PriMera Scientific Engineering
Volume 5 Issue 2 August 2024
DOI: 10.56831/PSEN-05-143
ISSN: 2834-2550

PriMera Scientific Publications

Adaptive Deep Learning for Image-Based Estrus Prediction and Detection in Dairy Cows

Type: Research Article
Received: July 10, 2024
Published: July 16, 2024

Citation:
Adisak Sangsongfa., et al. "Adaptive Deep Learning for Image-Based Estrus Prediction and Detection in Dairy Cows". PriMera Scientific Engineering 5.2 (2024): 12-37.

Watchara Ninphet, Noppadol Amdee and Adisak Sangsongfa*

Doctor of Industrial Technology Management, Faculty of Industrial Technology, Muban Chom Bueng Rajabhat University, Thailand

***Corresponding Author:** Adisak Sangsongfa, Doctor of Industrial Technology Management, Faculty of Industrial Technology, Muban Chom Bueng Rajabhat University, Thailand.

Abstract

Artificial intelligence (AI) technology is significant in modern daily life. It is so influential that many consider technology the cornerstone of this era. Even in agriculture, there is a new concept known as Smart Farming. In a recent study, deep learning was adapted for predicting and detecting estrus in cows by adjusting the parameters of the deep learning model. A Convolutional Neural Network utilizing the Artificial Immunity System algorithm was employed to optimize the hyperparameters. The results of this optimization showed an accuracy of 98.361%. YOLOv5 deep learning was also used to detect real-time estrus, with mAP50 = 0.995, mAP50-95 = 0.887, and F1-Score averages = 0.993.

Keywords: dairy cows; deep learning; estrus prediction; image

Introduction

AI stands for artificial intelligence, which refers to the processing system found in computers, robots, machines, and electronic devices. This system has the capacity for in-depth analysis similar to human intelligence and can take actions based on its analysis. The learning process of AI is comparable to that of humans, which includes remembering, understanding, responding to language, making decisions, and solving problems. AI relies on large amounts of repetitive data to function effectively and bring maximum benefit. It's essential to analyze and select AI systems based on their intended use and the quality of the information used for making predictions [1]. Deep learning is another exciting science. From its meaning and explanation, it can be said that deep learning is one of the functions of artificial intelligence (AI) that learns how the human brain works in processing data and creating models for use in decision-making. Deep Learning is also a subset of Machine Learning in Artificial Intelligence (AI). It is a robust network of Unsupervised Learning from unstructured and unsupervised data. This is also known as Deep Neural Learning and Deep Neural Network [2].

However, deep learning algorithms require virtual neural networks, similar to how the nervous system works in the human brain. These networks have Neurons connected to our Nervous system and communication It uses parallel processing methods to make it able to understand and learn from the large amounts of data it receives continuously [3]. This makes deep learning capable of being similar to the human mind, enabling researchers to apply the algorithm to complex tasks that require high accuracy, as in the following example. Frank Emmert-Streib [4] provides an introductory review of various deep learning approaches, which include Deep Feed-forward Neural Networks (D-FFNN), Convolutional Neural Networks (CNNs), Deep Belief Networks (DBNs), Autoencoders (AEs), and Long Short-Term Memory (LSTM) networks. These models form the fundamental architecture of modern deep learning models, and their differences and usefulness are described, along with their benefits for researchers who utilize them. Matan Fintz and others [5] utilized deep learning in their research. They trained an exploratory Deep Neural Network (DNN) model to predict human decisions in four different scenarios of robberies. The researchers discovered that the DNN model was more accurate than the explicit model. They developed two models: the reward-focused model, which prioritizes selecting the most rewarding option, and models that ignore rewards but are trained to predict human decisions without reward information. The research revealed predictable decision-making patterns that do not solely focus on rewards and suggested that this may impact human decision-making. Notably, the researchers demonstrated how theory-driven perception models can be employed to analyze the behavior of DNNs, thereby making DNNs a valuable explanatory tool in scientific investigations. Reza Arablouei et al. [6] developed an end-to-end deep neural network-based algorithm for classifying animal behavior using acceleration data on the embedded artificial intelligence of things (AIoT) devices installed in the tag. A wearable collar, the results show that real-time in situ behavioral inference can be achieved from accelerometer data. It does not affect the available computational resources, memory, or power of the embedded system, and the inference times are fast and accurate. Safdar Sardar Khan [7] researched using a combination of two convolutional neural network architectures, VGG-19 and DenseNet121, to develop an effective method for classifying animal species based on images. A hybrid deep learning model was introduced for predicting animal species using a dataset of 10 animals from Kaggle to improve accuracy. The results demonstrated that the combined use of these two architectures achieved an accuracy of 91.00% etc.

Based on the examples of research provided, it was found that deep learning can be widely applied in research. This can lead to effective prediction of results, and if implemented in industry, it could bring about significant beneficial changes. This is especially true for work that relies on new technology to increase productivity, particularly in the agricultural industry, a significant factor in human consumption. One agricultural industry that is important to human daily life is the dairy production industry because milk is an entirely natural food. and has high nutritional value Rich in all nutrient groups, namely protein, vitamins, minerals, carbohydrates, and fats, especially milk sugar or lactose, and protein called Casein is found only in nature, namely in milk or sap. Milk is, therefore, very important in developing the body and brain of children and youth [8]. Based on the report on milk consumption demand from 2014 to 2018, the demand for milk consumption in essential countries of the world showed a trend of increasing by 0.89 percent per year. In 2018, the demand for consumption reached 183.98 million tons, an increase of 0.80 percent from 182.52 million tons in 2017. India stands out as the country that consumes the most milk, at 66.80 million tons, followed by the European Union at 33.50 million tons and the United States at 26.20 million tons during the same period. The total number of dairy cows in Thailand is increasing by 2.82 percent annually. In 2018 (as of January 1), the country had 660,155 cows, an increase from 645,261 cows in 2017, accounting for a 2.31 percent increase. The number of milking cows is also rising by 1.52 percent per year. 2018 there were 276,321 milking cows, up from 267,932 cows in 2017, reflecting a 3.13 percent increase. From 2014 to 2018, raw milk production has been increasing by 1.62 percent per year. In 2018, the production reached 1,233,483 tons, up from 1,191,143 tons in 2017, a 3.55 percent increase. This growth is attributed to the birth of new dairy calves, contributing to the overall increase in production [9]. The average number of cows milked in a year has increased, indicating potential growth in the dairy industry and dairy farming, likely due to government promotion and growing public interest in health. For a cow to produce milk, it must have offspring. After giving birth, the cow will start giving milk, but the duration of milk production can vary depending on the cow's ability, breed, and other factors. Generally, milking can begin around 5-10 months after the cow gives birth. The uterus typically returns to normal within 30-70 days. If a cow shows signs of estrus within 25 days after giving birth, it should not be bred yet, as its reproductive organs are still recovering. Most breeders wait for the second estrus, typically 45-72 days after birth, before mating the cow [10]. The conclusion is that for a dairy cow to be able to produce milk, the mother cow must have a child. But before having a child, the young female cow must show signs of estrus first, which is an important symptom and doesn't show for a long time, agriculture needs to Watch out for estrus symptoms in dairy cows. Cows will be in estrus approximately 24 hours after the end of estrus. An egg will ovulate after 6-10 hours, and the sperm used for artificial insemination can stay in the uterus for 24-30 hours. Therefore, the best time for insemination is between the 12th and 30th hours. At the 24th hour of estrus, some cows show unclear signs of estrus. You must observe. During estrus, cows will show symptoms such as restlessness, snuggling with others, trying to mount other animals, and attempting to escape; it can be difficult to know which cows are in estrus. Cows in actual estrus will allow others to ride calm, which is a suitable time for mating. The vulva will be swollen, and the tail will bob slightly obliquely. Farmers should observe the signs of estrus; if not fertilized, the cow will return to estrus in 18-21 days. It's important to note that estrus symptoms usually appear at night [11].

Therefore, many researchers have focused on using technology to help detect estrus in cows, each of which has tried to ensure high confidence in the research results. For example: Valesca Vilela Andrade [12] was researched to evaluate changes in reticulo-rumen temperature (RRT) and activity (ACT) in heifers during estrus. The study involved 45 heifers and aimed to test different models for predicting estrus. The heifers' RRT and ACT were recorded using reticulo-rumen boluses at various intervals before and during estrus. Analysis of variance for RRT and ACT was performed using mixed models. Logistic regression, random forest, and linear discriminant analysis models were used with RRT, ACT, time of day, and temperature-humidity index (THI) as predictors. The study found that RRT and ACT increased significantly during estrus compared to the day before and after estrus. The random forest model showed the best performance, with a sensitivity of 51.69% and a specificity of 93.1%. These findings suggest that RRT and ACT can help identify estrus in Dairy Gyr heifers. Watchara Ninphet [13] presented this research and developed the prediction of estrus in dairy cows using artificial intelligence, which uses convolutional neural networks to help prediction. This research has compared the Convolutional Neural Network before adjusting parameters. It has an accuracy of 95.082%, and when using the Artificial Immunity System algorithm to adjust the parameters, it has an accuracy of 98.361%. The prediction uses 4 types of dairy cow movements to predict. Ryotaro Miura [14] presented research evaluating methods for predicting estrus focusing on the number of follicles ≥ 10 mm (large follicles) along with the corpus luteum that plays a role in lactation. Holstein dairy cattle This study divided cows into two groups with one large follicle and two or more follicles on ovarian examination using transrectal ultrasound. Testing for estrus was performed after an ovarian examination. Estrus greater than 75% occurred 10 days after ovarian examination in cows with 1 large follicle on ovarian examination, while estrus greater than 75% occurred 10 days after ovarian examination. Occurs within 9 days of ovarian examination in cows with two or more large follicles on ovarian examination. Therefore, the researchers suggest that assessing the number of follicles ≥ 10 mm may help predict the timing of estrus expression in lactating Holstein dairy cows. Researchers use this test to detect estrus because cows are more active when they are in estrus, and this movement can be measured with a test device called an estrus test. A "pedometer" estimates estrus using movement changes detected with an artificial neural network (ANN) model. From cow movement data, 78 animals showed 184 estrus and climate data over 7 months. In addition, the model also considers and evaluates various information such as the age of the cows, the amount of feeding, and the number of days elapsed from estrus. ANN, with a network count of two layers, with 37 neurons in the first layer and 40 neurons in the second layer, was the most successful model with a score of 0.1775 F-Score [15]. Jun Wang [16] conducted a study using machine learning to detect the onset of estrus in dairy cows based on their movement data. The study involved 296 lactating cows observed over 8 months, during which 325 estrus events occurred. The research focused on using an artificial neural network algorithm called Backpropagation (BPNN) with optimized parameters to automatically identify estrus from seven behavioral indicators. The study found that this method's performance was comparable to traditional methods such as support vector machine (SVM) and logistic regression (LR) algorithms. The results suggest this automated approach could be a reliable and efficient alternative to labor-intensive visual observation.

