

AI-ML Powered Pig Behavior Classification and Body Weight Prediction

Sanjana M Bharadwaj

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Sook Shin Ha, Chair

Dong Ha

Creed F. Jones

May 2, 2024

Blacksburg, Virginia

Keywords: Image Processing, Machine Learning, Deep Learning, Precision Livestock
Farming

Copyright 2024, Sanjana M Bharadwaj

AI-ML Powered Pig Behavior Classification and Body Weight Prediction

Sanjana M Bharadwaj

(ABSTRACT)

Precision livestock farming technologies have been widely researched over the last decade. These technologies help in monitoring animal health and welfare parameters in a continuous, automated fashion. Under this umbrella of precision livestock farming, this study focuses on activity classification and body weight prediction in pigs. Activity monitoring is essential for understanding the health and growth of pigs. To automate this task effectively, we propose efficient and accurate sensor-based deep learning (DL) solutions. Among these, the 2D Residual Networks emerged as the best performing model, achieving an accuracy of 95.6%. This accuracy was 15.6% higher than that of other machine learning approaches. Additionally, accurate pig weight estimation is crucial for pork production, as it provides valuable insights into growth rates, disease prevalence, and overall health. Traditional manual methods of estimating pig weights are time-consuming and labor-intensive. To address this issue, we propose a novel approach that utilizes deep learning techniques on depth images for weight prediction. Through a custom image preprocessing pipeline, we train DL models to extract meaningful information from depth images for weight prediction. Our findings show that XceptionNet gives promising results, with a mean absolute error of 2.82 kg and a mean absolute percentage error of 7.42%. In comparison, the best performing statistical model, support vector machine, achieved a mean absolute error of 4.51 kg mean absolute percentage error of 15.56%.

AI-ML Powered Pig Behavior Classification and Body Weight Prediction

Sanjana M Bharadwaj

(GENERAL AUDIENCE ABSTRACT)

With the increasing demand for food production in recent decades, the livestock farming industry faces significant pressure to modernize its methods. Traditional manual tasks such as activity monitoring and body weight measurement have been time-consuming and labor-intensive. Moreover, manual handling of animals can cause stress, negatively affecting their health. To address these challenges, this study proposes deep learning-based solutions for both activity classification and automated body weight prediction. For activity classification, our solution incorporates strategic data preprocessing techniques. Among various learning techniques, our deep learning model, the 2D Residual Networks, achieved an accuracy of 95.6%, surpassing other approaches by 15.6%. Furthermore, this study also compares statistical models with deep learning models for the body weight prediction task. Our analysis demonstrates that deep learning models outperform statistical models in terms of accuracy and inference time. Specifically, XceptionNet yielded promising results, with a mean absolute error of 2.82 kg and a mean absolute percentage error of 7.42%, outperforming the best statistical model by nearly 8%.

Dedication

To my mom, dad, brother and grandparents. Whatever I am today is because of them.

Acknowledgments

Firstly, I would like to extend my heartfelt gratitude to my advisor, Dr. Sook Shin Ha, who has played a pivotal role in my graduate studies. Her expertise, attention to detail, guidance & patience has helped me get better as a student and researcher. I value the one-on-one sessions I had with her where we discussed ideas, research questions and career progression among many other things. I am sure that the lessons I have learnt from her would stay with me forever.

I also would like to thank my committee members, Dr. Dong Ha for his constant support and encouragement, Dr. Creed Jones, for his kind involvement and suggestions. A special thanks to Dr. Gota Morota from the School of Animal Sciences, his students, Kenan and Ye for their constant support and timely help in our projects.

A big thank you to my research teammates, Pranjal Ranjan and Yingqi Pei who were always ready to try out that new idea, spend hours working on the code and go that extra mile to help one another out. I could not have asked for a better team than these amazing people.

A huge shout out to my rock-solid support system, my parents and my brother, Shashaank. Their belief in me, words of encouragement equipped me with the motivation and energy to work throughout my graduate studies. Last but definitely not the least, I would like to thank my friends here in Blacksburg - Gauri, Prajakta, Nikhil & Rohit for being with me through thick and thin. A special mention to my friends back in India, Richa and Manasi, who ensured that despite the time differences, I was doing well.

Contents

List of Figures	x
List of Tables	xiii
1 Introduction	1
1.1 Motivation & Background	1
1.1.1 Behavior Classification	3
1.1.2 Weight Prediction	5
1.2 Contributions	7
2 Review of Literature	8
2.1 Precision Livestock Management	8
2.2 Sensor-based Behavior Classification	8
2.2.1 Machine Learning Solutions	9
2.2.2 Deep Learning Solutions	10
2.3 Image-based Pig Body Weight Prediction	10
2.3.1 Digital Image Processing for Weight Prediction	11
2.3.2 Depth Images Processing for Weight Prediction	12
2.3.3 Point Cloud Data Processing for Weight Prediction	12

3	Behavior Classification	14
3.1	Data Acquisition	14
3.1.1	Sensors	14
3.1.2	Sensor Placement & Camera Setup	15
3.1.3	Data Collection	16
3.1.4	Data Labeling	17
3.2	Experimental Setup	18
3.2.1	Data Preprocessing	18
3.2.2	Feature Extraction	18
3.2.3	Proposed Time Window Segmentation Method	19
3.2.4	Proposed Chronological Train/test split	21
3.3	Models	24
3.3.1	Machine Learning Models	24
3.3.2	Deep Learning Models	25
3.4	Results & Discussions	27
4	Body Weight Prediction	31
4.1	Data Acquisition	31
4.1.1	Camera Setup	31
4.1.2	Data Collection	32

4.1.3	Data Labeling	32
4.2	Experimental Setup	33
4.2.1	Data Preprocessing	33
4.2.2	Object Detection	33
4.2.3	Segmentation	34
4.2.4	Cropping	35
4.2.5	Cleaning	36
4.2.6	Data Augmentation & Normalization	37
4.2.7	Cross-Validation	38
4.3	Models	39
4.3.1	Statistical Models	39
4.3.2	Deep Learning Models	40
4.4	Results & Discussions	45
5	Conclusions	49
	Bibliography	52
	Appendices	60
	Appendix A Segment Anything Model	61
	Appendix B Model Details	63

B.1 Residual Networks	63
---------------------------------	----

List of Figures

1.1	Flow of data in a typical precision livestock farming application	3
1.2	Illustration of data leakage problem	5
3.1	The MMC circuit board (left and center), a hand holding a coin next to a cell phone (right). The pictures are courtesy of MBIENTAB (https://mbientlab.com).	15
3.2	The coin sensor and the sensor case used in the study	16
3.3	Principal Component Analysis of Accelerometer data along three axes. This graph shows clustering of data points together showing correlation between points.	21
3.4	Principal Component Analysis of Gyroscope data along three axes. This graph shows clustering of data points together showing correlation between points.	22
3.5	Distribution of activity labels in the train dataset. This shows that the train dataset has a good balance of all activity labels.	23
3.6	ML-based pig activity classification results with different data preparation methods on (a) time domain data and (b) feature domain data.	30
4.1	An overview of the image preprocessing pipeline. This pipeline utilizes segmentation masks to implement most of the tasks in the pipeline. These methods help achieve a cleaner dataset that minimizes extraneous noise in the images.	33

4.2	Segment Anything Model automatic mask generation. The model takes as input both image and box prompts for generating segmentation masks. . . .	34
4.3	Segment Anything Model (SAM) segmentation masks. SAM does not require any additional training to provide the output on custom dataset images. . .	35
4.4	Cropping pigs' body using the bounding box coordinates generated by YOLO-V7 model.	36
4.5	Identifying pigs on edges using the segmentation mask generated by SAM. This example clearly shows the pig body to be distorted at the edges.	37
4.6	Data Augmentation techniques. On the original image (left), we applied rotate by 90 degrees (center) and flipping (right) operations to add more variations in the existing dataset.	38
4.7	Extracting biometric features such as length (L), width (W), and height (H) of the pig contour in pixel space. The point denoting the height is the center of the pig's body. These features are estimated using depth images.	39
4.8	Example of an original depth image and its multiresolution analysis (MRA) decomposition using wavelet transform. MRA of an image helps capture both fine details and coarse features simultaneously which helps to analyze the image.	42
4.9	Multi-input 2D-CNN architecture. Here, the integration of two modalities is achieved by fusing their features internally. The parallel processing helps to improve feature extraction for better accuracy in body weight prediction. . .	43
4.10	ResNet-34 architecture. The skip connections used in the model help propagate the gradient value from one layer to the other, thus reducing the effect of vanishing gradient problem.	43

4.11	XceptionNet architecture used in our design. We see that this model utilizes separable convolution to efficiently train and pass weight in deep layers. . . .	45
4.12	Comparison of Statistical and deep learning model performance on weight evaluation. Deep learning models clearly outperform their statistical model counterparts.	48
A.1	Data Flow in the SAM model. We can see here that once an image is given an input it goes through encoders and concatenates with encoded prompts to finally go through decoder to get the final output. Courtesy : https://ai.meta.com/blog/segment-anything-foundation-model-image-segmentation/	62

List of Tables

3.1	Percentage of Six Major Behaviors	17
3.2	Comparison of classification performances on datasets with proposed train/test split vs other train/test split	28
3.3	Comparison of classification performances on datasets segmented with proposed windowing method vs segmented with other windowing method	29
3.4	Comparison of classification performances on datasets with proposed split and windowing vs other split and windowing	29
4.1	Basic Statistics of Ground Truth Weight Values	32
4.2	Weight Prediction Results of Statistical Methods	46
4.3	Weight Prediction Results of Deep Learning Methods	47

List of Abbreviations

CNN Convolutional Neural Networks

DL Deep Learning

KNN K-Nearest Neighbors

LSTM Long Short Term Memory

ML Machine Learning

PLF Precision Livestock Farming

RF Random Forest

SVM Support Vector Machine

Chapter 1

Introduction

1.1 Motivation & Background

The livestock industry is a cornerstone of the global food system and plays a vital role in ensuring food security, and advancing agricultural development. According to the Food and Agriculture Organization (FAO), livestock accounts for 40% of the global agricultural value and sustains the livelihoods and food security of nearly 1.3 billion individuals worldwide. Animal production industry in its traditional format is seen to be a market with marginal profits. This makes growth, reproduction, and well-being monitoring of each animal crucial for a profitable business. Additionally, there is a huge focus on improving livestock sector practices so that they are more viable, more profitable, and pose less risk to animal and human health.

Before moving forward, let us understand what we mean by, 'livestock'. Livestock encompasses domesticated animals that are reared for purposes such as food, fiber, labor, and other products. This category includes animals such as cattle, swine, poultry, sheep, and goats, among others. Livestock is integral to agriculture and serves as a crucial component of the food production industry.

On the other hand, livestock farming involves raising animals for a range of purposes including food production, clothing, and labor. Globally, livestock farming is integral to both the economy and cultural practices. In the United States, for instance, the livestock industry contributes billions of dollars in revenue annually and sustains millions of jobs.

Recent trends reveal three significant challenges on a global scale emerging from livestock farming. Firstly, the demand for livestock products such as meat, eggs, and dairy is growing rapidly due to the rising global population and the increasing accessibility of these goods. Secondly, the number of livestock farmers and the extent of farmland have been declining steadily for years, primarily due to limited land availability. Lastly, there is a growing concern about the negative impact of livestock on major global issues such as climate change, deforestation, and environmental degradation.

To tackle these challenges, emerging solutions in the field of precision livestock farming (PLF) show promise, although they are still in the early stages of widespread adoption.

According to Berckmans, PLF is defined as the, “*management of individual animals by continuous, automated, and real-time monitoring of health, welfare, production/reproduction, and environmental impact*” . Precision Livestock Farming (PLF) is a cutting-edge method that leverages technology and data analytics to enhance livestock production and management. It combines sensors, automation, and monitoring systems to collect real-time data on animal health, behavior, and environmental factors, as shown in Figure 1.1. This information empowers farmers to make strategic decisions regarding nutrition, health care, breeding, and general animal welfare. PLF seeks to boost productivity, reduce resource wastage, improve animal welfare, and support sustainable and efficient livestock farming methods.

Our study majorly focuses on pigs, we try to address research questions around how we can improve the health monitoring and pork production. We collaborated with the Virginia Tech School of Animal Sciences to get high quality raw data and understand domain specific concepts.

There are two facets to our study, firstly, where we focus on identifying the activities of pigs using sensor-based analysis. Secondly, we propose methods for automatic weight prediction of pigs using depth images. In both the projects, we primarily highlight the use of deep learning solutions and show that these approaches give better results than the other solutions present in the literature. We talk in more detail about these two projects in the following sub-sections.

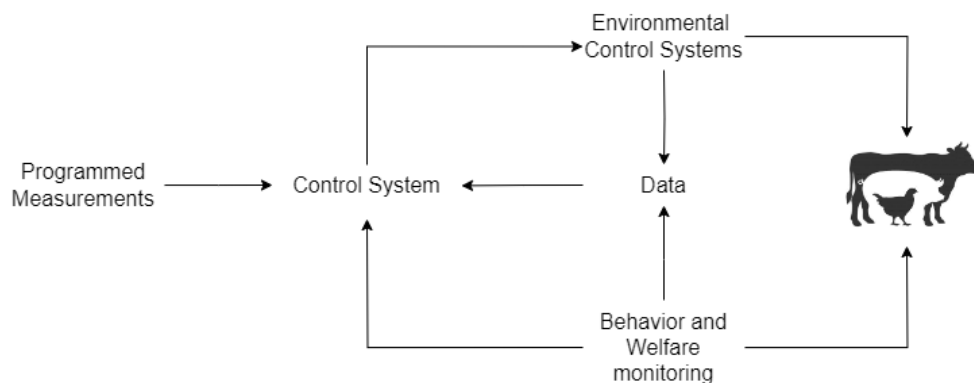


Figure 1.1: Flow of data in a typical precision livestock farming application

1.1.1 Behavior Classification

Pigs, like other animals, display various behaviors that can signal their comfort levels and general health [28, 32]. There is a direct correlation between improving pork production and ensuring animal care and welfare. For example, pigs are known to be comparatively more sensitive to external environmental conditions. They tend to show heightened activity levels as a direct reaction to some internal discomfort they are experiencing.

