

Automated Estrus Detection for Dairy Cattle through Neural Networks and Bounding Box Corner Analysis

Nilo M. Arago¹, Chris I. Alvarez², Angelita G. Mabale³, Charl G. Legista⁴, Nicole E. Repiso⁵
Rodney Rafael A. Robles⁶, Timothy M. Amado⁷, Romeo Jr. L. Jorda⁸, August C. Thio-ac⁹
Jessica S. Velasco¹⁰, Lean Karlo S. Tolentino¹¹

Department of Electronics Engineering, Technological University of the Philippines^{1, 2, 3, 4 5, 6, 7, 8, 9, 10, 11}
University Extension Services Office, Technological University of the Philippines¹¹
Ermita, Manila 1000, Philippines^{1, 2, 3, 4 5, 6, 7, 8, 9, 10, 11}

Abstract—Thorough and precise estrus detection plays a crucial role in the fertility of dairy cows. Farmers commonly used direct visual monitoring in recognizing estrus signs which demands time and effort and causes misinterpretations. The primary sign of estrus is the standing heat, where the dairy cows stand to be mounted by other cows for a few seconds. Through the years, researchers developed various detection methods, yet most of these methods involve contact and invasive approaches that affect the estrus behaviors of cows. So, the proponents developed a non-invasive and non-contact estrus detection system using image processing to detect standing heat behaviors. Through the TensorFlow Object Detection API, the proponents trained two custom neural network models capable of visualizing bounding boxes of the predicted cow objects on image frames. The proponents also developed an object overlapping algorithm that utilizes the bounding box corners to detect estrus activities. Based on the conducted tests, an estrus event occurs when the centroids of the detected objects measure a distance of less than 360px and have two interior angles with another fixed point of less than 25° and greater than 65° for Y and X axes, respectively. If the conditions are met, the program will save the image frame and will declare an estrus activity. Otherwise, it will restart its estrus detection and counting. The system observed 17 cows, a carabao, and a bull through the cameras installed atop of a cowshed, and detects the estrus events with an efficiency of 50%.

Keywords—Dairy cows; estrus detection; image processing; TensorFlow Object Detection API; custom neural network; object overlapping

I. INTRODUCTION

The estrus cycle of mammals, such as dairy cattle and water buffaloes, is the period from one estrus to the next. On a typical basis, the cycle has an average duration of 21 days. In the Philippines, farmers observe a period of between 18 and 24 days. Research shows that estrus usually lasts between 10 and 18 hours. Even so, recent studies show that modern dairy cows' cycles are about 8 hours shorter [1] [2]. A livestock requires thorough heat detection, and correct timing of artificial insemination. So, not being able to detect in-heat signatures of cattle may lead to low fertility. If the producers could not detect and differ the in-heat and non-heat signs of the cattle, the farm may suffer. Also, the extended calving intervals and semen expenses affect the farm's economic status.

Farmers and researchers have introduced various methods to determine in-heat signatures in livestock. Today, farmers

commonly use visual observation of estrus signs of cows. But doing so may lead to misinterpretations as well. Meanwhile, some farmers track the roaming activities of the cows through a motion sensor on the cows' neck or leg. This method still varies depending on the efficiency and accuracy of the devices [3].

Several companies in America and Europe developed electronic products and services such as the AfiACT, the HeatWatch system, the MountCount, etc. to identify the cows' estrus behaviors [4]. But in Asia, there are few companies known to offer such products. And in the Philippines, companies offering these types of products and services are non-existent. These show how underdeveloped the cattle industry in the Philippines is. According to the Philippine Statistics Authority, the fourth reading of the total cattle production in 2018 is 0.33 percent lower than in 2017. The stock of cattle is also decreased by 0.73 percent, and the rate of slaughter is high and rising [5]. These statistics proved that the Philippines' performance in cattle production is slower than in other ASEAN countries. That is why farmers and researchers should develop new methods to meet the demands of the country.