The research uses artificial intelligence, particularly deep learning technology, to predict estrus in dairy cows using images from CCTV cameras. The goal is to demonstrate the effectiveness of deep learning in tasks such as prediction, classification, and detection, potentially surpassing human capabilities. The study aims to optimize parameters using a heuristic algorithm and implement YOLO (You Only Look Once) deep learning as a Realtime Object Detection Model. This model is known for its speed and accuracy, which can be used to detect estrus in cows most efficiently.

Related work

You Only Look Once (YOLO)

Using AI to help analyze images in an era where information is accessible everywhere is not tricky. Suppose we send a picture of a cat for AI to analyze. Usually, a classic use case is an Image Classification, which means we want the AI to tell us what the image is, such as the example image with a picture of a cat. Typically, if it were a human, of course, everyone would know that it was a cat, but for AI we have to send the image with the label that it is a picture of a cat so that it keeps learning. And this is the primary job of something called Image Classification. Next, if we want to know more about where the cat is in the picture, we get the problem. This is another type of problem called Image Detection, which will tell you the location of the image. Of course, there are many ways to do Image Detection such as R-CNN, Faster R-CNN, EfficientDet, and YOLO. However, YOLO's advantage is that it will benefit from being sensitive to data analysis. We choose to use this YOLO and have ready-made documents, including Customized, to access our information and goals [17].

YOLO is one of the most popular object detection model architectures and algorithms. Arguably, it is one of the best neural network architectures for detecting images to create high accuracy and overall processing speed. This is the main reason for its popularity. When searching for words related to object detection, the Google search results are YOLO Algorithm. The YOLO algorithm aims to predict object classes and bounding boxes that determine the object's location within the imported image. It identifies each bounding box using four numbers: center of the bounding box (b_x, b_y), width of the bounding box (b_w), and height of the bounding box (b_h). Moreover, YOLO predicts the corresponding class number, c, and the prediction probability (P_c). As for the working principle of YOLO, the advantage is that it is fast as mentioned above because it analyzes the image once and then creates a Label to cover the entire image, where from one full image it divides the Grid cells into n × n grid (for example, from 1 image may be divided into 13 grids, the more divided, the more detailed the trade-off with the calculation) Then in each grid, we have to add a Label to it (if any) such as [P_c, b_x, b_y, b_h, b_w, c_1, c_2,. . . , c_n] where Pc is the Probability that there are objects in that Grid if there are none. is 0, if it is 1bx, by is the middle position of the Object, which coordinate is b_h, b_w is the height and width of the Object, how high, how wide, and c1, c2, .., cn is the result of what class it is. If our problem only detects one image, it will have only one class, and the grid has a value of 1, but if there are multiple objects, there will be successive numbers (e.g., The grid in the middle of that image will have C1 = 1, with other grids C1=0). When dealing with multiple objects, we use a principle called Anchor Box. This box allows us to set a quantity (e.g., 2). Within this box, will be 2 labels in 1 grid (the number can be adjusted), and YOLO will calculate which image is closest to the Anchor. Based on the Intersection over Union (IOU) value, each image will be assigned to the corresponding Anchor. However, the essential keywords of YOLO are Intersection over Union (IOU) and Non-max suppression (NMS), which can be briefly explained as follows [18].

Intersection Over Union (IOU)

IOU presents several critical roles in YOLO, primarily when the prediction model produces multiple box predictions for the same object. In such cases, IOU is used as a metric to select the most suitable box representing that specific object. IOU is often combined with NMS methods to refine YOLO results and identify objects in an image for future use. So, IOU measures the accuracy of our detection given the ground truth bounding box and the detected bounding box. We calculate IOU as the ratio of the overlap area and the combined area as follows [18]:

$$IOU = \frac{Area\ of\ the\ intersection\ between\ B_1\ and\ B_2}{Area\ of\ Union\ between\ B_1\ and\ B2} \quad (1)$$

where *B1* and *B2* are two bounding boxes.

From the IOU formula, it can be shown in Figure 1 as follows.

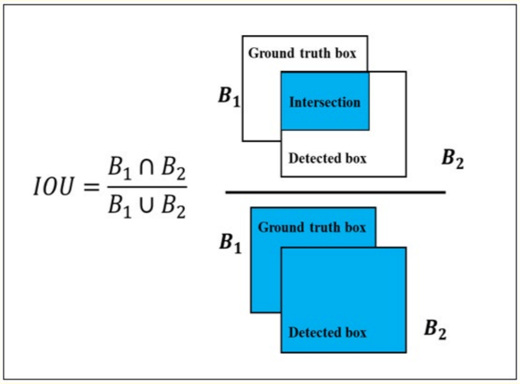

Figure 1: Represent the Intersection over Union.

Non-max suppression (NMS)

One potential issue when using object detection algorithms is when the algorithm predicts multiple bounding boxes for a single class. In such cases, we must select only one box with the highest probability per class. However, what if multiple objects of the same class are in the image? We can utilize the Non-Maximum Suppression (NMS) algorithm to address this. We select the box with the highest probability during the first suppression stage. Then, we compare this box with all other boxes of the same class using a metric called Intersection over Union (IOU). This metric measures the overlap between bounding boxes. In computer vision, IOU is also known as the Jaccard index. Figure 2 shows how NMS works with IOU in image detection [18].

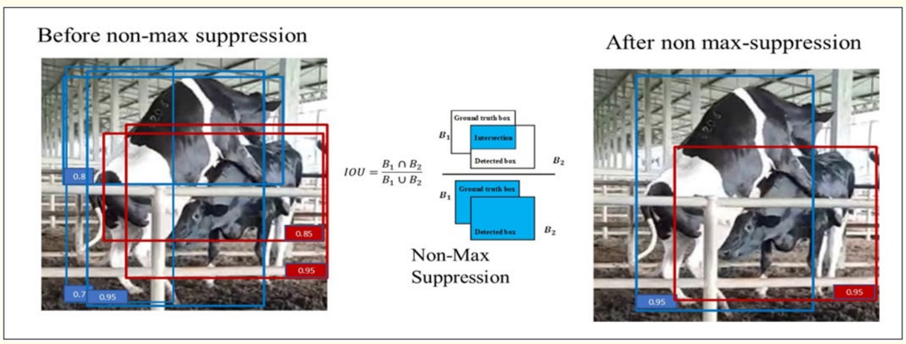

Figure 2: Represent the Intersection over Union and Non-Max Suppression.

Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) [19] is a type of neural network well-suited for processing and analyzing visual data, such as images and videos. CNNs are commonly used in computer vision applications, including object recognition, image classification, and face detection. These networks consist of layers of interconnected nodes, similar to other types of neural networks, but with a unique structure that makes them ideal for processing image data. A critical feature of CNNs is convolutional layers, which apply filters to input data to extract features. These filters are designed to identify specific patterns in the data, such as edges, shapes, or textures. Each convolutional layer learns to recognize different features, and combining all the layers allows CNNs to identify more complex patterns and objects. Therefore, the Convolutional Neural Network is a Deep Learning algorithm, that works in such a way that it receives input as an image and then learns various features of those images in each layer, continuing upwards from points, vertical lines, horizontal lines, diagonal lines, Crosshairs, angles, curves, circles, surfaces, patterns, eyes, faces, and other related things until reaching the object we define, as shown in the architecture in Figure 3. To understand more about convolutional neural networks and how they differ from other types of neural networks. Convolutional neural networks are particularly effective at processing data related to audio, images, words, or sounds. The operation of each layer can be explained as follows.

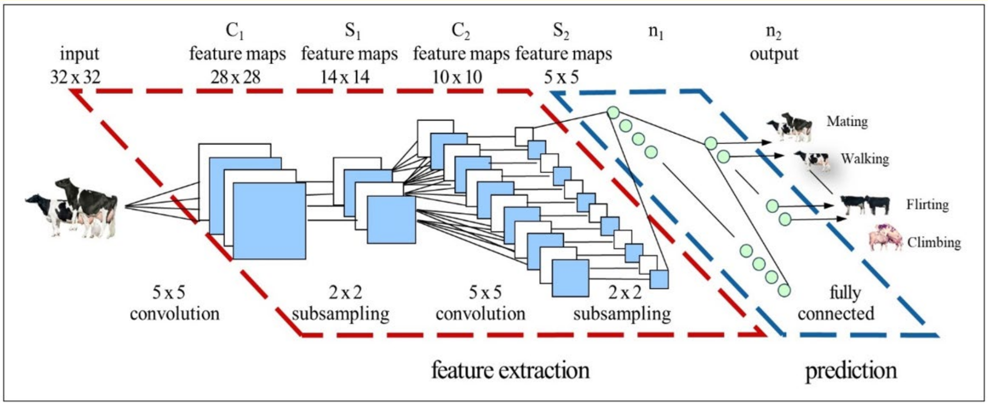

Figure 3: Represent the CNN architecture.

Convolutional layer

The convolution layer (ConvLayer) is the first layer of the CNN model and is responsible for extracting essential features from an image. Compared to a simple neural network that connects every neuron of the previous layer, it preserves the relationships of pixels in the same area. The ConvLayer connects only the desired location, called the receptive field. Different filters for image convolution have different meanings, such as finding edges, blur, sharpness, and straight lines at the first layer, while curves in later layers become more abstract. To understand the working of ConvLayer, it is enough to give an example: set Input size 3x3x1 and Weight size 2x2x1. The calculation process is to find the sum of the multiplication between Input and weight using the original weight set and then scan the entire Input from left to right, and from top to bottom, the result (Z) is

$$Z1 = W1X1 + W2X2 + W3X4 + W4X5 \quad (2)$$

$$Z2 = W1X2 + W2X3 + W3X5 + W4X6 \quad (3)$$

$$Z3 = W1X4 + W2X5 + W3X7 + W4X8 \quad (4)$$

$$Z4 = W1X5 + W2X6 + W3X8 + W4X9 \quad (5)$$

If written in Neural network format, it will be in Figure 4. There are differences between a regular neural network and a ConvNet. In a regular neural network, the weight values used to calculate Z are all the same, meaning the model has to learn many weight values. This is one reason ConvNets are used instead. A ConvNet consists of the convolutional layer and the pooling layer. The convolutional layer has hyperparameters including weight window size, input scaling (padding), and scanning step.