For instance, to beat the heat stress and decrease the heat and improve thermoregulation, pigs reduce movement [31]. This shows that animal behavior has a direct correlation with their well-being in general, making the task of behavior monitoring important.

Our main motivation stemmed from the fact that historically, monitoring these behaviors involved manual, labor-intensive methods that struggled with providing continuous oversight and were prone to human error. Moreover, manual intervention in monitoring activities can prove to be subjective and therefore, not very reliable.

We believe that introducing sensor-based pig monitoring system can bring in the required objectivity in this task. Recent technological progress has attested for the sensor and video camera-based monitoring systems, that when combined with machine learning and deep learning approaches, offer improved efficiency and continuous monitoring [1, 52]. Sensors like accelerometers and gyroscopes are heavily used in such tasks. Accelerometers help identify animal activities based on their body movements by calculating the linear acceleration. Additionally, gyroscopes give us the information about the orientation and rotation of the animal. These two sensors were mounted on a micro controller designed to work as a wireless sensor node (WSN), which was used to capture the raw data in our study.

Notably, several studies have overlooked the data leakage problem – that refers to the use of leaked information during model training and validation which would not be available in the prediction stage, as shown in Figure 1.2. This issue mostly occurs when the test data spills over to the training/validation data, thereby making the results invalid. Oftentimes, this can lead to overly optimistic modeling performance (overfitting) on the test data, since the patterns of the data were already revealed to the model during training. Consequently, the same model can show poor performance when it is applied to a new, unlabeled data.

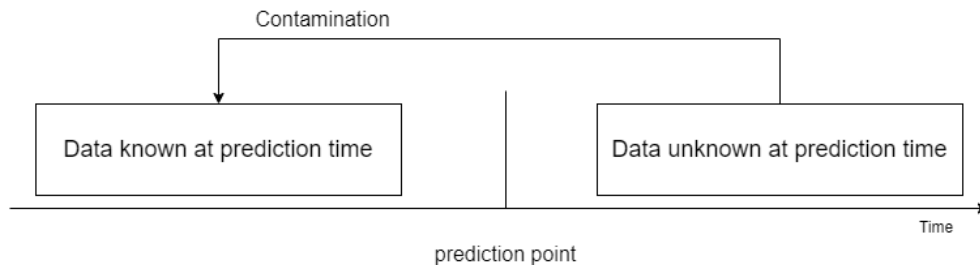


Figure 1.2: Illustration of data leakage problem

In our study, we have proposed two methods such as "Non-class-based data segmentation" and "Chronological train/test split". Post this, we extracted features and applied different machine learning models and trained deep learning models on raw data. Additionally, we study how our proposed data processing methods work as opposed to the regular "class-based segmentation" and "randomly splitting the data". Our proposed techniques ensure that there is minimal data leakage and the classification results obtained reflect the true performance of the models.

1.1.2 Weight Prediction

Pig health is vital for efficient production, with body weight serving as a crucial indicator of growth and well-being, influencing feed conversion rate and growth rate [5], [20]. Traditional weighing methods entail manually transporting pigs to scales, causing stress to the animals and posing challenges in monitoring large populations regularly [11], [18].

To address these challenges, researchers are exploring automated weight prediction systems. These systems utilize non-invasive and efficient techniques such as image data and 3D depth data for weight prediction. In this study, we propose using a 3D depth video imaging system combined with deep learning models to estimate pig weight. The simplicity of our approach is a key highlight, offering both simplicity and accurate results.

Traditional automated weight prediction systems have relied on machine learning-based approaches, which require manual extraction of morphological identifiers. However, these methods often suffer from low performance due to the need for feature extraction. Therefore, we provide a comprehensive comparative analysis between traditional statistical methods and modern deep learning (DL) techniques. Our study demonstrates the superior performance of DL methods when trained and tested on the same data.

1.2 Contributions

This thesis majorly focuses on the use of deep learning (DL) solutions on the tasks in precision livestock farming. We want to be able to show that DL techniques provide better flexibility, generalizability and higher performance overall. The main contributions of this study can be summarized as follows:

- Behavior Classification
 - Proposed strategic data preprocessing techniques for time-series sensor data ensuring to avoid data leakage, non-class-based data segmentation and chronological train/test split
 - Deep learning-based solution for activity classification, with our best performing model, 2D residual networks achieving an accuracy of 95.6%
- Body Weight Prediction
 - Proposed a simple and efficient image preprocessing pipeline to work on depth images
 - Presented a comprehensive comparative analysis, evaluating traditional statistical methods alongside DL techniques on a custom-collected dataset to ensure a fair comparison
 - Showed that DL-based models, trained directly on raw depth images with minimal automated preprocessing, outperformed statistical methods, both in terms of accuracy and inference time

We have submitted two papers inspired from these two projects to the IEEE Multimedia Information Processing and Retrieval (MIPR) 2024.

Chapter 2

Review of Literature

2.1 Precision Livestock Management

In modern agriculture, pigs are essential protein sources, crucial for meeting global food demands [41], [9]. This increased demand has led to a surge in pig production, with precision livestock farming gaining momentum to meet these needs [4]. Technologies are being developed for continuous and automated monitoring of animal health and welfare parameters, offering opportunities to enhance productivity and detect health issues at early stages[39].

An interesting study by the Food and Agriculture Organization (FAO) reveals that by the year 2050, there is likely going to be an increase of 46% globally in comparison with 2012. They also discuss number pertaining to each field, such as pigs are going to increase three-fold, poultry over five-folds and cattle is predicted to grow by two-folds [23].

2.2 Sensor-based Behavior Classification

In the case of pigs, activity recognition systems typically monitor various behaviors, such as lying down, standing, drinking, eating, walking and interacting with one another [17, 37, 49, 51]. In [38], the author discussed how manual monitoring is not practical in a commercial setup. The author here, further proposes measuring standing, feeding and drinking activities

using only a single vision sensor providing 3D trajectories. However, use of only a single vision sensor did not help in validating other behavior categories like running, laying, etc., which can be seen as a downside of this study.

Some other studies used only accelerometer data to classify behaviors with the help of heuristically tuned thresholds, with time and frequency domain statistical parameters. In [12], proposes using 3D accelerometer data (acceleration in x,y,z directions) for determination of cows in standing position. They extract statistical data from sensor and estimate the values representing standing apart from lying. It is seen that these thresholds are fixed manually by analyzing the data.

2.2.1 Machine Learning Solutions

Previous studies have used traditional machine learning algorithms with varying success, often relying on hand-crafted features extracted from raw sensor data [6, 7, 10, 21]. Interestingly, in [21], the authors propose the idea of forming a collective ensemble of classifiers built in different domains such as, change, frequency and time. These are complicated machine learning algorithms which require the feature domain data for training and testing.

Furthermore, [7] calculates almost 585 features and applied on standard machine learning algorithms like support vector machines, decision trees, etc. This study used the human activity recognition dataset to classify activities like standing, walking, lying, sitting, and so on. This feature extraction process can be time-consuming and may not effectively capture complex patterns in the data. Moreover, the features are not sensor-agnostic. Different features will have to be calculated for different sensors. To overcome these challenges posed by traditional ML methods, in the section below, we discuss how DL has revolutionized analysis.

2.2.2 Deep Learning Solutions

The potential for leveraging deep learning algorithms for this task on motion sensor data remains largely unexplored in the field of agriculture and livestock [22, 46]. With deep neural learning transforming the way we handle data, it was surprising to see that studies had not leveraged this for the activity classification task in this domain.

To get a better understanding of how deep neural networks can be leveraged for time series data in general, [25] gives a comprehensive summary around this topic. The paper deep dives into the details of designing convolutional neural networks (CNN) and recurrent neural networks (RNN) for the tasks such as classification, forecasting, anomaly detection, etc. The paper proposes using fully connected networks (FCN) allowing input and output signals to have same dimensions.

An interesting point we picked from [47], is how we can leverage CNN to work for time series data by transforming time-series data into an image. This is where our idea of using 2D-CNN in our study comes from. Packaging 1-D data as 2-D can help leverage many deep learning models known to work well on images.

Geng et al. [26], talks about the idea of using cost sensitive convolutional neural network to be able to adapt the data used for training, based on the misclassifications being performed by the model.

2.3 Image-based Pig Body Weight Prediction

As discussed above in the introduction, weight measurement was conventionally largely done manually. Following the evolution of automating this task, it is seen that the accuracy of the weight prediction system depends on the input to the prediction model. To overcome

the challenges posed by manual weighing, with advances in camera technology and image processing techniques there has been a surge in exploration of non-invasive and cost-effective alternatives to manual weighing [43], [35], [48].

These alternatives primarily involve weight prediction systems utilizing computer vision, image processing, and machine learning [2], [29].

2.3.1 Digital Image Processing for Weight Prediction

Digital image analysis has become a prevalent method for predicting animal weight. In this approach, various body dimensions such as length, width, and area are measured and correlated with the animal's weight. For example, [40] used digital image analysis to estimate the weight of pigs, and the experimental results showed that the back area of pigs correlated best with weight, which is possible to be used to effectively determine pig weight. Although, this processing has shown great results, it is not without its drawbacks, such that it requires animals to be in a predefined, relatively straight posture for accurate dimension measurement. If the animal is in a bent posture, estimating body length or width becomes challenging.

Furthermore, initial studies utilized RGB cameras to capture 2D images of pigs in pens. Using these 2D images and image processing libraries, researchers identified the morphological attributes like the length and width.

Subsequent studies started using Deep Learning methods to extract features and train model on these features and get the estimated weight result. Some studies[8, 13], have utilized artificial neural networks on RGB images to estimate the weight. These models used fully connected convolutional layers

But these images lacked precision in estimating pig body volume or parameters relating to animal curvature information like body girth, volume and height [44]. Subsequent research

shifted to using depth cameras, such as Kinect, Intel Realsense, Asus Xtion, to obtain more accurate depth information for calculating pig body volume [14].

2.3.2 Depth Images Processing for Weight Prediction

Over the last couple of decades, to overcome the challenges posed by digital image analysis, there has been an increased use of 3D data to extract the volume feature from the data. 3D data such as depth images and point clouds are being used heavily in this application domain. Pigs height are estimated using the depth information in the images. Digital image processing techniques are commonly employed to extract biometric features such as length, width, height, and volume from depth images, with studies demonstrating a strong correlation between pig volume and weight , [24], [34], [50].

However, these methods necessitate manual extraction of features, which is inadequate as these features are limited, and even slight changes in the pig's posture can affect all parameters. Moreover, this approach usually involves meticulous image processing, including pre-processing, quality control, and ensuring optimal pig body posture and lighting conditions. This process can be time-consuming due to the dataset's size and the number of images.

2.3.3 Point Cloud Data Processing for Weight Prediction

Furthermore, 3D point cloud data is seen to provide more detailed information about the pig body height and girth. Studies [36, 53] have shown that when deep learning models using attention-based networks tends to give promising results.

Liu et al. [36] proposes an intriguing idea with collecting 3D point cloud data using a depth

camera. The weight prediction model uses a multi-modal architecture which is the point cloud data and the corresponding RGB data. This inspired us to use the multi-input CNN models. However, we use derived features from the depth images. This is to ensure that our models can work only with depth images.

Chapter 3

Behavior Classification

This section discusses the first project covered as part of this thesis, 'Sensor-based Behavior Classification'. Our main aim here is to be able to correctly distinguish different pig behaviors based on accelerometer and gyroscope data. Secondly, we also focus on the problem of 'data leakage' as a common oversight in multiple time series analysis tasks. We propose strategic data preprocessing techniques to mitigate the problem of data leakage and improve the performance of models. The following sections are organized into data acquisition, experimental setup, modeling and results and discussions.

3.1 Data Acquisition

We collected time-series sensor data using accelerometer and gyroscope attached to the pig's ears. We will talk in detail about the sensors and the data collection process in the following sections.

3.1.1 Sensors

The study utilized the MetaMotionC (MMC) sensor from MBIENTAB to measure linear acceleration and angular velocity across three dimensions, as depicted in Figure 3.1. The MMC is a wearable device that provides real-time, continuous monitoring of motion and environ-

mental data, with a 50 Hz sampling rate. Its inertial measurement unit captures 3-axis acceleration and angular velocity. More information on measurement ranges and hardware specifications is available on the MBIENTAB website (<https://mbientlab.com/documents/MetaMotionC-PS2.pdf>).



Figure 3.1: The MMC circuit board (left and center), a hand holding a coin next to a cell phone (right). The pictures are courtesy of MBIENTAB (<https://mbientlab.com>).

MMC’s compact design, featuring ultra-low-power capabilities, high-quality sensors, and a coin cell battery, makes it well-suited for use in pig ear tag sensors. The device integrates an ARM Cortex M4F processor, a 9-DoF IMU (Inertial Measurement Unit), a high-precision altimeter, and wireless communication at 33 kb/s on the 2.4GHz frequency, making it ideal for activity classification tasks.

The sensor measures 7/8 inch in diameter and 1/4 inch in thickness, while the pink case measures 1.12 inch in length, as shown in Figure 3.2. Each data point includes measurements of acceleration and angular velocity across three axes, a timestamp, and a counter value, which serve to verify data validity.