As a solution to the problem, in this paper, the researchers proposed a non-invasive and non-contact estrus detection system that uses image processing and artificial intelligence through TensorFlow Object detection API to identify standing heat behaviors of Holstein-Friesian and Sahiwal crosses. The research specifically aims to: (1) develop an automated estrus detection system which visualizes bounding boxes of the cattle objects, and verifies if the overlapping instances are estrus activities through the surveillance system; and (2) conduct an evaluation and assessment on the system's functionality and reliability of detection in comparison with the manual visual inspection methods of the farmers.

The findings of the study will benefit small and large farms in the cattle industry, given the current lack of commercially available products and services, and advanced breeding methods. The implementation of the estrus detection system minimizes the workload of farmers through the real-time monitoring capabilities of the system and increases the dairy production and fertility rate of cows through immediate insemination. Such benefits consequently contribute to the economic growth of the farms.

This research paper is structured as follows: Section II pertains to the gaps and limitations of the related researches, Section III defines the materials and methods used by the researchers, Section IV explains the detection and database results of the study, Section V declares the conclusion and Section VI enumerates possible future works of the research.

II. RELATED WORKS

Researchers develop high-tech devices that helps farmers track the estrus signs of cows. Such technologies based its efficiency on the detection of physical activities, mounting behaviors, body temperature, etc. [6].

In [7], the researchers developed an estrus detection system based on the following behavior of the cows for a short time using IP cameras. The system implements a motion detection technique to identify probable mounting regions, and blob analysis on the said regions to detect changes on the image frames. By incorporating both methods, the proponents were able to accurately identify true estrus events on the surveillance feed.

Talukder *et al.* tested the effectivity of implementing infrared thermography (IRT) in detecting estrus behaviors of dairy cattle. The proponents also incorporated a breeding indicator with IRT which resulted in a sensitive heat detector with false-positive results. The technology can only yield true estrus events only when the IRT was implemented during the ovulation phase of the subjects [8].

In [9], the researchers devised a cattle identifier based on Region Based Convolutional Neural Networks (R-CNN) in an open field setup using unmanned aerial vehicles (UAV) drones. The study has shown great results in detecting unique individual cow patterns through deep learning frameworks and end-to-end training of image datasets. However, false-positive results still occur due to the similarity of structures and features of some cows.

Yang *et al.* also proposed an estrus detection system based on the following and restlessness behaviors of the cows using infrared technology. The infrared cameras were able to monitor and detect estrus events at both daytime and nighttime with the aid of artificial lighting. Despite that, their experimentations showed that the efficiency for detecting objects was greater in contrast to the visual observation considering good illumination in the area [10].

Meanwhile, Xia *et al.* constructed an estrus detection system based on the activities of the cows using pedometers and readers. Through the pedometers and the readers, the system was able to gather and analyze cow information to declare estrus and notify the end-users via text messages. The results proved the system's accuracy, in which it can replace the conventional rectum identification of cows in detecting estrus [11].

In [12], the researchers also proposed an estrus detection system through geometric region analysis using fixed IP cameras. This system's operability is similar to the aforementioned studies that filter the collected image frames and extracts the relevant features of the cows from the images to perform analysis and identification of estrus. Still, the

proposed techniques in this research accurately recognized the mounting behaviors of the cows with minimal false-positive detection rates.

Table I shows the comparison framework of the related works in this research. Unlike with the aforementioned studies, this research performs estrus detection by detecting Holstein-Friesian and Sahiwal Crosses, a bull, and a water buffalo from the surveillance feed of three pan-tilt-zoom (PTZ) cameras (DH-SD22404T-GN Lite Series, 4 MP). The researchers also customized two neural network models using pre-trained frameworks from the TensorFlow Zoo for the object detection and utilized bounding box corners for the analysis of overlapping instances in the image sequences and declaration of estrus events.