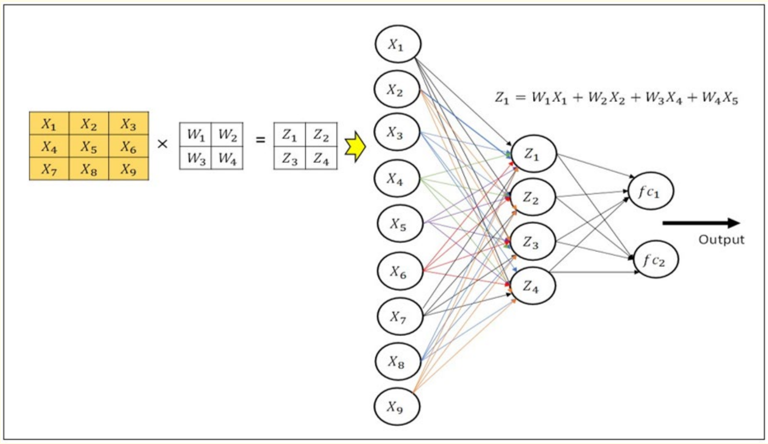

Figure 4: The figure shows CNN's working steps to write a Neural network model.

Pooling layer

The pooling layer is another type of layer in ConvNets and it involves data scaling. Instead of using the entire image data, the idea is to divide it into areas and then select some values from those areas. There are generally two types of pooling layers to choose from: 1) Max pooling: This involves selecting the maximum value from a pool of numbers. 2) Average pooling: This involves calculating the average of the numbers in the pool. To use pooling layers, we need to define certain hyperparameters, such as the scanning step (stride) and the size of the pool. For example, if we have an input size of 4x4, we can set the pool size to 2 and the stride to 2. This is illustrated in Figure 5.

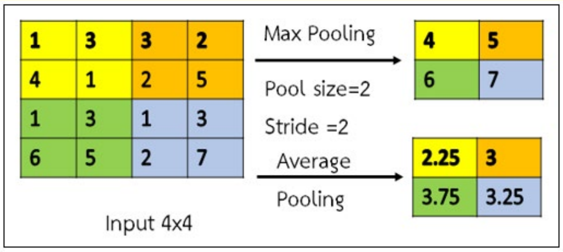

Figure 5: The figure shows the calculation in the Pooling layer.

In the Yellow Pool example, Max pooling selects the highest value in the pool, which is 4. For Average pooling, we find the average in the pool: (1 + 3 + 4 + 1)/4 = 2.25. The purpose of pooling is to reduce the data size so that the model does not need to learn values that weigh more than necessary.

Fully-connected (FC) layer

In this layer, receive output from Max pooling and then go to a new Convolution layer. The Output in this layer still considers the surrounding area as before. Still, the remarkable thing is that it doesn't consider each pixel in the Input image, but becomes considered from the Output of the previous layer that is considered. From the surrounding pixels again, we usually call the output that passes through the convolution layer a feature, where the feature from the first layer is a feature. Simple things like straight lines and curves, because from the image, the pixels next to it are only a small area, resulting in only the edge of a straight line, a short curve, or color information in that area in the color image. Then, when passing Max pooling in the next convolution layer will begin to be assembled into parts that accumulate over time, such as the example of separating images with or without facial features. The next layer will be ears, eyes, nose, and mouth, showing facial features. Then, everything will be combined to become a face, etc [20].

Materials and Methods

In this research, the researchers utilized deep learning with optimized hyperparameters to predict cows' estrus and achieve maximum accuracy in image analysis. They also adjusted the input data to use YOLO deep learning to detect estrus in cows. This AI technology can help farmers improve efficiency in farming. For more details, please refer to the research framework in Figure 6, which explains the following steps.

Data Preparation From the framework

The data preparation in this research can be divided into 2 parts: data preparation for predicting estrus in cows and data preparation for detecting estrus in cows. The detailed steps are explained below.

a) ***Data Preparation for prediction***; CCTV cameras were installed in a 4x8 cow pen with a roof at least 4 meters above the ground to prepare for capturing images of cows. The cows used in this experiment were dairy cows with Holstein-Friesian bloodlines 87.50% of the cows were graded 2nd and above, and they came from a dairy farm certified by the Department of Livestock Development to meet standards. There was a total of 10 cows involved in the experiment. The process consists of collecting real-time video images at a resolution of 1280 x 960 pixels and 16 frames per second. Data was collected over 3 months to capture sample video images of estrus detection in dairy cows, based on their climbing behavior. The images were then cropped to a size of 400x500 pixels with a resolution of 300x300 dpi to create a prediction model. In this research, 1,237 images were used, divided into 4 categories: 36 climbing images, 421 flirting images, 390 mating images, and 390 walking images.

b) ***Data Preparation for detection***; Preparing images for use in detecting estrus in cows, the researcher used 4 CCTV cameras installed in various positions in the cow barns to cover the area where movement and estrus could be observed. Cow estrus by movement starts when the cow walks towards the estrous cow (Walking) and shows symptoms of sniffing or licking (Flirting) until climbing on top (Mating) and climbing on top in the wrong position, such as climbing on the head (Climbing). Certified farm standards from the Livestock Department of the Ministry of Agriculture and the Dairy Cooperative were used in trials of Holstein Friesian dairy cows with Holstein Friesian bloodline levels ranging from 93.75% (93.75% HF). The recording process requires at least 3 months to collect image data. After that, the researcher separates the moving video clip into four still frames, each consisting of 380 images, for a total of 1520 images. Each image is labeled to indicate the direction of movement for detecting estrus in cows, as shown in the example image in Figure 8.

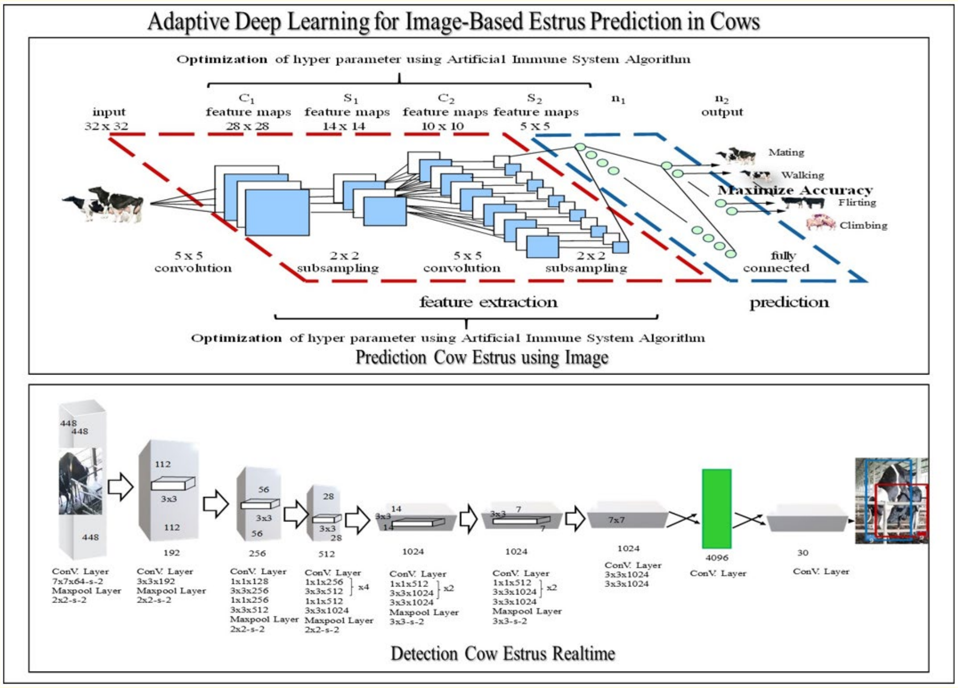

Figure 6: The figure shows the framework of Adaptive Deep Learning for Image-Based Estrus Prediction in Cows.

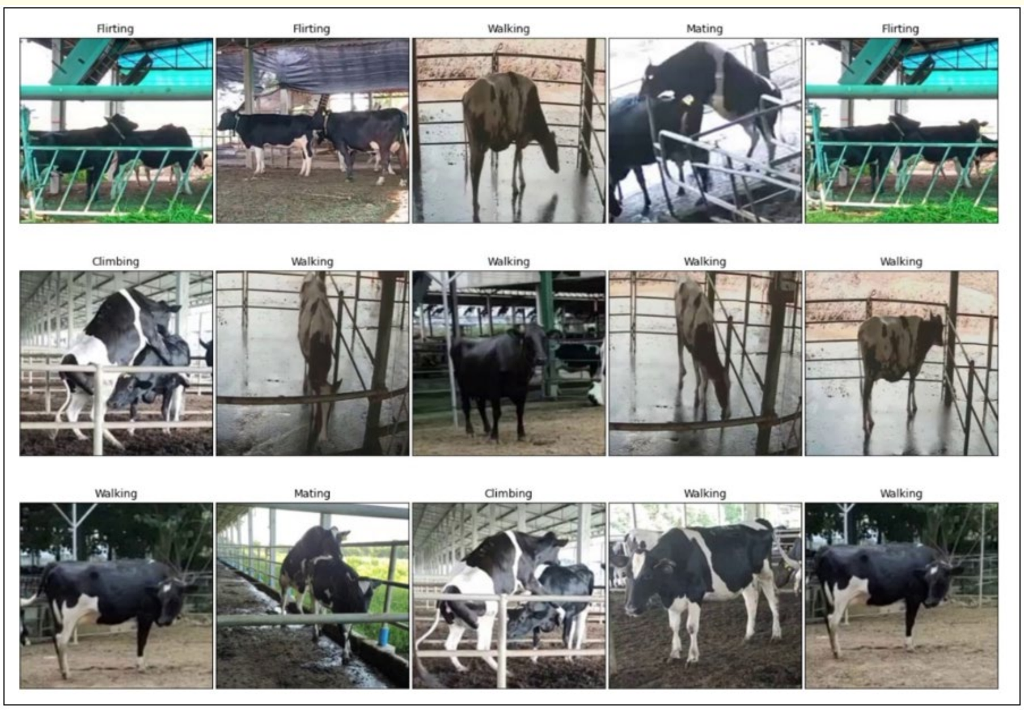

Figure 7: The figure shows an example data of the image of a dairy cow with 4 classes used for prediction.