3.1.2 Sensor Placement & Camera Setup

This sensor was strategically placed on the ears of pigs, using the sensor case as shown in Figure 3.2 To enhance the sensor data, we positioned a camera on the barn’s roof. This

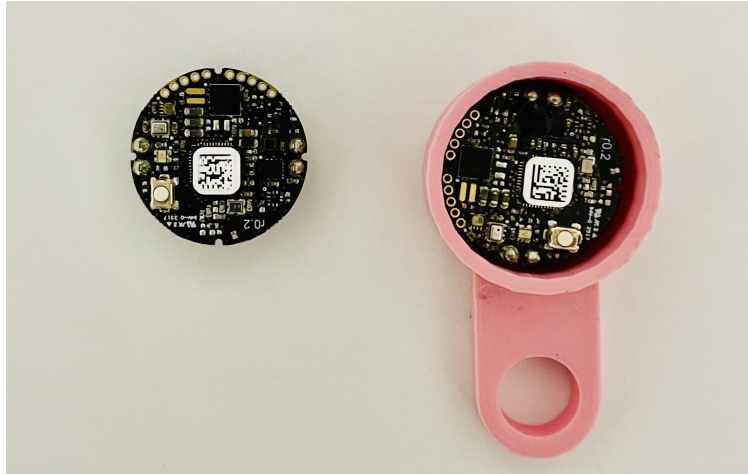


Figure 3.2: The coin sensor and the sensor case used in the study

camera recorded the pigs' movements and played a key role in helping us manually annotate the motion data gathered by the ear-tag sensor. We used an RGB camera to record videos at a rate of thirty frames per second. These video recordings served as a reference for validating data annotations.

3.1.3 Data Collection

A notable aspect of our study is the dataset we gathered and annotated. Over the course of 24 days in the fall of 2022, we observed two pigs, resulting in a robust dataset that covered a total of one hundred hours. Data was collected using both the MetaMotionC sensor and an RGB camera. The pigs, differing in size, shared a pigpen that was spacious enough for them to move freely during data collection. We ensured the pigs had continuous access to food, water, and toys. One evaluator observed the pigs' behaviors via a closed-circuit television from a nearby room, intervening if a pig attempted to tamper with the sensor node. Data was collected in two separate periods each day to maximize data variety. These periods were coordinated with the farm manager's schedule.

3.1.4 Data Labeling

After collecting raw data using the sensor and camera, we carefully reviewed one hundred hours of data to manually annotate it with SensiML Data Capture Lab. This tool was essential for labeling different events within the sensor data, offering easy-to-use graphing tools and a media player to sync video and audio files with the sensor data. Our annotated dataset covered a period of forty hours and included data from twelve distinct pig activities. Table 3.1 outlines the distribution of the various activities observed in the data.

Despite the richness of this curated dataset, we faced challenges such as noise and outliers due to sensor failures, transmission errors, and battery issues. This highlighted the importance of data preprocessing to improve prediction performance, which we will discuss further in the next section.

The carefully curated dataset is a key part of our research, allowing us to investigate and propose new methods for recognizing pig activities.

Table 3.1: Percentage of Six Major Behaviors

Class	Proportion (%)
Drinking	3.53
Eating	44.6
Interacting With Each Other	3.73
Laying	24.85
Standing	8.68
Walking	14.58

3.2 Experimental Setup

3.2.1 Data Preprocessing

To maintain the quality and reliability of the collected sensor data, we carried out essential preprocessing steps, including outlier detection and standardization. We began the process by identifying and correcting erroneous data points in the sensor data through an outlier detection method. This step is vital because outliers can introduce noise and inaccuracies, potentially affecting model performance. We used the robust interquartile range method, which calculates the difference between the first and third quartiles (the interquartile range) of the data. Any data point outside this range is flagged as an outlier. Once detected, we used linear interpolation to replace the outliers with appropriate values, keeping the dataset consistent over time.

Additionally, we standardized each dimension of the accelerometer and gyroscope sensors for uniformity and consistency across all data values. This standardization process involves scaling the data to have a mean of zero and a standard deviation of one. It ensures that each feature is on the same scale and has an equal influence on the model, also helping to minimize the impact of outliers on the model's performance.

3.2.2 Feature Extraction

In the field of machine learning, features refer to raw data converted into numerical representations that retain the original data distribution [30, 42]. In our study, we used time domain features such as mean, standard deviation, minimum, maximum, median, and interquartile range from accelerometer and gyroscope readings along the x, y, and z axes. These features are commonly used in activity recognition research because they are simple to compute and

require low computational power. They also offer valuable insights into pigs' movements, including range of motion, speed, and stability [19].

We also transformed the time domain dataset into a frequency domain dataset using the Fast Fourier Transform method. Features such as sample frequencies, phase, maximum amplitude frequency, power spectral density, power spectral entropy, and weighted frequencies were included. These frequency domain features are beneficial for detecting patterns in the data and capturing signal frequency content.

By combining both time and frequency domain features, we created a comprehensive 'feature domain dataset' for further analysis and modeling. This approach captures different facets of the data, providing a complete representation of pigs' movement patterns, which is crucial for animal behavior research.

3.2.3 Proposed Time Window Segmentation Method

Windowing is the process of segmenting continuous sensor data into smaller time intervals, or windows, for classification purposes. Each window typically spans a few seconds and is labeled with the activity occurring during that time [3].

The standard windowing method, known as "class-based windowing," organizes data according to activity labels before creating windows. This method assumes access to activity labels during both training and testing phases. However, this introduces bias since the system uses activity labels during training that it wouldn't have during real-world testing, potentially leading to high performance on training data but poor generalization to new, unfamiliar data.

To address this bias, we propose "non-class-based windowing," which divides sensor data into windows based on the natural time sequence of the recorded dataset, without sorting

by activity labels. This method more accurately mirrors real-world scenarios where the system does not have access to activity labels during testing. It ensures robustness and precise activity recognition, even in scenarios with interleaved activities that differ from the training data.

Finally, we removed classes with low-frequency occurrences, resulting in a dataset with six classes, each containing a similar number of samples. This dataset is known as the 'time domain dataset'. This dataset had the dimension of [55665,768]. While training models which required 1D data, we used this dataframe, where it had 55665 timesteps and 768 features for the model to train on.

For 2D data analysis, we stack the data points in a way such that we separate out each parameter with respect to 6 axes (acc_x, acc_y, acc_z, gyro_x, gyro_y, gyro_z). This converts our dataframe dimension to [55665, 128, 6]. Here we are stacking each axis against each other to make a 2D dataframe. Additionally, for the labels we expand them to form a binary matrix (one-hot-encoded vectors) for training purposes. The dimensions of our labels was then [55665, 6].

Our main concern here was to show that our 2D stacking makes sense spatially. For this, we conducted a principal component analysis (PCA) on our dataset. PCA is a statistical technique used to reduce the dimensionality of high-dimensional data while preserving most of the variability present in the data. We did a PCA on all the three axes data of accelerometer.

The graph in Figure 3.3 reveals the presence of two principal component axes. The x-axis represents the direction of maximum variance in the data, while the y-axis represents the direction of the second maximum variance. The proximity of points in the graph indicates the degree of similarity or dissimilarity between samples. We observe that the points cluster closely together, suggesting a high degree of similarity and relation among them.

A similar trend is evident in the analysis of gyroscope data, as depicted in Figure 3.4. Once again, points cluster along the principal components, although some dispersion is still noticeable in both graphs. We posit that a 2D CNN model could effectively capture these complexities in the data distribution, allowing for a more nuanced understanding of the underlying patterns.

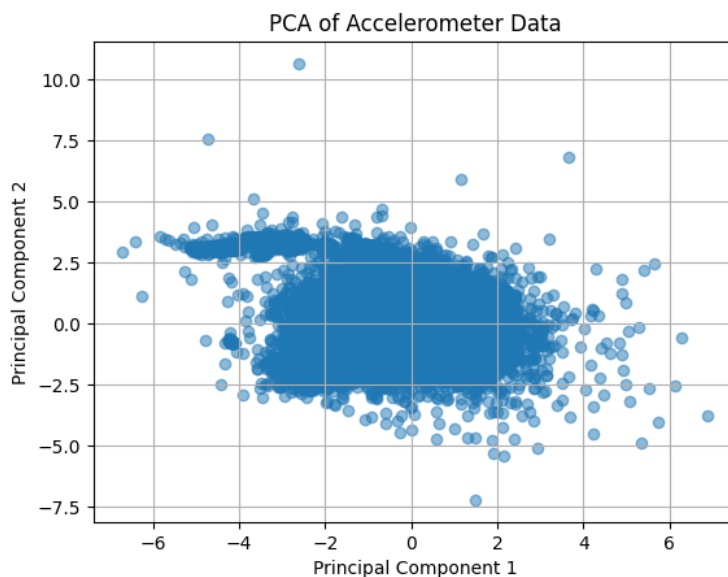


Figure 3.3: Principal Component Analysis of Accelerometer data along three axes. This graph shows clustering of data points together showing correlation between points.

3.2.4 Proposed Chronological Train/test split

Random sampling is a widely used method for creating training and validation sets from a dataset, as it aims to provide an unbiased representation of the population. However, this approach may not be suitable for time series data like ours because it does not maintain the temporal consistency of the data distribution. Time series data exhibit sequential dependence, where the value of a data point at a particular time is often influenced by preceding values. This characteristic of time series data leads to correlations between data points,

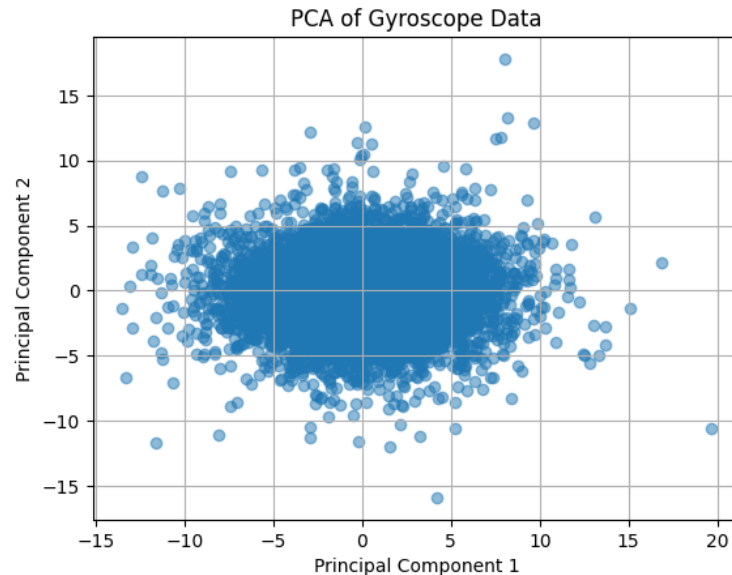


Figure 3.4: Principal Component Analysis of Gyroscope data along three axes. This graph shows clustering of data points together showing correlation between points.

which violates random sampling's assumption of independence. As a result, using random sampling for time series data can lead to information leakage between training and validation sets.

Information leakage occurs when the model is exposed to future samples from the validation set during training, potentially causing overfitting. In the context of pig behavior analysis, overfitting can result in unreliable predictions, negatively affecting the system's performance.

To address this, our study implemented "chronological time sampling" to separate the dataset into training and test sets based on the order in which the data was recorded, thus avoiding information leakage between these sets. Chronological time sampling segments the dataset sequentially according to recording time.

In our analysis, the first quarter of the dataset served as the test set, while the remaining three-quarters formed the training set. In other words, we used a 75% train and 25% test split in our analysis. The main intuition behind this approach is that it preserves temporal

dependencies in the data and ensures that both the test and train sets accurately represent the data distribution.

We recognize there may be concerns around chronological sampling, as there is a possibility that the training data might not incorporate all the activities. Another reason supporting random sampling was that, in large-scale data, the age of the pigs could influence their behaviors. As pigs grow older, they exhibit different behaviors, which might not be captured in the training data, since it is only looking at a specific period in the dataset.

The following reasons reinforce our intuition behind using chronological sampling methodology. First, we collected data from two pigs over a period of 24 days, with both pigs belonging to different age groups. Given the limited scope of our data collection, we can assure that the training set covers all activity labels and captures age-related changes observed in the behaviors of the two pigs. We believe that since our data collection period and the sample size were scoped, our approach of chronological sampling makes sense in our study. The graph showing the distribution of labels in our train data is shown in Figure 3.5.

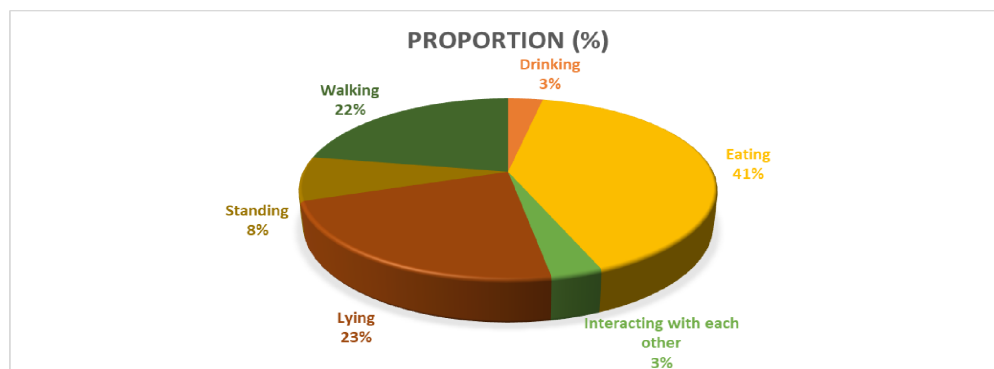


Figure 3.5: Distribution of activity labels in the train dataset. This shows that the train dataset has a good balance of all activity labels.

Second, an additional objective in our research question is to alleviate data leakage in time series analysis. And as identified above, random sampling causes data leakage. Thus, using our proposed method of chronological sampling we directly tackle that issue.

3.3 Models

Following the primary objective of this study, we wanted to compare our proposed preprocessing techniques to the commonly used methods. Additionally, based on the type of models we used, we wanted to show the comparative performance between ML and DL models. For evaluating the performance of the models, we primarily used classification accuracy.

3.3.1 Machine Learning Models

To benchmark traditional machine learning methods, we employed decision trees, random forest, and K-Nearest Neighbors.

Decision Trees (DT) model uses a recursive process to divide input data into smaller subsets. At each node, the algorithm chooses a feature that best separates the data into distinct classes and continues until it reaches a leaf node, where it assigns a class label. We opted for Gini Impurity as the measure of split quality in Decision Trees.

$$Gini(p) = 1 - \sum_{i=1}^C p_i^2 \quad (3.1)$$

At inference time, the decision tree is traversed from the root node down to a leaf node. At each internal node, the algorithm evaluates the value of a feature and follows the corresponding branch based on whether the feature value satisfies a certain condition. Once a leaf node is reached, the data point is assigned the class label associated with that leaf node. By learning decision rules from labeled accelerometer and gyroscope data, decision trees can accurately classify pig activities such as walking, running, feeding, or resting.