TABLE I. BRIEF COMPARISON FRAMEWORK OF THE RELATED WORKS

Authors	Breed of Cattle to be monitored	Materials and Methods	
		Sensors used	Techniques and Algorithms used
Tsai and Huang (2014) [7]	Holstein	IP Dome Camera	Motion Detection, Region Segmentation, Foreground Segmentation, and Blob Analysis
Talukder <i>et al.</i> (2014) [8]	Holstein-Friesian	Thermal Infrared Camera	Infrared Thermography
Andrew, Greatwood, and Burghardt (2017) [9]	Holstein-Friesian	UAV integrated camera	R-CNN Localization, and Tracking
Yang, Lin, and Peng (2017) [10]	Holstein	Infrared Camera	Motion Detection, Region Segmentation, Foreground Segmentation.
Xia <i>et al.</i> (2017) [12]	Holstein	Pedometer and reader	Motion Analysis
Guo, Zhang, He, Niu, and Tan (2019) [13]	Holstein	Fixed IP Camera	Background Subtraction with Color and Texture Features (BSCTF), Geometric and Optical Flow feature extraction, Support Vector Machine (SVM)

III. METHODOLOGY

A. Research Locale - Barn

In this research, the estrus detection system is deployed in a small-scale commercial farm in the province of San Ildefonso, Bulacan, in the Philippines. The barn houses 17 Holstein-Friesian and Sahiwal crosses, a bull, and a water buffalo. Similarly with the research of Porto *et al.* [13], they have observed some delimiting factors in the barn that may affect the automated detection system, such as: high variation in illumination in areas near the open side of the barn; metal surfaces of stable crossbars; color indifferences of cows; and surface reflection caused by manure or dirt. The panoramic top-viewed images of the barn are crucial in to capture image frames which shows the true shape of cow's body [13]. To

capture the panoramic top-view images, three 4 Megapixel Pan-tilt-zoom (PTZ) Network Cameras, as in [13], were installed at a height of 3.78m. Each camera monitors an area for about 4.87 m x 3.97 m with a separation distance of approximately 2.98 m apart atop the cowshed, as shown in Fig. 1 and Fig. 2.

B. TensorFlow Object Detection API

TensorFlow Object Detection API is a framework that is currently being utilized today to resolve object detection problems. With this, deploying accurate machine learning models that can localize and identify multiple objects in an image frame is easier, as in. Within the models, the feature extraction and the classification processes play vital roles in the cow pattern recognition, as in [14].

According to Huang *et al.*, there will be trade-offs between speed and accuracy in constructing an object detection architecture that depends on the application and platform [15]. In their repository, the user can modify the model to satisfy his/her requirements and platform. The TensorFlow Object Detection API library comprises of object detection structures, such as Single Shot Detector (SSD), Faster Region-based Convolutional Neural Network (Faster R-CNN), etc.

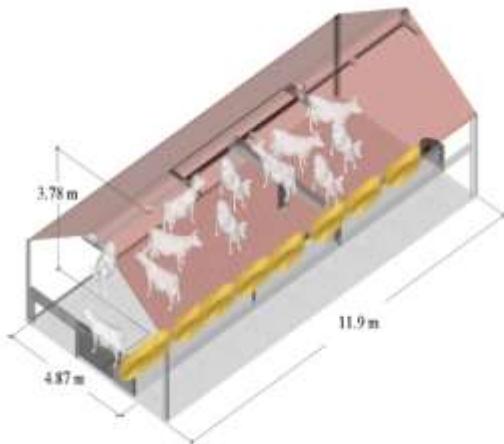


Fig. 1. The Isometric view of the Experimental Setup which Displays the Position of the Cameras.

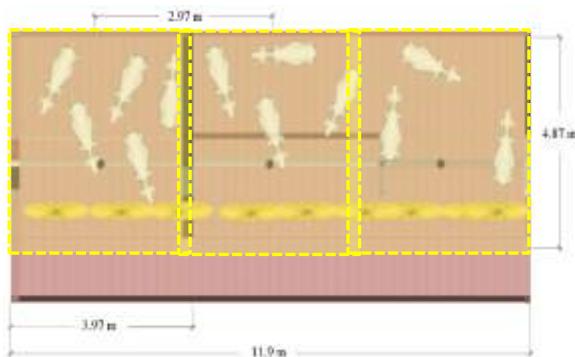


Fig. 2. The Bottom view of the Experimental Setup which Depicts the Camera Separation and the Field of views.