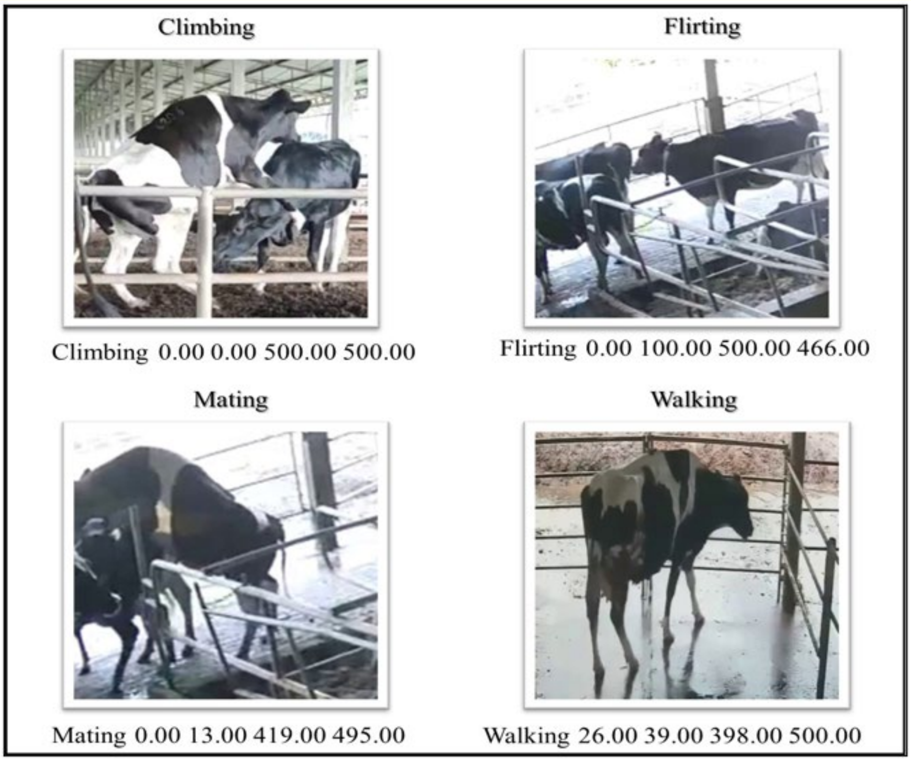

Figure 8: The figure shows example data for the image and label of a dairy cow with 4 classes used for detection.

Measuring the efficiency of the research

The performance measurement metrics used to evaluate the deep learning model in this research, including prediction and image processing for estrus detection, are as follows.

a) ***Accuracy*** refers to the number of samples that got the answer correct as a percentage of the total samples in the dataset. From the answer that matches the true answer to the answer that matches the wrong answer, the calculation formula is:

$$Prediction\ accuracy = \frac{Correct\ prediction}{Total\ predictions}\ x\ 100\% \quad (6)$$

b) ***Precision*** (Positive Accuracy) refers to the ability to correctly answer that is positive. When the model predicts a positive result, the calculation formula is:

$$Precision = \frac{Number\ of\ True\ Positive}{Number\ the\ model\ predicts\ to\ be\ Positivepredictions}\ x\ 100\% \quad (7)$$

c) ***Recall*** (ability to get all positive correct answers) refers to the ability to get all positive correct answers (available in the data set) when the model predicts them to be positive. The calculation formula is:

$$Recall = \frac{Number\ of\ True\ Positive}{Number\ of\ real\ Positive}\ x\ 100\% \quad (8)$$

d) ***F1-Score*** is the average of the consistency of Precision and Recall, which gives good results when both Precision and Recall are equally high. The formula for calculating is:

$$F1 - Score = 2\,x\,\frac{Precision \times Recall}{Precision + Recall} \quad (9)$$

e) ***Confusion Matrix*** is a matrix that shows the prediction results of the model into 4 fields: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN), where

- True Positive (TP) is a number that the model predicts is positive and is positive.
- False Positive (FP) is a number that the model predicts to be positive but it is negative.
- False Negative (FN) is a number that the model predicts to be negative but it is positive.
- True Negative (TN) is a number that the model predicts to be negative and It is negative.

		Actual Values	
		Positive (1)	Negative (0)
predictive Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 9: The figure shows a 2x2 confusion matrix table.

Using the Confusion Matrix helps to analyze problems, and improve the model, and can be calculated as Precision, Recall, and Accuracy as follows.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (10)$$

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

f) ***Mean Average Precision*** at K (mAP@k) Average Precision (AP) is the calculation of the AP of each data set by cutting the Threshold at rank any k that we are interested in and averaging the AP@k values of every data set obtained to get the mAP@k value. The calculation formula is:

$$mAP = \frac{1}{N}\sum_{i=1}^{N}(AP_i) \quad (13)$$

Configuration hyperparameters and optimal hyperparameters with artificial immunity system algorithm (AIS) of convolutional neural networks for prediction of cow estrus

The configuration hyperparameter settings for this research are divided into two categories: one for prediction and finding the most appropriate value using AIS, and another for detecting estrus in cows. The default settings for prediction are illustrated in Figure 10. The hyperparameters for prediction include 6 Conv2d filters with the following specifications - 32, 64, 64, 64, 64, 64, kernel size (4,4), stride (2,2), max pooling (2,2), Dense 64, Activation function Relu, padding same, dropout 0.25, batch size 128, activation value function of output softmax, and optimizer ADAM. The initial operation consists of 100 epochs. The images used have been scaled from

400x500 to 500x400. 70% of the images are used for learning, while the remaining 30% are used for testing. To account for uneven class distribution, percentage randomization is employed. The prediction classes are divided into 4 classes as described in the data preparation. Once the initial values for determining the most suitable value are acquired, this study uses the artificial immune system (AIS) to identify the optimal hyperparameter value. Figure 11 illustrates the pseudocode for determining the optimized hyperparameter value. The process involves identifying the number of filters in each layer to find the optimized value. The number of layers is fixed at 6, however, the variable value is the number of nodes filtered in each layer, with values obtained randomly from the range [28, 128]. Furthermore, the weight of the activation value function of the output using softmax can also be determined from random values within the range [0, 1]. The process runs for 100 epochs to find the optimal value.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 150, 150, 32)	2432
max_pooling2d (MaxPooling2D)	(None, 75, 75, 32)	0
conv2d_1 (Conv2D)	(None, 75, 75, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 37, 37, 64)	0
conv2d_2 (Conv2D)	(None, 37, 37, 96)	55392
max_pooling2d_2 (MaxPooling2D)	(None, 18, 18, 96)	0
conv2d_3 (Conv2D)	(None, 18, 18, 96)	83840
max_pooling2d_3 (MaxPooling2D)	(None, 9, 9, 96)	0
flatten (Flatten)	(None, 7776)	0
dense (Dense)	(None, 512)	3981824
activation (Activation)	(None, 512)	0
dense_1 (Dense)	(None, 5)	2565

Total params: 4,143,749
Trainable params: 4,143,749
Non-trainable params: 0

Figure 10: The figure shows configuration hyper-parameters optimized with adjustments from experiments.

Configuration the deep learning for detection images.

The deep learning technique used in this research to detect estrus in cows utilizes the You Only Look Once (YOLO) version 5 deep learning model with default values: batch size=32 and image size=640. The training involved 30 epochs, using 360 images per class for learning and 20 images per class for testing. In total, 1,520 images were used in the research.

Results and Discussion

In this section, the results of the experiment on predicting the estrus of cows by dividing them into 4 classes (Climbing, Flirting, Mating, and Walking), using the default hyperparameter settings as per item 3 of the research method, revealed an accuracy of 95.082%. The confusion values matrix is shown in Figure 12, and the prediction results are presented in Figure 13. These results were obtained from the initial setup and experimentation with hyperparameter adjustments by experts before using AIS to find the most optimized values. Once the researchers and experts established the hyperparameters, they utilized AIS negative selection algorithms to randomly define a population filter using the values pi= (f1, f2, f3, f4, f5, f6, lr) where i=1…100 and f1, f2, f3, f4, f5, f6, lr are a random number in interval. The model was trained with a population size of 100 for 100 epochs. The hyperparameters for prediction include 6 Conv2d filters with the following specifications - 32, 64, 128, 64, 32, 128, kernel size (4, 4), stride (2, 2), learning rate of 0.00000001, max pooling (2, 2), Dense 64, Relu activation function, padding set as same, dropout of 0.25, batch size of 128, softmax activation function for the output, and ADAM optimizer. The confusion matrix results can be seen in Figure 14, and the prediction images are displayed in Figure 15. Figures 14, 15, and 16 show that using the AIS algorithm to optimize the hyperparameters of the CNN increases the accuracy more than using experts to adjust the values, with an accuracy of 98.361%.