Random Forest (RF) is an ensemble learning algorithm that generates multiple decision

trees and merges their outputs. It uses bootstrapping to randomly sample the training data and trains a decision tree on each subset. Additionally, at each node of the decision tree, a random subset of features is considered for splitting, rather than using all features. The final prediction is an average of all the decision trees' outcomes.

It is noteworthy that RF algorithm evaluates the importance of each feature based on how much it contributes to the overall accuracy of the model. Features that are less informative or redundant are given lower importance and are less likely to be selected for splitting nodes in the decision trees. Due to this capability of handling high dimensional data, RF is known to give good performance in tasks such as multi-class classification.

The **K-Nearest Neighbors (KNN)** model calculates the distance between samples and identifies the K-nearest samples to the test sample. The class of the test sample is then determined by the majority class of its K-nearest neighbors.

In our experiment, we set k to 3 after hyperparameter tuning. It is known that, KNN algorithm does not require explicit training. When a new, unlabeled data point is presented for classification, KNN identifies the K nearest neighbors to the query point in the feature space. The "nearest" neighbors are determined based on a distance metric, which was euclidean distance in our case. Once the K nearest neighbors are identified, the majority class among these neighbors is assigned to the query point. This majority class serves as the predicted class label for the query point.

3.3.2 Deep Learning Models

For deep learning approaches, we explored different variants of convolutional neural networks and recurrent neural networks. We designed and tested the following models in our study:

- 1-D and 2-D CNN
- 1-D and 2-D CNN with Residual blocks
- Long Short Term Memory (LSTM)
- CNN + LSTM

Since our data is 1-D in nature, it was intuitive for us to try **1-D convolutional neural network (CNN)**. The model includes convolutional layers that extract features from the input data and pooling layers that decrease the dimensionality of the output. We also used dropout layers and dropping up to 50% of the layers to reduce overfitting.

The **2-D CNN** model, on the other hand, is designed to process two-dimensional data like images. For this we stacked the features in a way to form 2-D dataset and used 2D convolutional layers to design the model.

The **1-D Residual CNN** model is a variation of the 1-D CNN that incorporates residual connections to improve the flow of information through the network, enabling the model to learn more complex features by reusing earlier features. Residual blocks address the problem of vanishing gradients in very deep networks by introducing skip connections, which allow the network to learn residual functions. This architecture enables training of much deeper networks (with over 100 layers) while maintaining good performance. The ResNet-34 architecture is particularly useful for tasks involving high-dimensional data, such as our time series analysis.

Similarly, the **2-D Residual CNN** model is based on the ResNet-34 architecture, which is known for its outstanding performance in classification tasks due to its use of skip connections. More information can be found in [Appendix B](#)

The **long short-term memory (LSTM)** model is a form of recurrent neural network

(RNN) built to handle sequential data such as time series data. The model consists of a series of LSTM units that can learn long-term dependencies in the input data. LSTMs are effective at capturing temporal patterns in time series data like ours. They can also be combined with other models, such as CNNs, to leverage both long-term dependencies and local trend features. To explore this capability, we implemented a hybrid **CNN+LSTM** model, which first uses a convolutional neural network to extract features from the input data and then feeds the features through an LSTM layer to learn long-term dependencies.

3.4 Results & Discussions

In this section, we present the evaluation results of various machine learning and deep learning models using different data preparation methods. The analysis is divided into three parts, each corresponding to a specific comparison, detailed below. It is important to note here that 'Others' essentially refers to class-based segmentation and random train/test split, 'Proposed' refers to the methods used by us, i.e., 'Non-class-based segmentation' and 'Chronological train/test split'.

The first part of the analysis compares the proposed chronological time split method against the random train/test split method (referred as 'Others' in the table), while using the common class-based windowing approach (Table 4.2). The results show that the proposed splitting method significantly improves the performance of all models in both time and feature domains. For example, the 1-D CNN model accuracy increased from 77.5% with the common method to 89.2% using the proposed method in the time domain. Similarly, in the feature domain, the accuracy increased from 78.8% to 86.7%. This trend is consistent across all the models tested, indicating the advantages of the proposed splitting method.

In the second part of the analysis, we evaluate the impact of the proposed non-class-based

Table 3.2: Comparison of classification performances on datasets with proposed train/test split vs other train/test split

Model	Time Domain		Feature Domain	
	Others	Proposed	Others	Proposed
1-D CNN	77.5	89.2	78.8	86.7
1-D Res-CNN	79.2	90.1	79.5	87.5
2-D CNN	78.6	88.4	79.1	87.1
2-D Res-CNN	81.2	90.5	80.0	88.5
LSTM	78.5	87.4	78.2	88.1
CNN+LSTM	77.6	88.1	78.0	86.9
Random Forest	70.1	82.1	80.5	91.1
Decision Tree	56.2	70.8	64.5	87.2
KNN	52.1	61.2	60.3	82.2
Average	72.3	83.1	75.4	87.3

windowing method compared to the class-based windowing method (referred as 'Others' in the table), using the common random splitting method (Table 4.3). The results demonstrate that the proposed windowing method leads to better performance in both the time and feature domains for most of the models. For example, the 2-D Residual CNN model achieved 88.1% accuracy in the time domain with the proposed method, compared to 80.0% using class-based windowing. Similar improvements were observed for other models as well, highlighting the benefits of the non-class-based windowing method.

The third part of the analysis focused on the combined efficiency, comparing the performance of the models using both the commonly used class-based windowing and random split methods (referred as 'Others' in the table) against the proposed windowing and splitting methods (Table 3.4). The results reveal that the combination of the proposed methods further enhances the model's performance in both time and feature domains. Figure 3.6 represents the obtained results. In Figure 3.6a, the 2-D residual network stands out with the highest classification accuracy among the presented methods in the time domain. We conduct a comparative analysis, examining the performance of both data preprocessing methods

Table 3.3: Comparison of classification performances on datasets segmented with proposed windowing method vs segmented with other windowing method

Model	Time Domain		Feature Domain	
	Others	Proposed	Others	Proposed
1-D CNN	77.5	87.1	78.8	86.2
1-D Res-CNN	79.2	87.5	79.5	87.1
2-D CNN	78.6	86.8	79.1	86.4
2-D Res-CNN	80.0	88.1	80.0	87.5
LSTM	78.5	86.8	78.2	85.7
CNN+LSTM	77.6	87.1	78.0	85.9
Random Forest	63.2	80.4	81.1	90.2
Decision Tree	56.1	67.2	64.5	86.8
KNN	52.1	59.2	60.3	78.0
Average	71.4	81.1	75.5	85.9

Table 3.4: Comparison of classification performances on datasets with proposed split and windowing vs other split and windowing

Model	Time Domain		Feature Domain	
	Others	Proposed	Others	Proposed
1-D CNN	77.5	94.4	78.8	93.2
1-D Res-CNN	79.2	94.75	79.5	93.2
2-D CNN	78.6	90.5	79.1	91.8
2-D Res-CNN	80.0	95.6	80.0	93.3
LSTM	78.5	93.0	78.2	93.0
CNN+LSTM	77.6	93.2	78.0	91.5
Random Forest	63.2	85.0	81.2	94.1
Decision Tree	56.1	73.1	64.5	90.0
KNN	52.1	60.0	60.3	85.1
Average	71.4	86.6	75.4	91.7

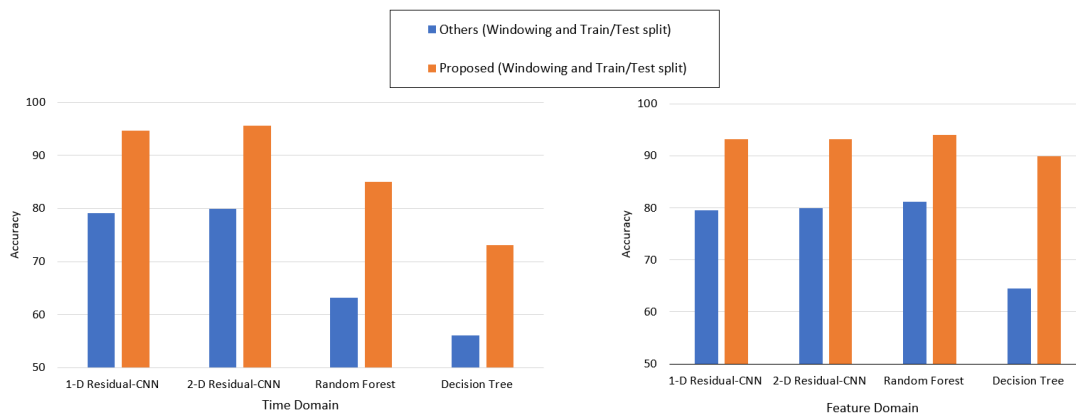


Figure 3.6: ML-based pig activity classification results with different data preparation methods on (a) time domain data and (b) feature domain data.

individually and when utilized in combination. Similarly, in Figure 3.6b, showcases results in the feature domain, with the random forest method demonstrating the highest accuracy. Notably, we observe an improvement in performance when applying our proposed methods even in the feature domain.

For example, the 1-D Residual CNN model achieved an accuracy of 79.2% using the commonly used methods in the time domain, which increased to 94.75% with the proposed methods. A similar trend was observed in the feature domain as well, with accuracy increasing from 79.5% to 93.2%. This improvement is consistent across all tested models, indicating that the proposed methods are effective in improving pig behavior analysis.

The advantage of using DL models is that, these models do not require manual feature engineering. We can use the raw data to train the models and get better results than most ML models. Furthermore, as demonstrated in our results, 'data leakage' is a valid problem in time series analysis which needs to be addressed for getting results truly indicative of the model's capabilities. Ensuring that there is no leakage of data is highly pertinent so that these systems work well in real-world scenarios testing on out of distribution datasets.

Chapter 4

Body Weight Prediction

In this section, we will discuss pig body weight prediction project. The main aim of this project is to estimate the weight of pigs using only depth images. We also present a comparative analysis of using traditional statistical models and modern deep learning models. We will cover the following topics in the coming sections, data acquisition, experimental setup, modeling and results and discussions.

4.1 Data Acquisition

4.1.1 Camera Setup

A depth video acquisition system was set up in a swine facility at Virginia Tech. The camera setup used in this study used an Intel RealSense D435 camera (Intel, Santa Clara, CA, USA), which was strategically positioned above the ceiling pipe, in the center of an indoor testing pen measuring 5×7 ft using a clamp. The camera-to-floor distance was set at 3.6 meters, and camera operation was managed from a laptop computer using the Intel RealSense Viewer interface.

4.1.2 Data Collection

A total of 4 pigs were used to collect depth video data over 27 days from September to November 2021 with a resolution of 848×480 . The pigs were a crossbreed of Yorkshire and Large White and they entered the trial at 5 wk postweaning. A digital scale (Arlyn Scales, New York, NY, USA) located in a pen adjacent to the image-recording pen was used to manually collect pig body weight records. Body weight records from the scale and image data for each pig were collected daily in the afternoon. Table 4.1 shows the ground truth data distribution of the scale-based body weight. These statistical measures provide insight into the distribution and central tendency of the weight data, offering a comprehensive understanding of the pig population under study. The data has a considerable variance range in terms of weight, from 16.5 kg to 56.6 kg.

Table 4.1: Basic Statistics of Ground Truth Weight Values

Pig ID	Mean (kg)	Median (kg)	Minimum (kg)	Maximum (kg)
A	30.38	29.4	16.5	55.5
B	35.13	32.9	16.5	56.6
C	25.27	25	19.3	31.8
D	23.14	22.4	17.4	30.9

4.1.3 Data Labeling

The dataset used in this study consisted of ninety weight sample points. We wanted to label all the images with their corresponding weight sample points to aid in the training process. To accomplish this, we mapped the file path of each image to its associated weight sample point, organizing the data in a tabular format. Subsequently, we consolidated these mappings into a master data file to facilitate streamlined access for analysis.

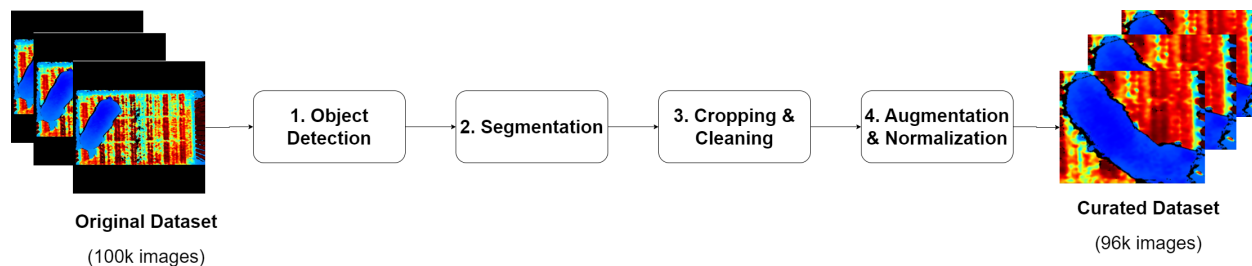


Figure 4.1: An overview of the image preprocessing pipeline. This pipeline utilizes segmentation masks to implement most of the tasks in the pipeline. These methods help achieve a cleaner dataset that minimizes extraneous noise in the images.

4.2 Experimental Setup

4.2.1 Data Preprocessing

The preprocessing pipeline used in this study is outlined in Figure 4.1. After data collection, we amassed a total of 100k images with a single pig in each image. Our preprocessing procedures included object detection, image segmentation, cropping, dataset cleaning, data augmentation and normalization, each step is explained in detail in this section.