Feature extractors such as Inception, MobileNet [16] and Resnet play critical roles in the speed and accuracy trade-off of

the framework. Even with the recent studies of various researchers, constructing convolutional networks from scratch requires a great volume of image datasets and a long period of training and testing time. That is why transfer learning is more applicable with pre-trained models like the TensorFlow API [17]. Transfer learning is a technique in which a model is reprocessed as a starting point for a second function model [18] [19].

In this research, two (custom) object detection frameworks using TensorFlow CPU and the pre-trained Faster R-CNN [20] and SSD [21] models were developed and integrated as its core architectures from the TensorFlow Zoo.

C. Data Acquisition and Pre-Processing

In this research, all of the cows, including the bull and the water buffalo, are pre-identified with a corresponding ID. In building the dataset, a total of 1400 images for each defining class for the Faster R-CNN model, and a total of 21,912 images of cows for the SSD model were used. By accessing the playback videos from the Network Video Recorder, and using image processing techniques through OpenCV, the image frames were obtained at a rate of 1 frame per second.

To provide the necessary supervised learning for the detection system, the researchers used a label annotator tool, as in [19]. For the Faster R-CNN model, each cow object on every image frame were annotated as: "BULL"; "CARACOW"; "COW A"; "COW B"; "COW C"; "COW D"; "COW E"; "COW F"; "COW G"; "COW H"; "COW I"; "COW J"; "COW K"; "COW L"; "COW M"; "COW N"; "COW O"; "COW P"; and "COW Q" in accordance to its COW ID whereas, for the SSD model, all objects were labeled as "COW". The annotations will be saved as Extensible Markup Language data files (XML) and will be processed after the data slicing. Next, the image datasets were divided into the training and the testing data. The partition used for data slicing is 90:10 wherein 90% is for the training data while the 10% is for the testing data, as in [9] [17] [22] [23].

Afterwards, two label maps for each model were created, in which 19 labels were listed for the Faster R-CNN model but only 1 label for the SSD model. From the XML data files, TensorFlow Records in "RECORD" format will be generated. These records contain the filename, the labels (classes), the height and width of the images, and the bounding box corners (xmin, ymin, xmax, and ymax), as in [9] [24].

D. Configuring the Pipeline

In selecting a pre-trained model, the performance, speed, and mean Average Precision (mAP) that define the accuracy of the detector were considered, as in [16] [18]. According to the analysis of Huang *et al.* [15], the Faster R-CNN model with Inception V2 and SSD model with Inception V2 yields a mAP of 28 and 24, respectively, which requires a speed of at least 58 ms and 42 ms per image, respectively. To configure the pipeline, the researchers utilized two of the pre-trained models provided by TensorFlow Zoo. The speed and mAP of the given pre-trained models were considered, and the Faster R-CNN and the SSD with Inception V2 models will be implemented.

The pipeline configurations given in Fig. 3 and Fig. 4 only show the changes made from the pre-configured models.

Adjusting some of the parameters does not necessarily give similar results on other applications.

E. Training the Networks

In training the custom neural network models, it is expected to obtain a minimum TotalLoss value of 1.0 or less. The training job for both the Faster R-CNN and SSD with Inception V2 models can be monitored using the TensorBoard. Once the optimal range of TotalLoss is observed, the training job can be interrupted. Also, checkpoints that represent the training steps are being saved in the system unit as the training progresses. These checkpoints will be used in visualizing the training performance. The training for both networks took approximately 387 hours.

Fig. 5 depicts the TotalLoss graph obtained from training the Faster R-CNN with Inception V2 model while Table II shows the model's training metrics having TotalLoss between approximately 0.04 and 0.14.

Fig. 6 depicts the TotalLoss graph obtained from training the SSD with Inception V2 model while Table III shows the model's training metrics having TotalLoss between approximately 1.7 and 2.0.

Once the training jobs are complete, trained inference graphs will be generated to be integrated into the object detection program.

