), https://www.sciencedirect.com/science/article/pii/S153751101930892X

[10] Zhang, K., Li, D., Huang, J., Chen, Y.: Automated video behavior recognition of pigs using two-stream convolutional networks. Sensors 20(4) (2020), https://www.mdpi.com/1424-8220/20/4/1085

[11] Yang, Q., Xiao, D.: A review of video-based pig behavior recognition. Applied Animal Behaviour Science 233, 105146 (2020), https://www.sciencedirect.com/science/article/pii/S0168159120302343

[12] Chen, C., Zhu, W., Steibel, J., Siegford, J., Wurtz, K., Han, J., Norton, T.: Recognition of aggressive episodes of pigs based on convolutional neural network and long short-term memory. Computers and Electronics in Agriculture 169, 105166 (2020), https://www.sciencedirect.com/science/article/pii/S0168169919319556

[13] Zhang, K., Fu, J., Liu, D.: Flow-guided transformer for video inpainting (2022), https://arxiv.org/abs/2208.06768

[14] Li, Z., Gavrilyuk, K., Gavves, E., Jain, M., Snoek, C.G.: Videolstm convolves, attends and flows for action recognition. Computer Vision and Image Understanding 166, 41–50 (2018), https://www.sciencedirect.com/science/article/pii/S1077314217301741

[15] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 618–626 (2017)