4.2.2 Object Detection

As a precursor to the segmentation step, we had to get the bounding boxes around pig’s body. For this, we trained YOLO v7 on manually annotated images, and then inferred from the model to get the bounding box coordinates for the remaining set of images. Manual annotation involved annotating a subset of 500 images using the lblImg tool [45], where bounding boxes were drawn around the pigs.

These annotated images were then used to train the YOLO v7 object detection model, which specializes in real-time object detection and localization. The trained YOLO v7 model was

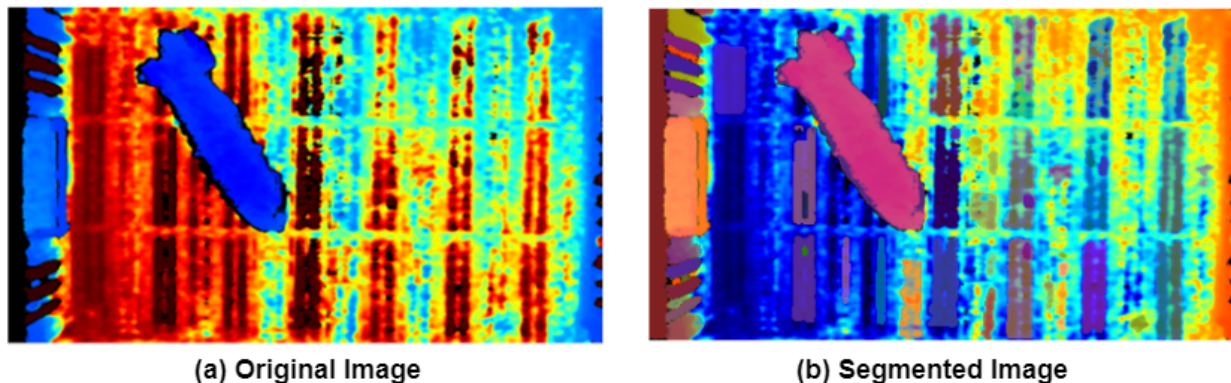


Figure 4.2: Segment Anything Model automatic mask generation. The model takes as input both image and box prompts for generating segmentation masks.

applied to the entire dataset of pig depth images, generating bounding box coordinates for each pig instance. This automated process efficiently identified the regions of interest containing the pigs. Subsequently, we used these bounding boxes as prompts for getting the segmentation masks of the depth images.

4.2.3 Segmentation

For our segmentation efforts, we chose to leverage SAM [33]. SAM is a pre-trained segmentation system capable of identifying and segmenting objects in images without need for additional training. Due to its accessibility and high performance, implementing this model on our dataset was a swift process. First, we started with SAM’s automatic mask generation on our dataset to assess its performance, as shown in Figure 4.2. After validating the results of the mask generation, we were confident in SAM ability to generate segmentation masks for the subsequent stages of the experiment. Within SAM, the `SamPredictor` [33] class simplifies user interaction with the model. Users can specify an image using the `set_image` method, which computes the required image embeddings. Subsequently, the `predict` method efficiently generates masks based on the prompts provided. SAM can accept various prompt



Figure 4.3: Segment Anything Model (SAM) segmentation masks. SAM does not require any additional training to provide the output on custom dataset images.

types, including bounding box and masks from previous predictions.

In our implementation, we opted for the bounding box approach. As discussed in the above sub section, we used the bounding box coordinates extracted using YOLOv7 as input to the SamPredictor class instance, yielding results in minutes. The results, as depicted in Figure 4.3, showcased SAM generating three different segmentation masks for each input image by default.

The results were highly satisfactory, with the leftmost image in Figure 4.3 being selected as the best segmentation mask. Subsequently, we used normalized masks for further analysis. More information can be found in the Appendix A

4.2.4 Cropping

In the cropping phase 4.4, we created square bounding boxes of a fixed size around the pig's body while minimizing background clutter. This was critical to provide concise, contextually relevant images for model training. We utilized SAM-generated segmentation masks to delineate the pig's body boundaries. Since the pig bodies vary in size and shape throughout the dataset due to differences in their weights, we decided on a fixed size of 720x720 pixels for the square bounding box that would accommodate the largest pig body in the dataset.

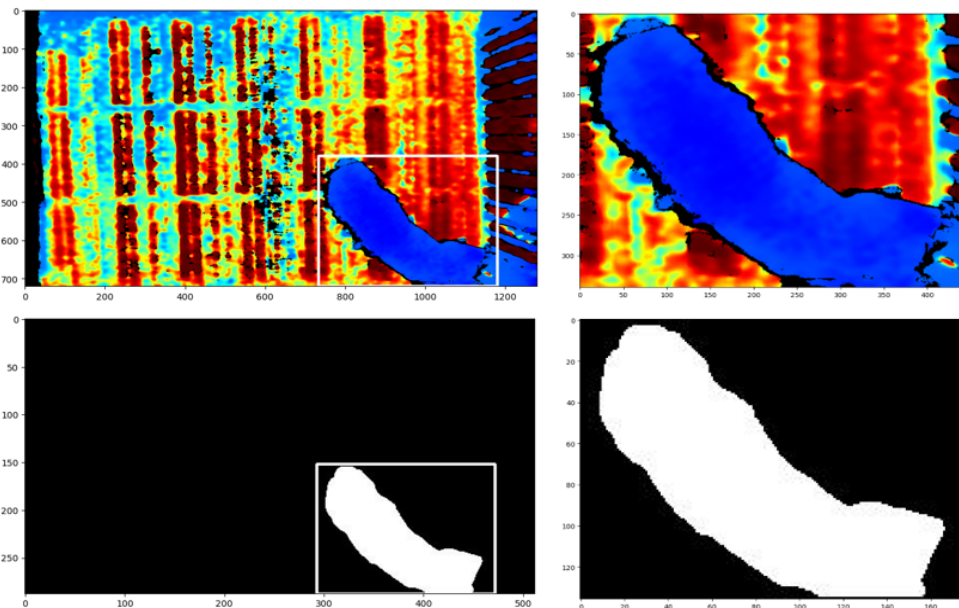


Figure 4.4: Cropping pigs' body using the bounding box coordinates generated by YOLO-V7 model.

We then centered this fixed-size square bounding box around the center of the pig's body, ensuring that the entire pig was included within the square frame. In cases where the pig's body was smaller than the fixed-size square, the remaining space was filled with the surrounding background. Once we acquired the coordinates for the fixed-size square bounding boxes for all the images, we cropped each image accordingly. Following cropping, we resized all the cropped images to the size of 256x256 pixels.

4.2.5 Cleaning

Cleaning 4.5 involved the removal of images where the pig's body was partially visible or distorted, typically due to truncation at the edges of the image. Ensuring the elimination of distorted images was crucial to prevent misguidance during model training, as the task relies heavily on the pig's body shape. For this task, we utilized segmentation masks generated by the Segment Anything Model (SAM) to detect continuous stretches of the pig's body at the

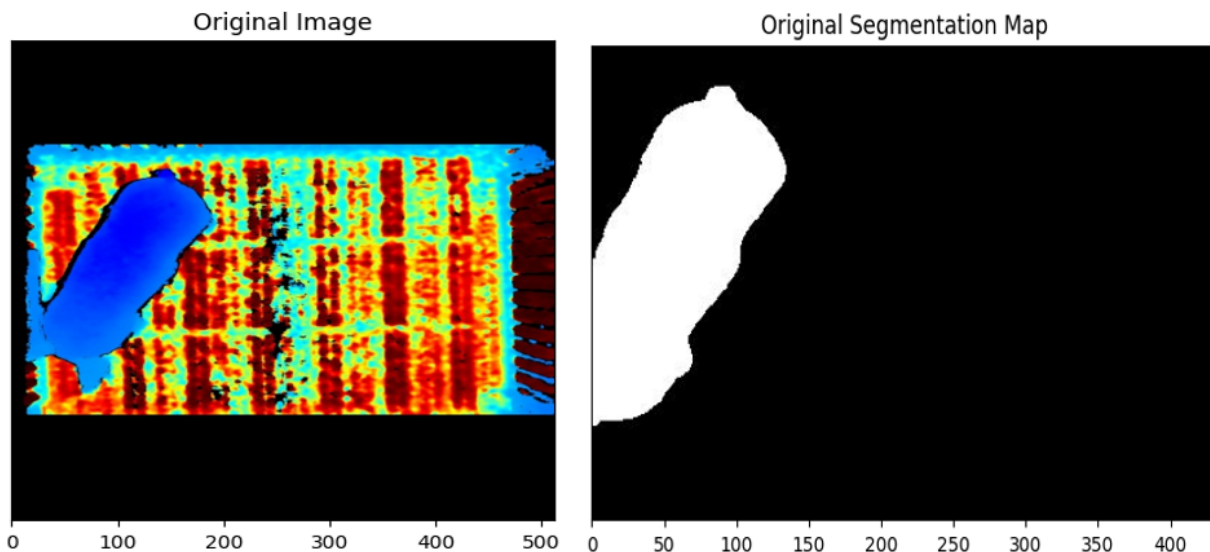


Figure 4.5: Identifying pigs on edges using the segmentation mask generated by SAM. This example clearly shows the pig body to be distorted at the edges.

edges of the images, as shown in Figure 4.3. Subsequently, we implemented a pixel check algorithm to validate the presence of the pig’s body in the segmentation mask. Specifically, we examined whether a range of pixels at the top, bottom, left, and right edges of the image appeared white, indicative of the pig’s body. Images identified as "pig_on_edge" were flagged and subsequently updated in the main dataset.

4.2.6 Data Augmentation & Normalization

Regarding data augmentation, we capitalize on the abundance of images capturing identical postures due to the high sampling rate of the camera. Following preprocessing, systematic transformations such as flipping and mirroring are applied to augment the training dataset. This augmentation procedure enhances the model’s ability to generalize while mitigating the risk of overfitting. Notably, scaling transformations are excluded from this process due to the observed positive correlation between the volume of the depth camera and the pig’s

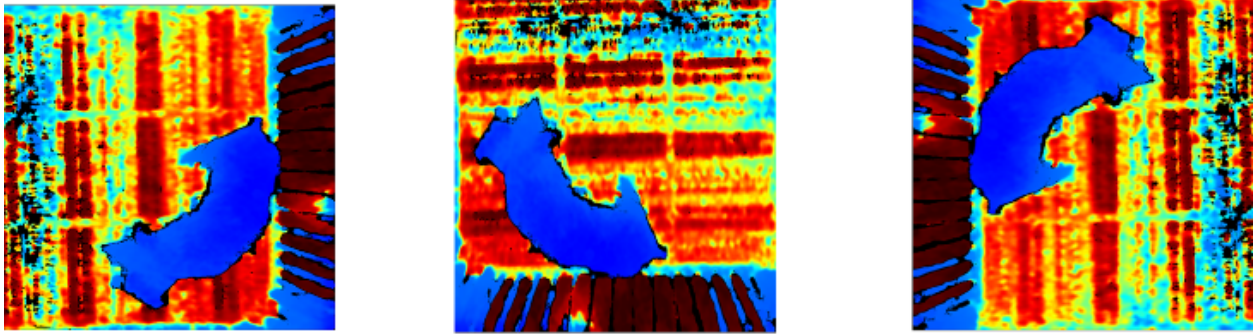


Figure 4.6: Data Augmentation techniques. On the original image (left), we applied rotate by 90 degrees (center) and flipping (right) operations to add more variations in the existing dataset.

weight.

Normalization plays a pivotal role in ensuring that each image pixel is on the same scale and contributes equally to the model. Here, we ensured that all pixel intensities fell within the range of $[0,1]$. After completing these preprocessing steps, we were left with a total of 96k images for training and validation purposes.

4.2.7 Cross-Validation

We applied a one-pig out cross-validation technique, where one pig was designated as the test set while the remaining three pigs constituted the training set. This approach helped to alleviate overfitting and enhance the overall generalizability of the model. Furthermore, this method also helps to address the issue of data leakage in image analysis. Since most of the pigs share common visual features, mixing of pig ids while training can lead to overfitting during training and validation.

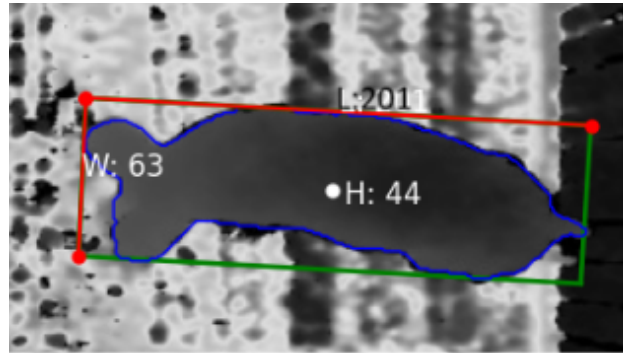


Figure 4.7: Extracting biometric features such as length (L), width (W), and height (H) of the pig contour in pixel space. The point denoting the height is the center of the pig's body. These features are estimated using depth images.

4.3 Models

In this section, we discuss the models we developed for body weight prediction. I, personally contributed to designing the Residual Network architecture which is explained in detail in the below section.

4.3.1 Statistical Models

The weight prediction approach using depth images starts with accurately segmenting pigs with the Segment Anything Model (SAM). After segmentation, we use thresholding, contour detection, and bounding boxes to isolate the pig in the images. First, the depth image is converted from RGB to grayscale, where the grayscale intensity values represent the distance of each pixel from the camera, with higher intensities indicating further distances. We then apply a threshold value of 128 to create a binary image, which is used to identify the contour and encase the pig's body in a bounding box.

The bounding box is crucial for accurately measuring physical dimensions such as width, length, and height (Figure 4.7). To calculate the pig's height, we use the grayscale depth

image and bounding box coordinates. We determine the center pixel of the bounding box by averaging the coordinates and rounding them to the nearest integer. The grayscale intensity value at this center pixel is then considered the height of the pig.

Additionally, the pig's volume is calculated by summing the pixel heights within the pig's contour in 3D depth images, capturing the pig's three-dimensional shape. These features are crucial for representing the pigs' overall size and shape, directly correlating to their weight.