F. Estrus Detection Criteria

According to the research done by Tsai *et al.* an estrus event in images projects an object with a size of about 2-cows which will change into roughly 1.5-cows during the activity. Furthermore, based on the blob analysis and segmentation approach, if the distance between two centroids of the cows exhibiting "following" behavior is equal to or less than the distance threshold for more than 2 seconds or exactly equal to 4 seconds, the system will declare an estrus activity [7]. By adapting this research with the abovementioned study, the researchers were able to construct a similar detection rule for identifying the standing-heat activities of cows. The researchers initially hypothesized that in a panoramic top-viewed image depicting a standing-heat activity, the mounting (top) cattle's head and half body overlaps the other (bottom) cattle's half body. Consequently, having both objects stand very close to each other, an estrus activity can be declared.

In the numerical and photographic perspective, if the cattle's head and half of its body is treated as 0.5-cow while it mounts the other cattle's body (1.0-cow) on the prescribed time, the total length will eventually be equivalent to roughly 1.5-cows, giving the idea that the cow's features in pixels will be in the same range of value with the latter. Also, if the distance and the angles between their centroids meets a certain threshold, an estrus activity can be declared while taking all into account that the objects are highlighted by bounding boxes through the TensorFlow Object Detection API.

Faster R-CNN Configuration

```
Input: training and testing TFRecord files, and a label map file of the subjects
Output: configuration file for training
1: num_classes = {19};
2: feature_extractor:
3: type = {faster_rcnn_inception_v2};
4: second_stage_post_processing:
5: batch_non_max_suppression:
6: scales = {0.25, 0.5, 1.0, 2.0}
7: aspect_ratios = {0.5, 1.0, 2.0};
8: first_stage_nms_iou_threshold = {0.7};
9: second_stage_post_processing:
10: batch_non_max_suppression:
11: score_threshold = {0.0}
12: iou_threshold = {0.75}
13: max_detections_per_class = {1}
14: max_total_detections = {300};
15: train_config:
16: batch_size = {1}
17: learning_rate = {0.0002, 0.00002, 0.000002};
18: num_steps = {200,000};
19: data_augmentation_options = {autoaugment_image};
20: eval_config:
21: num_examples = {26,600};
```

Fig. 3. Pipeline Configuration for the Faster R-CNN Model.

SSD Configuration

```
Input: training and testing TFRecord files, and a label map file of the subjects
Output: configuration file for training
1: num_classes = {1};
2: feature_extractor:
3: type = {ssd_inception_v2};
4: anchor_generator:
5: ssd_anchor_generator:
6: aspect_ratios = {0.333, 0.5, 1.0, 2.0};
8: loss:
9: hard_example_miner:
10: iou_threshold = {0.99};
13: post_processing:
14: batch_non_max_suppression:
15: score_threshold = {0.0}
16: iou_threshold = {0.75}
17: max_detection_per_class = {19}
18: max_total_detections = {19};
19: train_config:
20: batch_size = {4};
21: learning_rate = {0.0002, 0.00002, 0.000002};
22: num_steps = {200,000};
23: data_augmentation_options = {random_rotate_90,
random_horizontal_flip,
random_vertical_flip, ssd_random_crop};
24: eval_config:
25: num_examples = {21,912};
```

Fig. 4. Pipeline Configuration for the SSD Model.

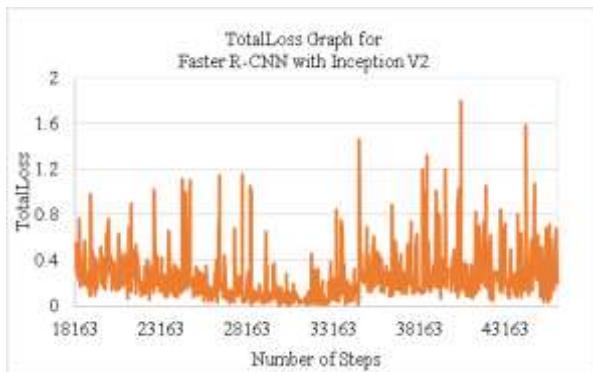


Fig. 5. The Line Graph Representation of the TotalLoss for the Faster R-CNN with Inception V2 Model.