<u>Algorithm Optimized Training algorithm of CNN using Immunity System Algorithm</u>

Input:

train_x, train_y: feature and label of Training Set (Caw Mating, Walking, Flirting, Climbing)
test_x, test_y: feature and label of Test Set (Caw Mating, Walking, Flirting, Climbing)

Output:

$\boldsymbol{w_{ij}^l}$,$\boldsymbol{b_j^l}$**:** weights and bias of Convolutional and Pooling Neural Network (CPNN)
$\boldsymbol{w_{jk}}$, $\boldsymbol{b_{jk}}$**:** weights and bias of Full Connection Neural Network (FCNN)
prediction: Caw Mating, Walking, Flirting, Climbing

Required parameters:

max_time: maximum value n of nACs in every 1SP
target_error: when the current training error is less than the target error, the training is finished.
n_CPNN: the learning rate of CPNN

Initialization work:

$\boldsymbol{w_{ij}^l}$,$\boldsymbol{b_j^l}$, $\boldsymbol{w_{jk}}$, $\boldsymbol{b_{jk}}$**:** weights and scaling parameters of CNN (CPNN+FCNN) are set as random numbers.
t: t is the current simulation time, initialized as ***t=1*** before the training loop.
L(t): L(t) is the mean square error at simulation time ***t, L(t)*** is initialized as ***L(1) =1>target_error.***

Artificial Immunity system algorithms

Initialization the variable of Immunity for CNN $\boldsymbol{w_{ij}^l}$,$\boldsymbol{b_j^l}$, $\boldsymbol{w_{jk}}$, $\boldsymbol{b_{jk}}$ then define r_{ij} matrix, number of Dedicated cell ($\boldsymbol{w_{ij}^l}$,$\boldsymbol{b_j^l}$, $\boldsymbol{w_{jk}}$, $\boldsymbol{b_{jk}}$)and number of shared cells, upper bound and lower bound of part in each dedicated cell

Select the first combination of parts and machines as an initial solution (S_0).
For all part and machine combinations
Generate the *popsize* (200 items) of the S_0 permutation randomly as initial population, determine the iteration value according to size of S_0
Calculate the affinity function of these antibodies (*accuracy of prediction*)
For *max gen*
Select the best n_c ($\boldsymbol{w_{ij}^l}$,$\boldsymbol{b_j^l}$, $\boldsymbol{w_{jk}}$, $\boldsymbol{b_{jk}}$) of the antibodies
Make (n_c^{k+1}) copies of each antibodies (clone)
Mutate all of the cloned antibodies
Calculated the affinity function value for all of the mutated antibodies
The n_c of the worse antibodies in current population are replaced by the n_c of the best mutated antibodies
End for
Store the obtained best affinity function value
End for

Begin:

Set the required parameters and complete the initialization work
While *t< max_time* and ***L(t) > target_error***
for all trainingSet:
train_p (prediction of the label) is calculated according to ***tarin_x*** and forward calculation formula as bellow (1-9) :

$$net_{mn}^l = convolution(O^{l-1}, w^l, m, n) + b^l = \sum_{i=0}^{size^l-1}\sum_{j=0}^{size^l-1}\left(O_{m+i,n+j}^{l-1}\right).w_{i,j}^l + b^l \quad (1)$$

$$convolution(x,y) = \begin{matrix} x_{11}y_{11} + x_{12}y_{12} + x_{21}y_{21} + x_{22}y_{22} & x_{12}y_{11} + x_{13}y_{12} + x_{22}y_{21} + x_{23}y_{22} \\ x_{21}y_{11} + x_{22}y_{12} + x_{31}y_{21} + x_{32}y_{22} & x_{22}y_{11} + x_{23}y_{12} + x_{32}y_{21} + x_{33}y_{22} \end{matrix} \quad (2)$$

$$O_{mn}^l = \mathrm{F}(net_{mn}^l) = sigmoid(net_{mn}^l) = \frac{1}{1+e^{-net_{mn}^l}} \quad (3)$$

$$Y_{ij} = \mathrm{pool}(x,i,j) = \frac{\sum_{m=1}^{size^l}\sum_{n=1}^{size^l} x_{size^l\times(i-1)+m,size^1\times(j-1)+n}^{l-1}}{size^l \times size^l} \quad (4)$$

$$\mathrm{pool}(x) = \begin{matrix} \frac{x_{11}+x_{12}+x_{21}+x_{22}}{4} & \frac{x_{13}+x_{14}+x_{23}+x_{24}}{4} \\ \frac{x_{31}+x_{32}+x_{41}+x_{42}}{4} & \frac{x_{33}+x_{34}+x_{43}+x_{44}}{4} \end{matrix} \quad (5)$$

$$O_i^{-2} = O_{mn}^{-3}, m = int\left(\frac{i}{size^{-2}}\right) + 1, n = i - size^{-2} \times (m-1) \quad (6)$$

$$net_j^{-1} = \sum_{i=1}^{size^{-2}} \left(O_i^{-2} \cdot w_{ij}^{-1} + b^{-1}\right), j = 1,2,\ldots, size^{-1} \quad (7)$$

$$O_j^{-1} = \mathrm{F}\left(net_j^{-1}\right) = \mathrm{sigmoid}\left(net_j^{-1}\right) = \frac{1}{1+e^{-net_i^{-1}}} \quad (8)$$

$$\hat{y}_n = o^{-1} \quad (9)$$

end for

L(t) is re-calculated as ***L(t)*** $=\frac{1}{2}\sum_{n=1}^{N}\left(train_{p(n)} train_{y(n)}\right)^2$, N is the total number of training Set

$\Delta w^l, \Delta b^l, \Delta w_{ij}^{-1}, \Delta b^{-1}$ are updated according to the formula (20-23*)* as bellow:

$$\Delta w^l = \frac{\partial L}{\partial w^l} = \frac{\partial L}{\partial net^l} \times \frac{\partial net^l}{\partial w^l} = \delta^l . O^{l-1} \quad (20)$$

$$\Delta b^l = \frac{\partial L}{\partial b^l} = \frac{\partial L}{\partial net^l} \times \frac{\partial net^l}{\partial b^l} = \delta^l \quad (21)$$

$$\Delta w_{ij}^{-1} = \frac{\partial L}{\partial w_{ij}^{-1}} = \frac{\partial L}{\partial net_j^{-1}} \times \frac{\partial net_j^{-1}}{\partial w_{ij}^{-1}} = \delta_j^{-1} . O_i^{-2} \quad (22)$$

$$\Delta b^{-1} = \frac{\partial L}{\partial b_j^{-1}} = \frac{\partial L}{\partial net_j^{-1}} \times \frac{\partial net_j^{-1}}{\partial b_j^{-1}} = \frac{1}{size^l} \sum_{j=1}^{size^{-1}} \delta_j^{-1} \quad (23)$$

$w^l(t), b^l(t), w_{ij}^{-1}(t), b_j^{-1}(t)$ are adjusted according to the formula (24-27) as bellow:

$$w^l(t+1) = w^l(t) - \eta\_\mathrm{CPNN} \times \Delta w^l \quad (24)$$

$$b^l(t+1) = b^l(t) - \eta\_\mathrm{CPNN} \times \Delta b^l \quad (25)$$

$$w_{ij}^{-1}(t+1) = w_{ij}^{-1}(t) - \eta\_\mathrm{CPNN} \times \Delta w_{ij}^{-1} \quad (26)$$

$$b^{-1}(t) = b^{-1}(t) - \eta\_\mathrm{CPNN} \times \Delta b_j^{-1} \quad (27)$$

t++

end while

End

Figure 11: The figure shows the pseudocode of CNN optimized Immunity system algorithm. (Cont.).

	precision	recall	F1-score	support
0	1.00	1.00	1.00	1
1	0.91	1.00	0.95	20
2	1.00	0.94	0.97	18
3	0.95	0.91	0.93	22
accuracy			0.95	61
macro avg	0.97	0.96	0.96	61
weighted avg	0.95	0.95	0.95	61

Confusion Matrix:
[[1 0 0 0]
[1 20 0 0]
[0 0 17 1]
[0 2 0 20]]

Accuracy: 0.95082

Figure 12: CNN's confusion matrix and accuracy by testing cow estrus dataset.

Figure 13: The image shows the results obtained from the prediction using the CNN, the accuracy is 95.082%.

	precision	recall	F1-score	support
0	1.00	1.00	1.00	1
1	1.00	0.95	0.97	20
2	1.00	1.00	1.00	18
3	0.96	1.00	0.98	22
accuracy			0.98	61
macro avg	0.97	0.99	0.99	61
weighted avg	0.95	0.98	0.98	61

Confusion Matrix:

```
[[1  0   0 0]
 [0 19  0 1]
 [0  0 18 1]
 [0  2  0 22]]
```

Accuracy: 0.98361

Figure 14: The confusion matrix and accuracy of CNN optimized with Immunity System by testing cow estrus dataset.

Figure 15: The image shows the results obtained from the prediction using the CNN optimized with Immunity System, the accuracy is 98.361%.

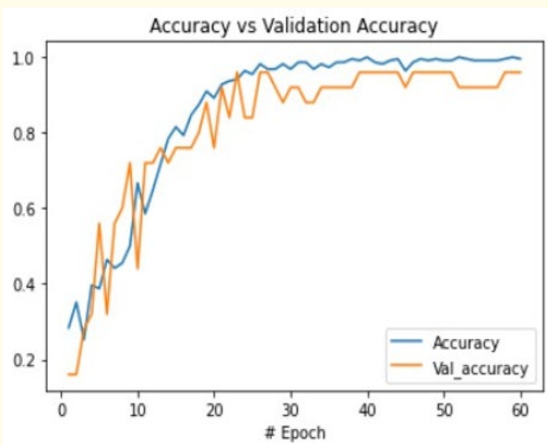

Figure 16: The accuracy tends to increase between accuracy and validation accuracy.

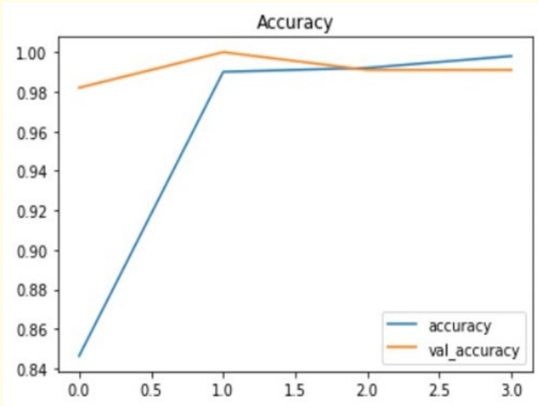

Figure 17: The accuracy graph shows the accuracy and validation accuracy of CNN optimized by AIS.

From the results of using the AIS algorithm to adjust the hyperparameter, the removal value was 98.361%, the optimum value. Searching and comparing with other methods was found to be better or equivalent to different techniques as shown in the values. The accuracy of other research is in Table 1 as shown.