The extracted features form the foundation for building statistical models. Models such as Linear Regression (LR), Random Forest (RF), and Support Vector Machine (SVM) are trained on these features to explore different statistical approaches for accurate weight prediction.

Linear Regression provides a simple model of weight as a linear function of dimensions, offering an initial understanding of the relationship between physical attributes and weight. Random Forest Regressor, however, handles more complex and nonlinear relationships in the data by combining predictions from multiple decision trees, each trained on different data subsets, enhancing prediction accuracy and robustness. Lastly, Support Vector Machine (SVM) constructs a hyperplane in a high-dimensional space, providing precision and effectively handling high-dimensional feature sets by maximizing the margin between support vectors and the hyperplane.

4.3.2 Deep Learning Models

For deep learning models, we designed and inferred the following models:

- Convolutional Neural Networks (CNNs)
- Multi-input CNNs

- ResNets
- XceptionNet

Each of these models is described in sections below.

Convolutional Neural Networks and Multi-Input Models

Our method begins with a Convolutional Neural Network (2D-CNN) tailored for processing depth images to predict pig weight. This model is designed to handle cropped depth images converted to grayscale, then normalized and resized to 256x256 resolution to maintain consistency across the dataset, as outlined in the preprocessing section. A series of convolutional layers capture spatial hierarchies of features like edges, textures, and shapes, essential for understanding the pigs' physical dimensions. This setup is visualized as a single branch on the multi-input CNN model shown in Figure 4.9. Subsequent pooling layers reduce dimensionality and focus the model on the most significant features. Fully connected layers then correlate these refined feature representations with pig weights, establishing a baseline for weight prediction accuracy.

Building on the base 2D-CNN model, the multi-input CNN framework introduces the integration of segmentation maps and multiscale representation analysis (MRA) for enhanced feature extraction. The first multi-input model combines CNNs with segmentation maps, allowing the model to process depth images with segmentation maps that emphasize the pig against its background. This integration improves the model's focus and accuracy in feature extraction, leading to more precise body weight prediction results. The second multi-input model incorporates CNNs with MRA (Figure 4.8), providing a unique approach to capture and analyze image textures at various scales. By utilizing depth information and texture insights from MRA, this model gains a comprehensive understanding of the pig's physical

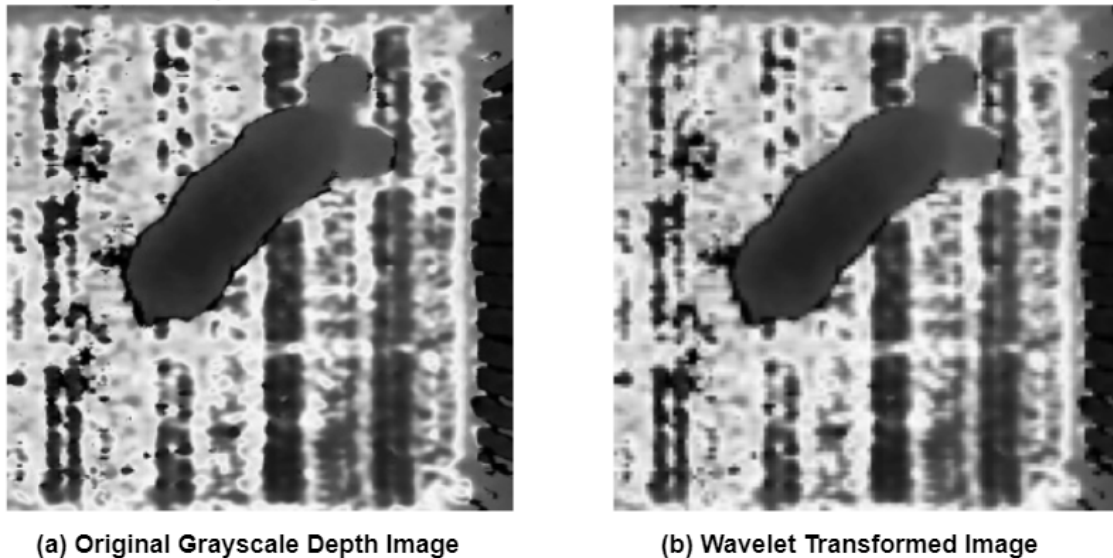


Figure 4.8: Example of an original depth image and its multiresolution analysis (MRA) decomposition using wavelet transform. MRA of an image helps capture both fine details and coarse features simultaneously which helps to analyze the image.

attributes, enabling more accurate weight prediction.

Residual Networks

Residual Networks (ResNet) represent a significant advancement in CNNs, specifically addressing issues such as vanishing/exploding gradient and optimization problems that arise when increasing the depth of CNNs [27]. Traditional deep neural networks are challenged when gradients exponentially increase or decrease as they propagate through many layers during training. This phenomenon, known as the exploding or vanishing gradient problem, can hinder learning.

In ResNets 4.10, the fundamental concept revolves around enabling the learning of identity mappings for its building blocks when necessary. This is accomplished through the incorporation of shortcut or skip connections. These skip connections facilitate the direct addition of outputs from previous layers, effectively by passing certain layers in the network. By

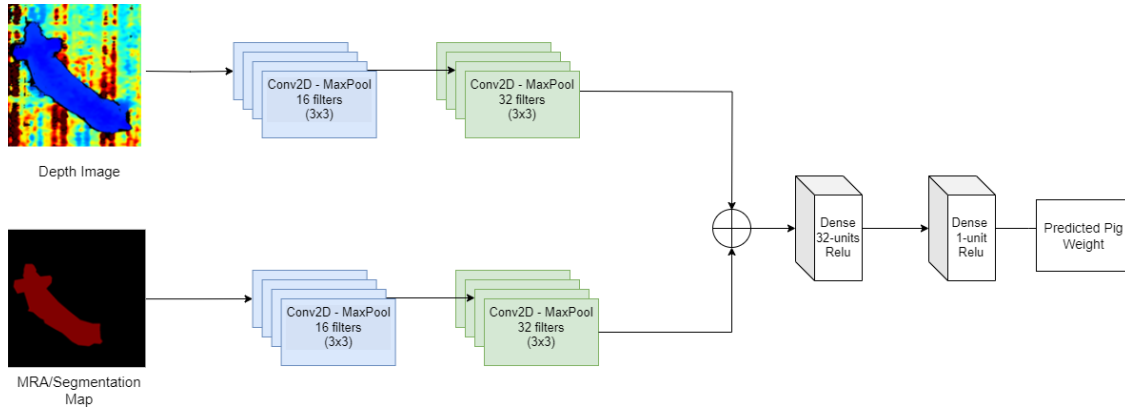


Figure 4.9: Multi-input 2D-CNN architecture. Here, the integration of two modalities is achieved by fusing their features internally. The parallel processing helps to improve feature extraction for better accuracy in body weight prediction.

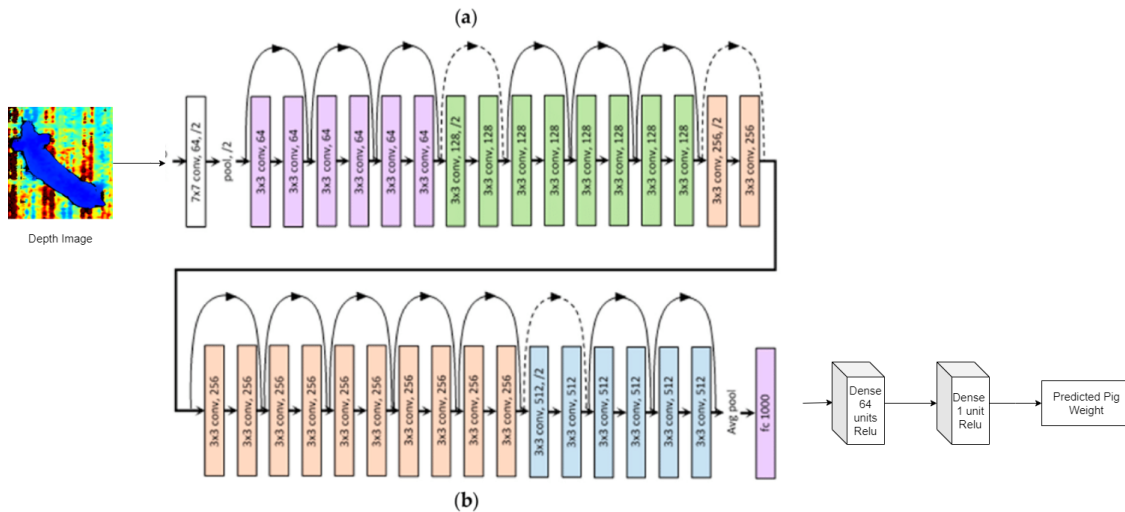


Figure 4.10: ResNet-34 architecture. The skip connections used in the model help propagate the gradient value from one layer to the other, thus reducing the effect of vanishing gradient problem.

doing so, ResNet ensures the continuity of gradient flow during training, thereby mitigating the issues associated with vanishing or exploding gradients.

In our experiment, Residual Networks were considered as baseline deep learning models because they are known to work well with image data. Concurrently, we integrated the ResNet-34 model, which consists of 34 layers with skip connections. To facilitate the training process, we maintained batch size of 64, this helped to strike a balance between the training speed and results accuracy. Furthermore, we used the 'Adam optimizer' to optimize the learning process efficiently.

XceptionNet

The Xception model, as shown in 4.11 which extends the Inception design, introduces two key concepts: Group Convolution and Separable Convolution [15]. Group Convolution splits segmentation channels into multiple groups, with convolution operations occurring within each group. Separable Convolution employs a 1x1 convolution (point-wise convolution) in the middle of a convolutional layer, effectively reducing the number of parameters while enhancing channel count and feature representation capabilities. The combination of these elements leads to the use of depth-wise separable convolution in Xception. By streamlining the Inception module on which Xception is based, we focus on retraining only the branches with 3x3 convolutions and concatenating all 1x1 convolutions. This step increases the number of branches with 3x3 convolutions to match the output channel count of 1x1 convolutions, creating the core building block of Xception. Additionally, the model includes residual connections akin to ResNet, delivering exceptional performance in various image-related tasks.

In our experiments, we fine-tuned Xception using pre-trained parameters from the ImageNet

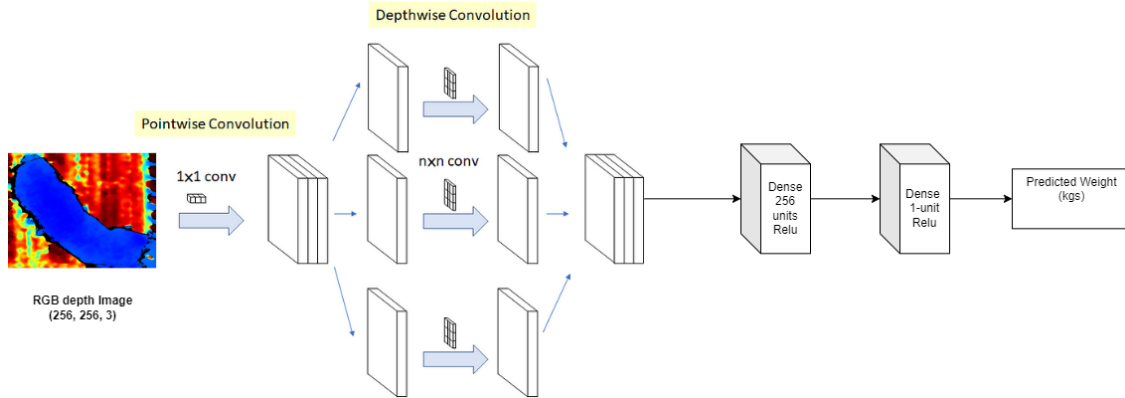


Figure 4.11: XceptionNet architecture used in our design. We see that this model utilizes separable convolution to efficiently train and pass weight in deep layers.

dataset [16]. We enhanced the model architecture by adding an average pooling layer, two dense layers, and setting the batch size to 64. To avoid overfitting, we employed regularization techniques such as L2 regularization (weight decay) and a dropout rate of 0.2. We carefully froze specific layers or parameters to prevent them from being updated during fine-tuning. Specifically, we froze all layers initially and trained our model, then froze the lower-level parameters while fine-tuning the parameters in the last 20 layers. We retrained the model using a lower learning rate of 0.0001. We implemented 4-fold cross-validation, with each pig serving as a testing dataset.

4.4 Results & Discussions

In this section, we present the performance results of our frameworks and models based on mean absolute error (MAE) and mean absolute percentage error (MAPE) as the metrics to evaluate the accuracy of pig weight prediction. Lower values of these metrics indicate better model performance.

MAE measures the average absolute magnitude between the observed values and the pre-

dicted values, as shown in equation (4.1), using a regression model.

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \bar{Y}_i| \quad (4.1)$$

On the other hand, MAPE measures the absolute percentage difference between the observed and predicted values, as represented in equation (4.2) below.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{Y_i - \bar{Y}_i}{\bar{Y}_i} \right| \quad (4.2)$$

Here n represents the total number of observations. Y_i and \bar{Y}_i represent the observed and predicted values, respectively. To ensure robust generalization performance and prevent data leakage, we employed the 'leave-one-pig-out' method for cross-validation. This involved selecting one pig as the test set and the remaining three pigs as the training set in each iteration.

Table 4.2: Weight Prediction Results of Statistical Methods

Model	Train		Test	
	MAE	MAPE	MAE	MAPE
Linear Regression	2.96	8.78	5.54	22.65
Random Forest	2.39	7.32	4.90	19.28
Support Vector Regressor	1.81	5.54	4.51	15.56

From Table 4.2, we observed the performance of standard machine learning algorithms after preprocessing the input data to extract the morphological biometric features. The best

Table 4.3: Weight Prediction Results of Deep Learning Methods

Model	Train		Test	
	MAE	MAPE	MAE	MAPE
2D-CNN	1.57	3.47	4.06	13.67
CNN+ MRA	1.33	3.49	3.17	9.72
CNN+ Segmentation	1.89	4.02	3.05	9.57
ResNet34	0.99	2.49	3.67	13.49
XceptionNet	1.47	3.61	2.82	7.42

performing model among statistical methods is Support Vector Machines (SVM), with a test MAPE of 15.56%. The results highlight that the importance of image quality control, as applying statistical methods without ensuring image quality can lead to unreliable accuracy. Moreover, training these models on all types of images, including pigs in various positions (such as bent over, lying down, sitting, etc), may contribute to subpar performance.