TABLE II. TRAINING METRICS OF THE FASTER R-CNN MODEL

Steps	Value
45616	0.040272284
45672	0.431099415
45699	0.083715603
45727	0.708893716
45755	0.06934201
45783	0.584971786
45810	0.254837424
45866	0.302880734
45894	0.229811206
45977	0.143239096

The formula for the Euclidean distance, as in [25], (1) and the interior angles between centroid (2 and 3) are as follows:

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

$$\theta_y = \sin^{-1}[(y_2 - y_1)/D] \cdot (180^\circ/\pi) \quad (2)$$

$$\theta_x = \sin^{-1}[(x_2 - x_1)/D] \cdot (180^\circ/\pi) \quad (3)$$

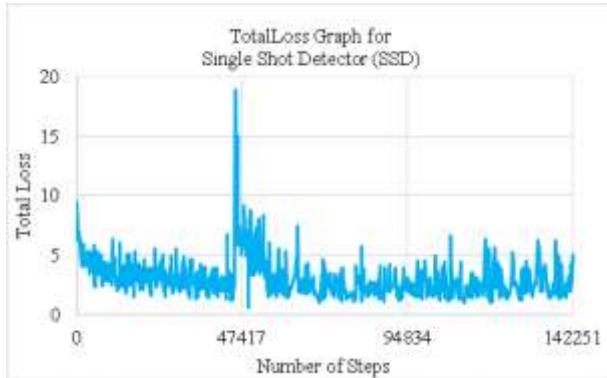


Fig. 6. The Line Graph Representation of the TotalLoss for the SSD Model.

TABLE III. TRAINING METRICS OF THE SSD MODEL

Steps	Value
140129	1.688523769
140171	1.409463882
140213	2.607795238
140467	1.906678438
140764	1.774537325
140978	3.171329737
141275	3.082624435
141445	1.949746

where D is the Euclidean distance between two centroids in pixels, x_1 is the centroid of the first object in x-axis, x_2 is the centroid of the second object in x-axis, y_1 is the centroid of the first object in y-axis, y_2 is the centroid of the Second object in y-axis, θ_y is the interior angle between centroids in y-axis, and θ_x is the interior angle between centroids in x-axis.

G. Overall Structure of the System

In the input section, the program will load the necessary packages, the label map, and the frozen inference graph that is generated and trained. Consequently, the camera will process the image frames through the VideoCapture objects of the program. In the image processing section, the SSD-based neural network will visualize "COW" predictions and identify object overlapping activities through bounding box corner analysis in real-time, as in [24], if the prediction score exceeds seventy percent. The program will also generate data frames [23] to contain information such as the Cow Name, ID, box coordinates and angles, and date and time of detection, considering there is only one class to be predicted in the image. If the data frames contain more than one detection, the program will filter out the prediction and will calculate the distances between two centroids of object instances and the interior angles between the two centroids and a point connecting it. After meeting the criteria, the program will iteratively count for the overlapping of object instances from 2 to 8 frames per second. If an overlapping of object instances occurred, as in [9] [24], then a copy of the frame will be directed to the Faster RCNN model, which will be initialized to perform image classification and object detection. The model will also be generating data frames to contain the Cow Names, IDs, box coordinates and angles, and date and time of detection of the nineteen classes predicted in the image. If the similar conditions are met in the Faster R-CNN model, an object overlapping or estrus activity will be declared, and the current image frame and record will be locally saved. Subsequently, the program will restart its counter and will continue to perform object detection. A flowchart represents the program flow of the automated estrus detection system using TensorFlow object detection API is illustrated in Fig. 7.