Author Name, Year	***Accuracy (%)***
Namrata Karlupia and et. al. (2023) [21]	94.50
Afia Zafar and et. al. (2023) [22]	95.50
Kalaiarasi P and et. al. (2021) [23]	97.30
Khadija Aguerchi and et. al. (2024) [24]	98.23
Darunee Watnakornbuncha and et. al. [25]	89.83
Watchara Ninphet and et. al.	98.36

Table 1: The table displays the accuracy of image prediction of estrus in cows using CNN with hyperparameter adjustment with AIS compared to other methods.

After achieving a 98.36% accuracy in predicting estrus in dairy cows, the researcher pursued further research using deep learning and CNN to detect estrus in cows. They aimed to enable dairy farmers to leverage AI technology for more efficient work processes. The deep learning technique employed was YOLOv5, which showcased an 89.80% accuracy using default settings and underwent training with several cycles of 30 epochs. After experimenting with hyperparameters, a batch size of 8, an image size of 640, and training with 40 epochs yielded the most suitable results, with a test output accuracy of nearly 99.50%. The researcher utilized a Lenovo workstation with an Intel Core i7 10th Gen GPU, 2.60 GHz speed, and 32 GB RAM, running on Microsoft Windows 11 Pro. The total processing time for the 40 epochs was 2.279 hours, and the optimizer resulted in a 14.5MB-sized output. So, the researcher has presented all processing results in the order shown in Figure 18-33.

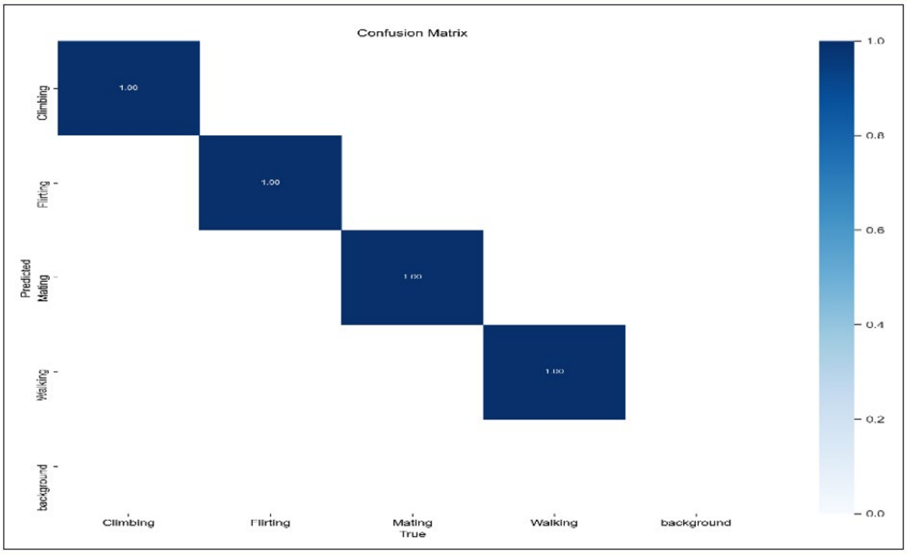

Figure 18: YOLOv5's confusion matrix and accuracy by testing cow estrus dataset for detection.

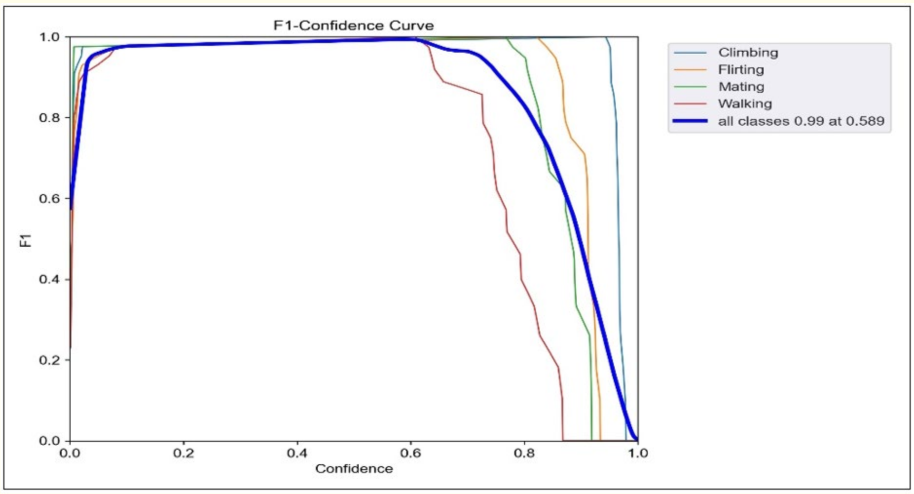

Figure 19: Shows graph value of F1-Score for detecting estrus in cows with all 4 classes of YOLOv5.

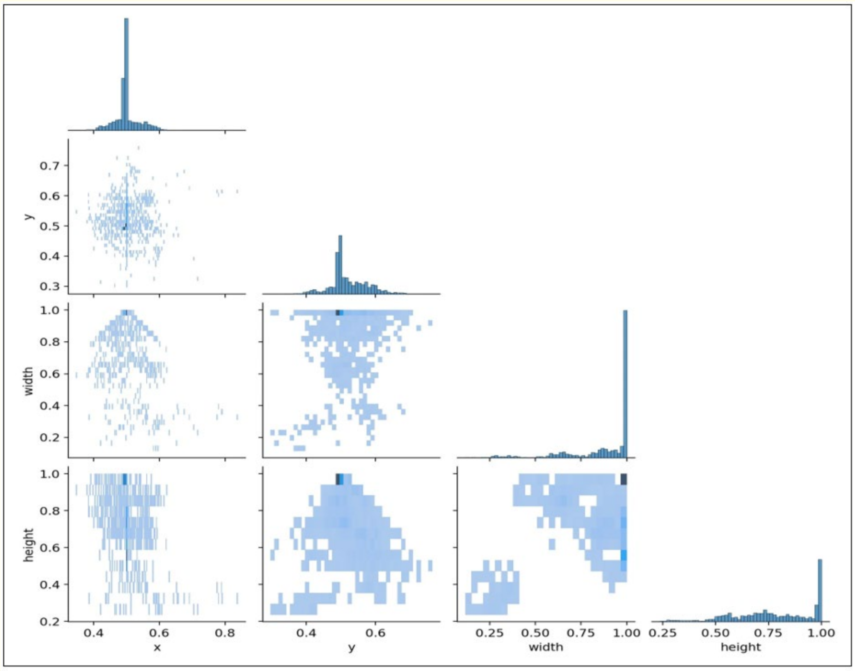

Figure 20: The graph shows correlation statistics, which evaluate the random nature of the four variables.

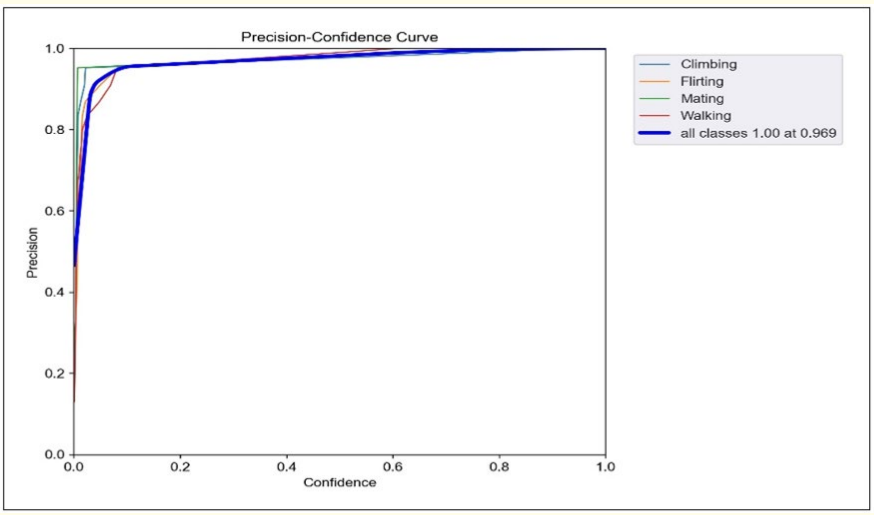

Figure 21: Shows graph value of Precision for detecting estrus in cows with all 4 classes of YOLOv5.

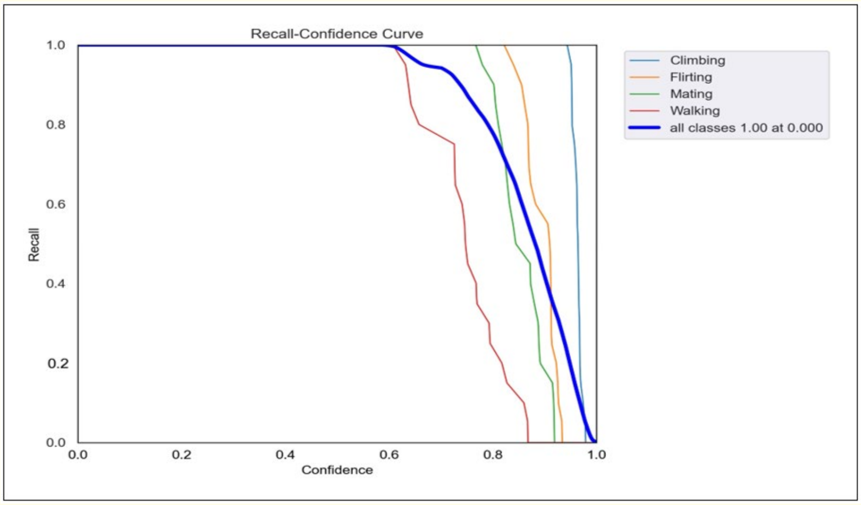

Figure 22: Shows graph value of Recall for detecting estrus in cows with all 4 classes of YOLOv5.