In contrast, the DL model results in Table 4.3 demonstrate superior performance. Both 2D-CNN and ResNet, as standard deep learning models, achieve an average MAPE of 13.58% surpassing the best performing statistical model (4.12). This indicates that DL models excel at learning high-level features incrementally from the data.

Additionally, DL models exhibit lower inference time, as demonstrated by our XceptionNet model, which took only 42 seconds to process test data consisting around 25k images. This emphasizes the suitability of DL models for real-time applications, even with large datasets and complex architectures.

However, a drawback of DL methods is the longer training time due to the large number of parameters. For example, our XceptionNet model consisted of 21 million parameters and took approximately 8 hours to train for 20 epochs on a GPU with 32 GB of RAM. To address these limitations, future studies will leverage faster hardware, such as GPUs or TPUs, to accelerate the matrix and convolution operations, ultimately reducing the training time. In

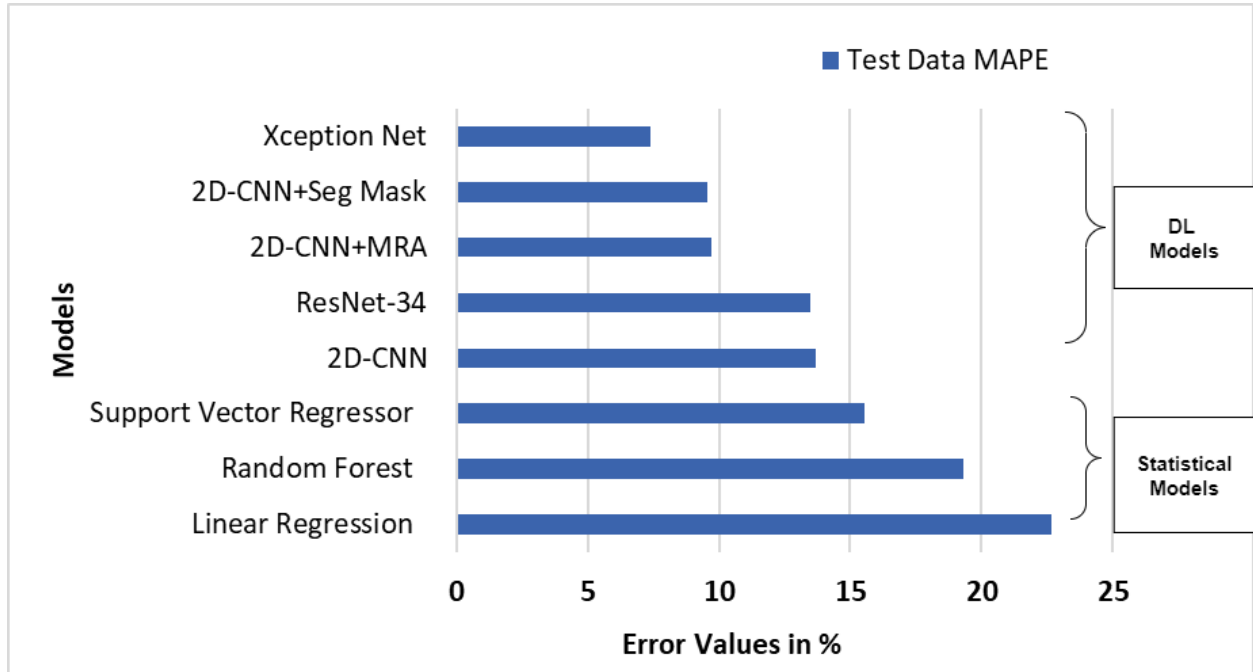


Figure 4.12: Comparison of Statistical and deep learning model performance on weight evaluation. Deep learning models clearly outperform their statistical model counterparts.

addition, we plan to incorporate more variability in the data by including pigs of different ages, breeds, sexes, and body sizes during data collection. This approach aims to improve model robustness by capturing a broader spectrum of features and characteristics.

Chapter 5

Conclusions

With increasing demand in the livestock sector, it is pertinent to have viable, efficient, scalable solutions in place to make the livestock farming process economical and convenient for the farmers.

We believe this study takes us one step closer to that goal of having automated solutions making the lives of farmers easier. In this thesis, we broadly discussed two applications, first one is the behavior classification using automated, sensor-based monitoring systems offer a promising avenue for improving monitoring in livestock farming practices. We introduced and assessed a novel data preparation framework aimed at eliminating data leakage in pig behavior analysis. The results revealed that alternative data preparation methods frequently resulted in overfitting on the training set, failing to accurately represent performance on a real-world test set encountered in practical settings. By effectively addressing data leakage, the proposed data preparation framework markedly enhanced classification performance. Specifically, we observed an impressive increase of 15.6% in model accuracy after implementing these strategies. These findings hold significant implications for researchers and practitioners, emphasizing the importance of designing efficient sampling and data preparation strategies for similar tasks in the future.

Secondly, we focus on weight prediction system in pigs. Our research demonstrates that weight prediction can be achieved with high accuracy using only depth images, as opposed to systems using multi-modal networks requiring RGB images and Point Cloud data. We

present a thorough comparative analysis of pig weight prediction techniques, encompassing both statistical and deep learning methods. Our experiments reveal that deep learning methods offer enhanced accuracy with reduced inference time compared to traditional statistical approaches, with an overall reduction of 8.14% in MAPE of best performing DL model, XceptionNet, as compared to the best performing ML model of Support Vector Regressor.

In our future analysis, to measure bias in the results, we would include residual analysis. We can compute the residual between predicted and actual values and check if the residual mean is consistently different from zero. In case of bias, we can enforce more regularization on the model like lasso or ridge regression.

Additionally, these deep learning models show potential for real-time weight prediction applications due to their superior accuracy and efficiency. A significant aspect of our study is the integration of SAM, an advanced segmentation model, into our preprocessing pipeline. To our knowledge, SAM has not been previously used in weight prediction tasks, rendering our approach novel and innovative. Moreover, the simplicity and implementation of our method, requiring only a depth camera positioned above the drive channel, make it easily adaptable for widespread use in various animal science settings.

Results for both the sections are promising with a fair trade-off between the accuracy performance and the latency in getting results. Based on the results in both the applications, we can say that DL models are an ideal option for precision livestock applications, requiring no aspect of feature engineering, saving manual efforts.

For future research, firstly, we would like to expand our current methodology to facilitate the forecasting of pig weights. This application is essential as it provides livestock owners with the ability to plan and manage their animals more effectively. Accurate weight forecasting enables farmers to make informed decisions regarding feed management, health

interventions, and overall herd management. Additionally, it allows for better planning of marketing and sales strategies, ultimately improving the efficiency and profitability of pig farming operations.

Secondly, incorporating cutting-edge attention-based modeling techniques could further enhance the performance of our predictive models. Recent trends in computer vision have shown that attention-based models can effectively capture intricate patterns and relationships within complex datasets. By integrating attention mechanisms into our deep learning architectures, we can improve the model's ability to focus on relevant features and enhance its predictive accuracy. This approach has the potential to further optimize weight prediction accuracy and contribute to more efficient and sustainable livestock farming practices.

Lastly, with regards to our data collection efforts, we plan on increasing the number of pigs and varying the sample space by including pigs of different breeds, ages, and genders. We believe that this will improve the overall diversity of the dataset and enhance the robustness of our trained models. A diverse dataset ensures that the models are exposed to a wide range of scenarios, making them more adaptable to real-world conditions. By including pigs of different breeds, ages, and genders, we can better capture the variability in pig behavior and physiology. These advancements have the potential to revolutionize the livestock farming industry by offering more accurate, efficient and reliable behavior classification and weight prediction models.

Bibliography

- [1] C.A. Aguilar-Lazcano, I.E. Espinosa-Curiel, J.A. Ríos-Martínez, F.A. Madera-Ramírez, and H. Pérez-Espinosa. Machine learning-based sensor data fusion for animal monitoring: Scoping review. *Sensors*, 23(1):5732, 2023.
- [2] Peter Ahrendt, Torben Gregersen, and Henrik Karstoft. Development of a real-time computer vision system for tracking loose-housed pigs. *Computers and Electronics in Agriculture*, 76(2):169–174, 2011. ISSN 0168-1699. doi: <https://doi.org/10.1016/j.compag.2011.01.011>. URL <https://www.sciencedirect.com/science/article/pii/S0168169911000263>.
- [3] Saleh Alghamdi, Zhuqing Zhao, Dong S. Ha, Gota Morota, and Sook S. Ha. Improved pig behavior analysis by optimizing window sizes for individual behaviors on acceleration and angular velocity data. *Journal of Animal Science*, 100(11):skac293, November 2022.
- [4] Marta E. Alonso, José R. González-Montaña, and Juan M. Lomillos. Consumers’ concerns and perceptions of farm animal welfare. *Animals*, 10(3), 2020. ISSN 2076-2615. doi: 10.3390/ani10030385. URL <https://www.mdpi.com/2076-2615/10/3/385>.
- [5] N. R. Augspurger and M. Ellis. Weighing affects short-term feeding patterns of growing-finishing pigs. *Canadian Journal of Animal Science*, 82(3):445–448, 2002. doi: 10.4141/A01-046. URL <https://doi.org/10.4141/A01-046>.
- [6] A. Bagnall, J. Lines, J. Hills, and A. Bostrom. Time-series classification with cote: the collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2522–2535, 2015.

- [7] M. G. Baydogan, G. Runger, and E. Tuv. A bag-of-features framework to classify time series. *IEEE transactions on pattern analysis and machine intelligence*, 35(11): 2796–2802, 2013.
- [8] Suvarna Bhoj, Ayon Tarafdar, Anuj Chauhan, Mukesh Singh, and Gyanendra Kumar Gaur. Image processing strategies for pig liveweight measurement: Updates and challenges. *Computers and Electronics in Agriculture*, 193:106693, 2022.
- [9] Michel Bonneau and Bénédicte Lebret. Production systems and influence on eating quality of pork. *Meat science*, 84(2):293–300, 2010.
- [10] A. Bostrom and A. Bagnall. Binary shapelet transform for multiclass time series classification. In *International conference on big data analytics and knowledge discovery*, volume 9263, pages 257–269. Springer, 2015.
- [11] Nabil Brandl and Erik Jørgensen. Determination of live weight of pigs from dimensions measured using image analysis. *Computers and Electronics in Agriculture*, 15(1):57–72, 1996. ISSN 0168-1699. doi: [https://doi.org/10.1016/0168-1699\(96\)00003-8](https://doi.org/10.1016/0168-1699(96)00003-8). URL <https://www.sciencedirect.com/science/article/pii/0168169996000038>.
- [12] Patrick Busch, Hartmut Ewald, and Frank Stupmann. Determination of standing-time of dairy cows using 3d-accelerometer data from collars. pages 1–4, 12 2017. doi: 10.1109/ICSensT.2017.8304492.
- [13] Yan Cang, Hengxiang He, and Yulong Qiao. An intelligent pig weights estimate method based on deep learning in sow stall environments. *IEEE Access*, 7:164867–164875, 2019.
- [14] Chen Chen, Weixing Zhu, Yizheng Guo, Changhua Ma, Weijia Huang, and Chengzhi Ruan. A kinetic energy model based on machine vision for recognition of aggressive behaviours among group-housed pigs. *Livestock Science*, 218:70–78, 2018. ISSN 1871-1413.

- doi: <https://doi.org/10.1016/j.livsci.2018.10.013>. URL <https://www.sciencedirect.com/science/article/pii/S187114131830564X>.
- [15] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.
- [16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.
- [17] Qi-an Ding, Jia Chen, Ming-xia Shen, and Long-shen Liu. Activity detection of suckling piglets based on motion area analysis using frame differences in combination with convolution neural network. *Computers and Electronics in Agriculture*, 194:106741, 2022. doi: 10.1016/j.compag.2022.106741.
- [18] A. B. Doeschl-Wilson, C. T. Whittemore, P. W. Knap, and C. P. Schofield. Using visual image analysis to describe pig growth in terms of size and shape. *Animal Science*, 79(3):415–427, 2004. doi: 10.1017/S1357729800090287.
- [19] Yuvraj Domun, Lene Juul Pedersen, David White, Olutobi Adeyemi, and Tomas Norton. Learning patterns from time-series data to discriminate predictions of tail-biting, fouling and diarrhoea in pigs. *Computers and Electronics in Agriculture*, 163:104878, 2019.
- [20] Irene Doyle and Steve Leeson. Automatic weighing of poultry reared on a litter floor. *Canadian Journal of Animal Science*, 69(4):1075–1081, 1989. doi: 10.4141/cjas89-122. URL <https://doi.org/10.4141/cjas89-122>.
- [21] S. Fan, Y. Jia, and C. Jia. A feature selection and classification method for activity recognition based on an inertial sensing unit. *Information*, 10(10):290, 2019.

- [22] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4): 917–963, 2019.
- [23] Ilaria Fontana, Emanuela Tullo, Andy Butterworth, and Marcella Guarino. An innovative approach to predict the growth in intensive poultry farming. *Computers and electronics in agriculture*, 119:178–183, 2015.
- [24] Guilherme A. Franchi, Jacinta D. Bus, Iris J.M.M. Boumans, Eddie A.M. Bokkers, Margit Bak Jensen, and Lene Juul Pedersen. Estimating body weight in conventional growing pigs using a depth camera. *Smart Agricultural Technology*, 3:100117, 2023. ISSN 2772-3755. doi: <https://doi.org/10.1016/j.atech.2022.100117>. URL <https://www.sciencedirect.com/science/article/pii/S277237552200082X>.
- [25] John Cristian Borges Gamboa. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.
- [26] Yue Geng and Xinyu Luo. Cost-sensitive convolutional neural networks for imbalanced time series classification. *Intelligent Data Analysis*, 23(2):357–370, 2019.
- [27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [28] Mette Herskin and Karin Jensen. Effects of different degrees of social isolation on the behaviour of weaned piglets kept for experimental purposes. *Animal Welfare*, 9(3): 237–249, 2000.
- [29] Weijia Huang, Weixing Zhu, Changhua Ma, Yizheng Guo, and Chen Chen. Identification of group-housed pigs based on gabor and local binary pattern features.