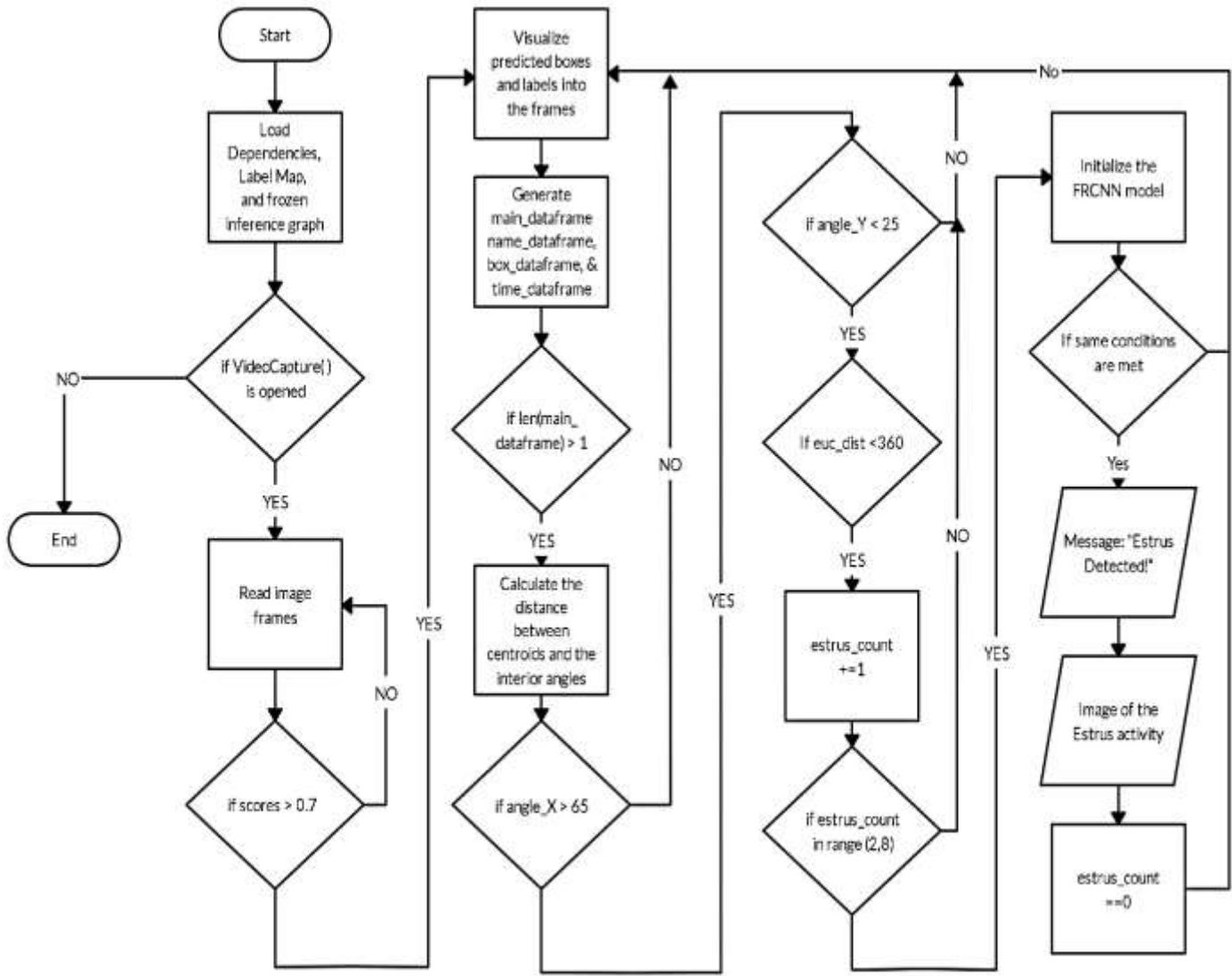


Fig. 7. The Line Graph Representation of the TotalLoss for the SSD Model.

IV. RESULTS AND DISCUSSION

A. Object Detection Results

The researchers deployed the system and operated locally in the barn for 4 months, with 10 hours of daylight and artificial light exposure in the barn. The system unit can execute the program at 30 fps and 1fps for image frame recognition with the SSD and the Faster R-CNN models, respectively.

Based on the results obtained, the system reported only two confirmed estrus events for 19 subjects in the trials, as shown in Fig. 8 and Fig. 9. Even after attaining acceptable and low TotalLoss values from the training of the Faster R-CNN model, the system still produced inaccurate cow predictions with 50% detection efficiency. According to the cow caretaker, the estrus activity depicted in Fig. 8 between “COW H” and “CARACOW” is validated. But in Fig. 9, the event is misidentified since it should be in-between the “BULL” and “COW Q”, but not in-between “COW P” and “COW N”, respectively.