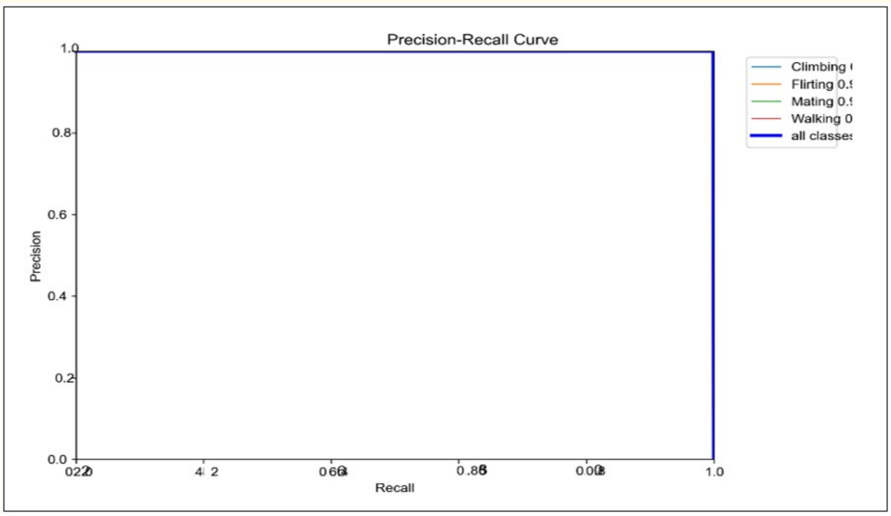

Figure 23: Shows graph value of Precision and Recall for detecting estrus in cows with all 4 classes of YOLOv5.

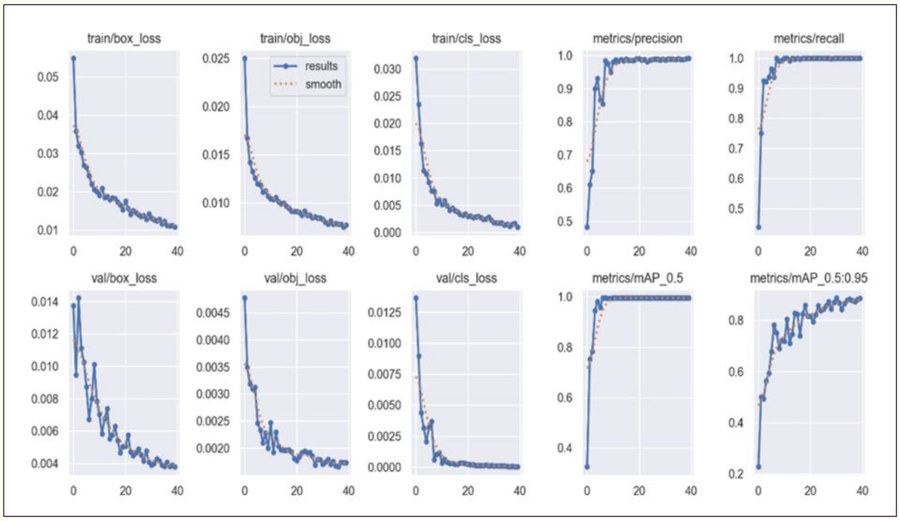

Figure 24: Shows graph value of Loss Function for detecting estrus in cows with all 4 classes of YOLOv5.

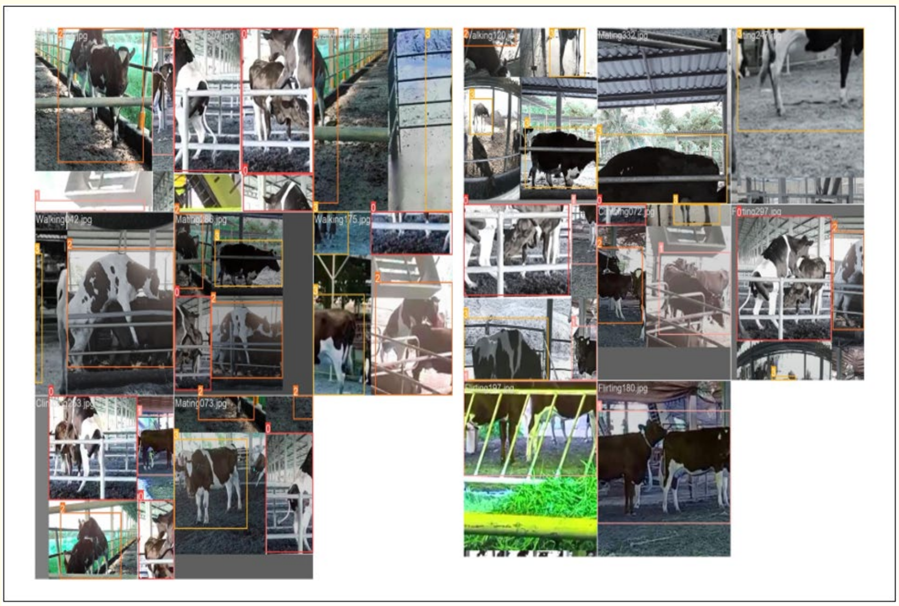

Figure 25: The image shows an example of YOLOv5 training in all 4 classes (Climbing, Flirting, Mating, and Walking).

Figure 26: The image shows an example of YOLOv5 training in all 4 classes (Climbing, Flirting, Mating, and Walking).

Figure 27: The image shows an example of YOLOv5 prediction in all 4 classes (Climbing, Flirting, Mating, and Walking).

Figure 28: The image shows an example of YOLOv5 prediction in all 4 classes (Climbing, Flirting, Mating, and Walking).

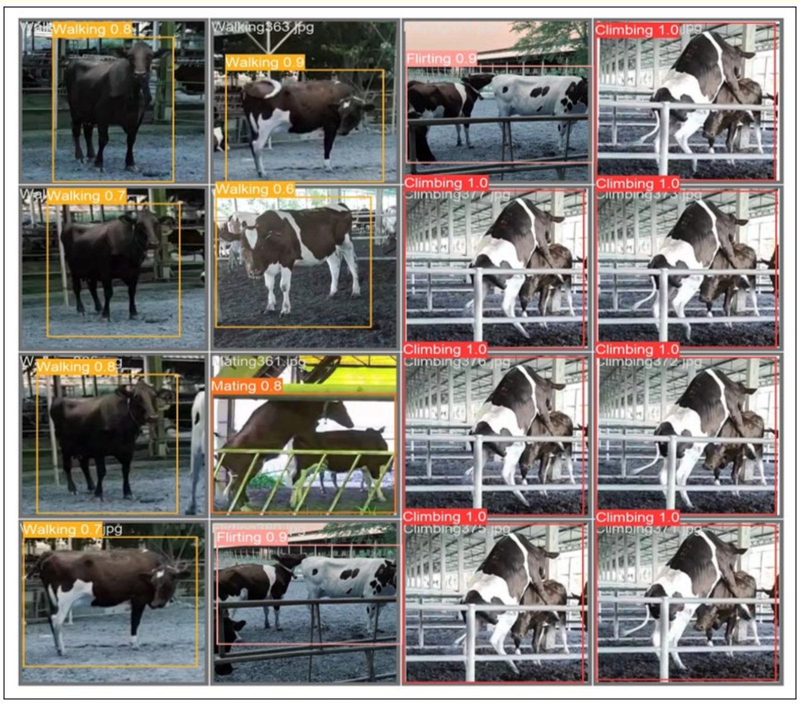

Figure 29: The image shows an example of YOLOv5 prediction in all 4 classes (Climbing, Flirting, Mating, and Walking).