- Biosystems Engineering*, 166:90–100, 2018. ISSN 1537-5110. doi: <https://doi.org/10.1016/j.biosystemseng.2017.11.007>. URL <https://www.sciencedirect.com/science/article/pii/S1537511017300041>.
- [30] T. Huynh and B. Schiele. Analyzing features for activity recognition. In *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages and Technologies*, pages 159–163. Springer, 2005.
- [31] TTT Huynh, AJA Aarnink, and MWA Verstegen. Reactions of pigs to a hot environment. In *Livestock Environment VII, 18-20 May 2005, Beijing, China*, page 544. American Society of Agricultural and Biological Engineers, 2005.
- [32] M. A. Kashiha, C. Bahr, S. Ott, C. P. Moons, T. A. Niewold, F. Tuyttens, and D. Berckmans. Automatic monitoring of pig locomotion using image analysis. *Livestock Science*, 159:141–148, 2014.
- [33] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.
- [34] Jørgen Kongsro. Estimation of pig weight using a microsoft kinect prototype imaging system. *Computers and Electronics in Agriculture*, 109:32–35, 2014. ISSN 0168-1699. doi: <https://doi.org/10.1016/j.compag.2014.08.008>. URL <https://www.sciencedirect.com/science/article/pii/S0168169914002075>.
- [35] Junbin Liu, Deqin Xiao, Youfu Liu, and Yigui Huang. A pig mass estimation model based on deep learning without constraint. *Animals*, 13(8), 2023. ISSN 2076-2615. doi: [10.3390/ani13081376](https://doi.org/10.3390/ani13081376). URL <https://www.mdpi.com/2076-2615/13/8/1376>.

- [36] Yao Liu, Jie Zhou, Yifan Bian, Taishan Wang, Hongxiang Xue, and Longshen Liu. Estimation of weight and body measurement model for pigs based on back point cloud data. *Animals*, 14(7), 2024. ISSN 2076-2615. doi: 10.3390/ani14071046. URL <https://www.mdpi.com/2076-2615/14/7/1046>.
- [37] S.G. Matthews, A.L. Miller, J. Clapp, T. Plötz, and I. Kyriazakis. Early detection of health and welfare compromises through automated detection of behavioural changes in pigs. *The Veterinary Journal*, 217:43–51, 2016.
- [38] S.G. Matthews, A.L. Miller, and T. Plötz. Automated tracking to measure behavioural changes in pigs for health and welfare monitoring. *Sci Rep*, 7:17582, 2017.
- [39] Juliette Schillings, Richard Bennett, and David Christian Rose. Exploring the potential of precision livestock farming technologies to help address farm animal welfare. *Frontiers in Animal Science*, 2, 2021. ISSN 2673-6225. doi: 10.3389/fanim.2021.639678. URL <https://www.frontiersin.org/articles/10.3389/fanim.2021.639678>.
- [40] C.P. Schofield, J.A. Marchant, R.P. White, N. Brandl, and M. Wilson. Monitoring pig growth using a prototype imaging system. *Journal of Agricultural Engineering Research*, 72(3):205–210, 1999. ISSN 0021-8634. doi: <https://doi.org/10.1006/jaer.1998.0365>. URL <https://www.sciencedirect.com/science/article/pii/S0021863498903651>.
- [41] Carolinne Secco, Leila Mendes da Luz, Eliane Pinheiro, Antonio Carlos de Francisco, Fabio Neves Puglieri, Cassiano Moro Piekarski, and Fausto Miguel Cereja Seixas Freire. Circular economy in the pig farming chain: Proposing a model for measurement. *Journal of Cleaner Production*, 260:121003, 2020.
- [42] Joanna Stachowicz, Roland Nasser, Felix Adrion, and Christina Umstätter. Can we detect patterns in behavioral time series of cows using cluster analysis? *Journal of Dairy Science*, 105(12):9971–9981, 2022.

- [43] Sirimonpak Suwannakhun and Patasu Daungmala. Estimating pig weight with digital image processing using deep learning. In *2018 14th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, pages 320–326, 2018. doi: 10.1109/SITIS.2018.00056.
- [44] Sirimonpak Suwannakhun and Patasu Daungmala. Estimating pig weight with digital image processing using deep learning. In *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 320–326. IEEE, 2018.
- [45] Tzutalin. Labelimg. Free Software: MIT License, 2015. URL <https://github.com/tzutalin/labelImg>.
- [46] Z. Wang, W. Yan, and T. Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017.
- [47] Zhiguang Wang and Tim Oates. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. 01 2015.
- [48] RP White, CP Schofield, DM Green, DJ Parsons, and CT Whittemore. The effectiveness of a visual image analysis (via) system for monitoring the performance of growing/finishing pigs. *Animal Science*, 78(3):409–418, 2004.
- [49] Lauren Williams, G.J. Bishop-Hurley, Angela Anderson, and Dave Swain. Application of accelerometers to record drinking behaviour of beef cattle. *Animal Production Science*, 59, 01 2017. doi: 10.1071/AN17052.
- [50] Haipeng Yu, Kiho Lee, and Gota Morota. Forecasting dynamic body weight of nonrestrained pigs from images using an rgb-d sensor camera. *Translational Animal Science*, 5(1):txab006, 2021.

- [51] K. Zhang, D. Li, J. Huang, and Y. Chen. Automated video behavior recognition of pigs using two-stream convolutional networks. *Sensors*, 20(4):1085, 2020. doi: 10.3390/s20041085.
- [52] L. Zhang, W. Guo, C. Lv, M. Guo, M. Yang, Q. Fu, and X. Liu. Advancements in artificial intelligence technology for improving animal welfare: Current applications and research progress. *Animal Research and One Health*, pages 1–17, 2023.
- [53] Xinyue Zhang, Gang Liu, Ling Jing, and Siyao Chen. Automated measurement of heart girth for pigs using two kinect depth sensors. *Sensors*, 20(14), 2020. ISSN 1424-8220. doi: 10.3390/s20143848. URL <https://www.mdpi.com/1424-8220/20/14/3848>.

Appendices

Appendix A

Segment Anything Model

In this study we have used the state of the art segmentation model, Segment Anything Model (SAM) released by Meta AI in February, 2022. SAM is capable of performing both automatic and interactive segmentation simultaneously. This model is pre-trained on 11M images and 1B+ masks, all of which has been made public. This is the first of its kind prompt-able model, with zero shot inference.

SAM was trained to generate a valid segmentation mask based on various types of prompts, including foreground/background points, rough boxes or masks, freeform text, or any other information indicating what should be segmented in an image. A valid mask is one that provides a reasonable segmentation even when the prompt is ambiguous and could refer to multiple objects. This training task serves to pre-train the model and enables it to perform general downstream segmentation tasks through prompting.

The model must operate in real-time on a CPU within a web browser, allowing annotators to interactively use SAM for efficient annotation. Although the runtime constraint implies a trade-off between quality and runtime, the team found that a simple design yields satisfactory results in practice.

In operation [A.1](#), an image encoder generates a one-time embedding for the image, while a lightweight encoder dynamically converts any prompt into an embedding vector in real-time. These two sources of information are then combined in a lightweight decoder to predict

Universal segmentation model

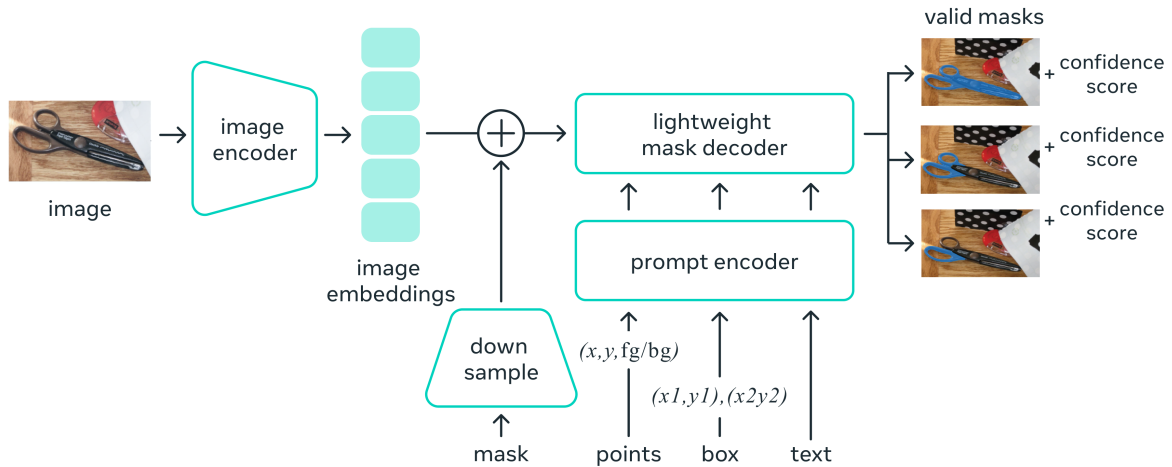


Figure A.1: Data Flow in the SAM model. We can see here that once an image is given an input it goes through encoders and concatenates with encoded prompts to finally go through decoder to get the final output. Courtesy : <https://ai.meta.com/blog/segment-anything-foundation-model-image-segmentation/>

segmentation masks. Once the image embedding is computed, SAM can produce a segment in just 50 milliseconds, given any prompt in a web browser.

Appendix B

Model Details

B.1 Residual Networks

This section discusses the 2D residual networks (ResNet-34) used in this study. This algorithm design can be utilized in both sensor and image-based analysis, with appropriate tweaks in input and output dimensions. The ResNet-34 architecture consists of several residual blocks. Each block contains multiple convolutional layers. There are four stages of residual blocks, each with a different number of blocks. The number of filters in the residual blocks is doubled every stage. Each stage starts with a convolutional block and is followed by several identity blocks.

In our design, the convolutional block in each stage includes two convolutional layers. The first convolutional layer has a kernel size of 3x3 and a stride of 2, effectively down-sampling the input. Batch normalization and ReLU activation functions are applied after each convolutional layer. A residual connection is added to the output of the second convolutional layer. The residual connection includes a 1x1 convolutional layer with a stride of 2 to match the dimensions of the main path.

Algorithm 1 2D ResNet Model

```

1: function IDENTITY_BLOCK( $x$ ,  $filter$ )
2:    $x_{skip} \leftarrow x$  ▷ Copy tensor to  $x_{skip}$ 
3:    $x \leftarrow \text{Conv2D}(x, \text{filters} = filter, \text{kernel\_size} = (3, 3), \text{padding} = \text{'same'})$ 
4:    $x \leftarrow \text{BatchNormalization}(x, \text{axis} = 3)$ 
5:    $x \leftarrow \text{Activation}(x, \text{'relu'})$ 
6:    $x \leftarrow \text{Conv2D}(x, \text{filters} = filter, \text{kernel\_size} = (3, 3), \text{padding} = \text{'same'})$ 
7:    $x \leftarrow \text{BatchNormalization}(x, \text{axis} = 3)$ 
8:    $x \leftarrow \text{Add}([x, x_{skip}])$ 
9:    $x \leftarrow \text{Activation}(x, \text{'relu'})$ 
10:  return  $x$ 
11: end function
12: function CONVOLUTIONAL_BLOCK( $x$ ,  $filter$ )
13:   $x_{skip} \leftarrow x$  ▷ Copy tensor to  $x_{skip}$ 
14:   $x \leftarrow \text{Conv2D}(x, \text{filters} = filter, \text{kernel\_size} = (3, 3), \text{strides} = (2, 2))$ 
15:   $x \leftarrow \text{BatchNormalization}(x, \text{axis} = 3)$ 
16:   $x \leftarrow \text{Activation}(x, \text{'relu'})$ 
17:   $x \leftarrow \text{Conv2D}(x, \text{filters} = filter, \text{kernel\_size} = (3, 3), \text{padding} = \text{'same'})$ 
18:   $x \leftarrow \text{BatchNormalization}(x, \text{axis} = 3)$ 
19:   $x_{skip} \leftarrow \text{Conv2D}(x_{skip}, \text{filters} = filter, \text{kernel\_size} = (1, 1), \text{strides} = (2, 2))$  ▷
    Process residue
20:   $x \leftarrow \text{Add}([x, x_{skip}])$ 
21:   $x \leftarrow \text{Activation}(x, \text{'relu'})$ 
22:  return  $x$ 
23: end function

```

```
1: function ResNet34(shape = (128, 6, 1), classes = 6)
2:   x_input ← Input(shape)
3:   x ← Conv2D(x, filters = 64, kernel_size = 7, strides = 2, padding = 'same')
4:   x ← BatchNormalization(x)
5:   x ← Activation(x, 'relu')
6:   x ← MaxPool2D(x, pool_size = 3, strides = 2, padding = 'same')
7:   block_layers ← [3, 4, 6, 3]
8:   filter_size ← 64
9:   for i ← 0 to 3 do
10:    if i == 0 then
11:      for j ← 0 to block_layers[i] - 1 do
12:        x ← identity_block(x, filter_size)
13:      end for
14:    else
15:      filter_size ← filter_size × 2
16:      x ← convolutional_block(x, filter_size)
17:      for j ← 0 to block_layers[i] - 2 do
18:        x ← identity_block(x, filter_size)
19:      end for
20:    end if
21:  end for
22:  x ← AveragePooling2D((2, 2), padding = 'same')(x)
23:  x ← Flatten()(x)
24:  x ← Dense(x, units = 512, activation = 'relu')
25:  x ← Dense(x, units = classes, activation = 'softmax')
26:  model ← Model(inputs = x_input, outputs = x, name = "ResNet34")
27:  return model
28: end function
```