epoch	train/ box_loss	train/ obj_loss	train/ cls_loss	metrics/ precision	metrics/ recall	metrics/ mAP_0.5	metrics/ mAP_0.5:0.95	val/ box_loss	val/ obj_loss	val/ cls_loss	x/lr0	x/lr1	x/lr2
0	0.054817	0.024973	0.031935	0.482410	0.437500	0.325270	0.227070	0.013747	0.004769	0.013652	0.070167	0.003315	0.003315
1	0.035838	0.016715	0.023476	0.610820	0.750770	0.751510	0.499650	0.009457	0.003486	0.008957	0.040002	0.006484	0.006484
2	0.031912	0.014161	0.016309	0.651630	0.925000	0.783660	0.493680	0.014212	0.003186	0.004370	0.009673	0.009487	0.009487
3	0.030215	0.013254	0.011294	0.900540	0.922670	0.945390	0.562400	0.011121	0.003094	0.003146	0.009258	0.009258	0.009258
4	0.026935	0.012543	0.010745	0.931430	0.937350	0.981390	0.593100	0.010267	0.003128	0.002047	0.009258	0.009258	0.009258
5	0.026344	0.011954	0.009245	0.865340	0.964510	0.958070	0.678330	0.008739	0.002463	0.003181	0.009010	0.009010	0.009010
6	0.024166	0.011822	0.007618	0.853640	0.937050	0.995000	0.781330	0.006736	0.002336	0.003673	0.008763	0.008763	0.008763
7	0.021974	0.011108	0.007523	0.984750	1.000000	0.995000	0.750880	0.007996	0.002104	0.000608	0.008515	0.008515	0.008515
8	0.020547	0.011230	0.005292	0.975160	0.990290	0.994400	0.689330	0.010115	0.002286	0.001015	0.008268	0.008268	0.008268
9	0.019998	0.010667	0.005946	0.949480	0.993200	0.994400	0.723910	0.007838	0.002000	0.001175	0.008020	0.008020	0.008020
10	0.019024	0.010397	0.005093	0.980760	1.000000	0.995000	0.718390	0.007017	0.002475	0.000340	0.007773	0.007773	0.007773
11	0.020914	0.010328	0.005772	0.986660	1.000000	0.995000	0.804500	0.005837	0.001925	0.000625	0.007525	0.007525	0.007525
12	0.018472	0.010556	0.004920	0.983290	0.992120	0.995000	0.710750	0.006784	0.002299	0.000393	0.007278	0.007278	0.007278
13	0.018736	0.010117	0.004091	0.987780	1.000000	0.995000	0.746100	0.007390	0.002039	0.000287	0.007030	0.007030	0.007030
14	0.017974	0.009923	0.004485	0.983870	0.997230	0.995000	0.827800	0.005517	0.001974	0.000293	0.006783	0.006783	0.006783
15	0.018462	0.009944	0.004016	0.989800	1.000000	0.995000	0.823140	0.005759	0.001962	0.000224	0.006535	0.006535	0.006535
16	0.018200	0.009632	0.003792	0.985350	0.997330	0.995000	0.739470	0.006295	0.001967	0.000217	0.006288	0.006288	0.006288
17	0.017336	0.009414	0.003324	0.985930	1.000000	0.995000	0.825200	0.005416	0.001976	0.000342	0.006040	0.006040	0.006040
18	0.016618	0.009149	0.003163	0.985620	1.000000	0.995000	0.857820	0.004667	0.001937	0.000315	0.005793	0.005793	0.005793
19	0.015331	0.009109	0.003406	0.990220	1.000000	0.995000	0.815980	0.005038	0.001813	0.000303	0.005545	0.005545	0.005545
20	0.017546	0.009111	0.002921	0.990400	1.000000	0.995000	0.815270	0.005072	0.001777	0.000242	0.005298	0.005298	0.005298
21	0.015652	0.009000	0.002994	0.986770	1.000000	0.995000	0.793980	0.005745	0.001845	0.000199	0.005050	0.005050	0.005050
22	0.014106	0.008714	0.002702	0.987870	1.000000	0.995000	0.822110	0.004708	0.001916	0.000179	0.004803	0.004803	0.004803
23	0.015116	0.009136	0.002885	0.981420	1.000000	0.995000	0.857200	0.004497	0.001953	0.000140	0.004555	0.004555	0.004555
24	0.014556	0.008708	0.002972	0.985960	1.000000	0.995000	0.837120	0.004667	0.001919	0.000144	0.004308	0.004308	0.004308
25	0.014037	0.008718	0.002696	0.988640	1.000000	0.995000	0.844860	0.004899	0.001930	0.000140	0.004060	0.004060	0.004060
26	0.013528	0.008465	0.002411	0.989070	1.000000	0.995000	0.857100	0.004535	0.001859	0.000119	0.003813	0.003813	0.003813
27	0.013781	0.008531	0.002506	0.989790	1.000000	0.995000	0.873260	0.004157	0.001693	0.000148	0.003565	0.003565	0.003565
28	0.012789	0.008423	0.002816	0.986860	0.998770	0.995000	0.843460	0.004768	0.001797	0.000086	0.003318	0.003318	0.003318
29	0.014192	0.008395	0.002260	0.989510	1.000000	0.995000	0.869600	0.004076	0.001791	0.000140	0.003070	0.003070	0.003070
30	0.013164	0.008277	0.001765	0.987160	1.000000	0.995000	0.887100	0.003891	0.001710	0.000092	0.002823	0.002823	0.002823
31	0.012759	0.007986	0.001718	0.989570	1.000000	0.995000	0.867500	0.003993	0.001747	0.000102	0.002575	0.002575	0.002575
32	0.012346	0.007755	0.001709	0.989430	1.000000	0.995000	0.843270	0.004286	0.001799	0.000110	0.002328	0.002328	0.002328
33	0.012857	0.008110	0.001720	0.989920	1.000000	0.995000	0.865050	0.004149	0.001705	0.000091	0.002080	0.002080	0.002080
34	0.011662	0.007775	0.001341	0.987850	1.000000	0.995000	0.878710	0.003895	0.001759	0.000082	0.001833	0.001833	0.001833
35	0.012321	0.007843	0.001465	0.990360	1.000000	0.995000	0.883010	0.003824	0.001677	0.000056	0.001585	0.001585	0.001585
36	0.011208	0.007784	0.001101	0.988770	1.000000	0.995000	0.877860	0.004083	0.001661	0.000043	0.001338	0.001338	0.001338
37	0.011135	0.007777	0.001537	0.988190	0.999740	0.995000	0.873460	0.003835	0.001743	0.000045	0.001090	0.001090	0.001090
38	0.011208	0.007465	0.001640	0.991010	1.000000	0.995000	0.881910	0.003892	0.001740	0.000030	0.000843	0.000843	0.000843
39	0.010769	0.007676	0.000962	0.991530	1.000000	0.995000	0.886680	0.003798	0.001740	0.000024	0.000595	0.000595	0.000595

Table 2: Shows a summary of the results of calculating precision, recall, map@k, and loss values for detecting estrus in cows with YOLOv5 for all 4 classes (climbing, flirting, mating and walking).

Conclusion

Smart Agriculture or Smart Farming is a modern approach to agriculture that replaces traditional methods with data and technology for managing agricultural operations. It aims to develop sustainable agricultural practices, reduce unnecessary costs, and increase profits. Both the government and the private sector need to pay attention to this because agriculture is a crucial indicator of Thailand's economic development and is the country's primary source of employment. Despite its importance, farmers' incomes are disproportionately low, with agricultural income typically contributing less than 50% of a household's total income. This reflects underlying issues, including challenges in managing production and high costs, which can be addressed by adopting Smart Agriculture practices.

In this research, we utilized artificial intelligence to improve the management of dairy farms, transforming them into Smart Farms. We optimized the hyperparameter values of the CNN to enhance its predictive capabilities. The results demonstrated that by using the AIS algorithm to adjust CNN parameters, we significantly improved accuracy. Specifically, the average accuracy using expert-determined values was 95.082%, whereas the AIS algorithm improved to 98.361%. Graphs Figure. 16 and Figure. 17 show that accuracy tends to increase, and the loss function indicates a good fit learning curve, signifying effective model learning. This model can accurately predict unseen data and generalize well to new data with minimal error. Moreover, this research discusses using deep learning to make farms "Smart". The researchers used deep learning to detect estrus in cows. Estrus in cows is a brief and unpredictable period, making it difficult for farmers to monitor, especially at night. The researchers propose using CCTV cameras to detect and monitor cow behaviors using YOLO-type deep learning. The study found that YOLOV5 can effectively detect estrus, as evidenced by the calculated and predicted values in the table. For instance, the test's F1-Score averages= 0.993, with Precision = 0.987, Recall =1.00, mAP50 = 0.995, mAP50-95= 0.887, and ACC = 0.995.

When observing Figure 24, it's evident that the learning and predicted values form almost the same line. This indicates that the data preparation, determination of work cycles, and configuration of various parameters are consistent. The loss curve demonstrates a good fit, showing that learning and prediction can be done efficiently. The results displayed in Figure 27-29 indicate that predictions can be accurately detected with very few errors. Based on this research, it is recommended to introduce artificial intelligence technology to help manage modern farms. If farmers choose suitable technology at an affordable price, it will enable them to reduce production costs and effectively increase productivity, making it a worthwhile investment. In future research, the detection methods used in this study can be combined with monitoring other cow behaviors such as illness and birth, which previously relied on human observation. Furthermore, advancements in detection techniques have led to the development of multiple versions of YOLO deep learning. Therefore, future researchers are encouraged to utilize the latest version of YOLO deep learning, which will streamline data preparation compared to the previous version. This will improve the efficiency of researchers, allowing them to spend less time on data preparation and more time on their actual work.

References

1. ETDA. "Artificial intelligence in government services". I knowledge Sharing (2024).
2. SAS. "What is deep learning? and how important is it?". Deep Learning (2024).
3. TECHSAUCE. "Get to know ai, machine learning, deep learning in an easy to understand version". Knowledge Sharing Platform (2024).
4. FE Streib., et al. "An introductory review of deep learning for prediction models with big data". Frontiers in Artificial Intelligence 33 (2020): 1-23.
5. M Fintz, M Osadchy and U Hertz. "Using deep learning to predict human decisions and using cognitive models to explain deep learning models". Scientific Report 12.1 (2022): 4736.
6. R Arablouei., et al. "Animal behavior classification via deep learning on embedded systems". Computers and Electronics in Agriculture 207 (2023): 1-14.
7. SS Khan., et al. "Hybrid deep learning approach for enhanced animal breed classification and prediction". International Information and Engineering Technology Association 40.5 (2023): 2087-2099.

8. T. encyclopedia for youth. "Rural agricultural development/consumption and dairy industry". Thai Junior Encyclopedia Foundation by His Majesty King Bhumibol Adulyadej the Great 12.5 (2024): 1-5.
9. DDP. "World dairy situation and thai 2019". World and Thai dairy cows in 2019 raise farm standards to keep up with disease (2019).
10. KSD Cooperative. "What is the knowledge about dairy farming?". Knowledge (2024).
11. D. of Livestock. Beef cattle health care. Bangkok Thailand: Livestock Development and Technology Transfer Office Department of Livestock Development 1 (2006).
12. VV Andrade., et al. "Estrus prediction models for dairy gyr heifers". Animals (Basel) 11.11 (2021): 3103.
13. W Ninphet, N Amm-Dee and A Sangsongfa. "Prediction cows estrus images using convolutional neural network with optimized parameters by the artificial immune system algorithm." Proceedings of the 20th International Conference on Computing and Information Technology (IC2IT 2024). Springer Link 973 (2024): 105-120.
14. R Miura., et al. "Prediction of the spontaneous estrus expression period based on large (≥ 10 mm) follicle numbers in lactating holstein dairy cows". Veterinary Sciences, MDPI 10.3 (2023): 231.
15. AK Yildiz and MM Ozguven. "Determination of estrus in cattle with artificial neural networks using mobility and environmental data". Journal of Agricultural Faculty of Gaziosmanpasa University 39.1 (2022): 40-45.
16. J Wang., et al. "Using machine-learning technique for estrus onset detection in dairy cows from acceleration and location data acquired by a neck-tag". Biosystems Engineering 214 (2022): 193-206.
17. I Kunakorntum. "Image detection using YOLOv5". Medium (2021).
18. E Zvornicanin. "What is yolo algorithm?" Baeldung (2024).
19. S Kanoktipsatharpron. "What is the convolutional neural network ?". BUA Lab (2019).
20. Z Kelta. "An introduction to convolutional neural networks (cnns)," Datacamp (2023).
21. N Karlupia., et al. "A genetic algorithm-based optimized convolutional neural network for face recognition". Int. J. Appl. Math. Comput 33.1 (2023): 21-31.
22. A Zafar., et al. "An optimization approach for convolutional neural network using non-dominated sorted genetic algorithm-ii". Computers Materials and Continua 74.3 (2023): 5641-5659.
23. Kalaiarasi P., et al. "Optimizing convolutional neural network using particle swarm optimization for face recognition". Turkish Journal of Computer and Mathematics Education 12.11 (2021): 3672-3679.
24. K Aguerchi., et al. "A cnn hyperparameters optimization based on particle swarm optimizationfor mammography breast cancer classification." Journal of Imaging 10.2.30 (2024): 1-17
25. D Watnakornbuncha, N AM-DEE and A Sangsongfa. "Adaptive deep learning with optimization hybrid convolutional neural network and recurrent neural network for prediction lemon fruit ripeness." Przeglad Elektrotechniczny 2024.3 (2024): 202-211.