Moreover, the confidence scores of the model for “COW N” and “COW P” are 71% and 75%, whereas, the confidence scores for “COW H” and “CARACOW” are 96% and 97%, respectively. Nevertheless, the SSD model effectively visualized “COW” objects with confidence scores of 94%, as shown in Fig. 10. These results suggest additional training time, dataset acquisition, and data cleaning to attain higher prediction scores for both models.

B. Database Results

Table III represents the validity of the results in monitoring the standing-heat of cattle. Based on the verification of the cow caretaker from the locally saved dataframes and images, the detected event in-between “CARACOW” and “COW H” is “TRUE” while the detected event in-between “COW N” and “COW P” is “FALSE due to the misidentification of the Faster R-CNN model which led to the 50% detection efficiency.

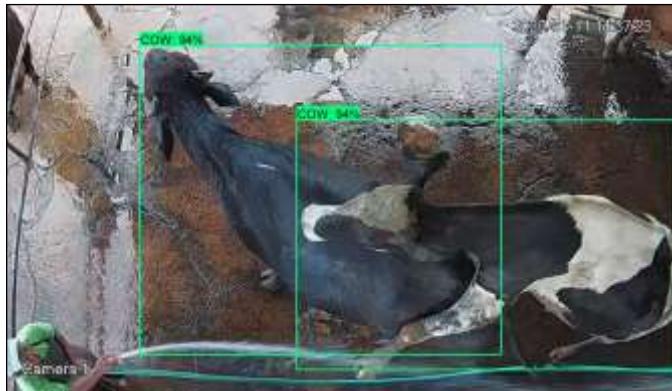


Fig. 8. An Image Frame of an Estrus Activity between the Cow (“COW H”) and the Water Buffalo (“CARACOW”).



Fig. 9. An Image Frame of an Estrus Activity between the Cow (“COW P”) and the Bull (“COW N”).



Fig. 10. An Image Frame of an Estrus activity between the Cow (“COW H”) and the Water Buffalo (“COW”).

As shown in Fig. 11, there are a total of 4 app-detections of standing-heat, 4 manually detected standing-heat signs, and 2 “True Positive” and “False Positive” detections from the program. As represented in Table IV, the end-user stated “FALSE” due to the incorrect detection of the system with “COW N” and “COW P” as in-heat cows, which instead should be the “BULL” and “COW Q”. Still, the system initially and correctly detected 4 standing-heat signs, but with 2 false predictions and identifications leading to 2 “True Positive” and 2 “False Positive” results, attaining a 50% detection efficiency.

C. Performance Assessment with other Related Works

Table V represents the summarized comparison framework between the proposed method and other relevant researchers in estrus detection. As abovementioned, this research deals with the detection of mounting behaviors of Holstein-Friesian and Sahiwal crosses, a bull, and a water buffalo. In contrast with the papers [7], [10]- [12], the proponents integrated a cattle identifier using customized neural network frameworks with a detection efficiency of approximately 90% and 50% for the Faster R-CNN and SSD models, respectively. Besides, most of the formulated methods do not include cattle identifiers since the researchers and the cow caretakers employ manual inspection of the cow tags after the process of standing-heat detection, by which, in this case, the system automatically identifies the cows and declares the estrus event at the same time.

The system also calculated a detection efficiency of 50% as a subsequent effect from the system's image classifier or cattle identifier. These results suggest the integration of other machine learning algorithms such as Foreground segmentation, background subtraction, support vector machine, and more within the deep learning framework, or the application of unsupervised learning in the detection system.

TABLE IV. TABULAR REPRESENTATION FOR THE VERIFICATION OF THE ACTUAL PROJECT TESTING

Cow ID	Date of Detection	Time of Detection	Estrus validity	Inseminated
1	January 11, 2020	2:37:22 PM	TRUE	NO
1229-1	January 11, 2020	2:37:22 PM	TRUE	YES
257-2b	April 2, 2020	06:24:13 AM	FALSE	NO
67	April 2, 2020	06:24:13 AM	FALSE	NO

SUMMARY OF STANDING-HEAT FOR THE YEAR 2020

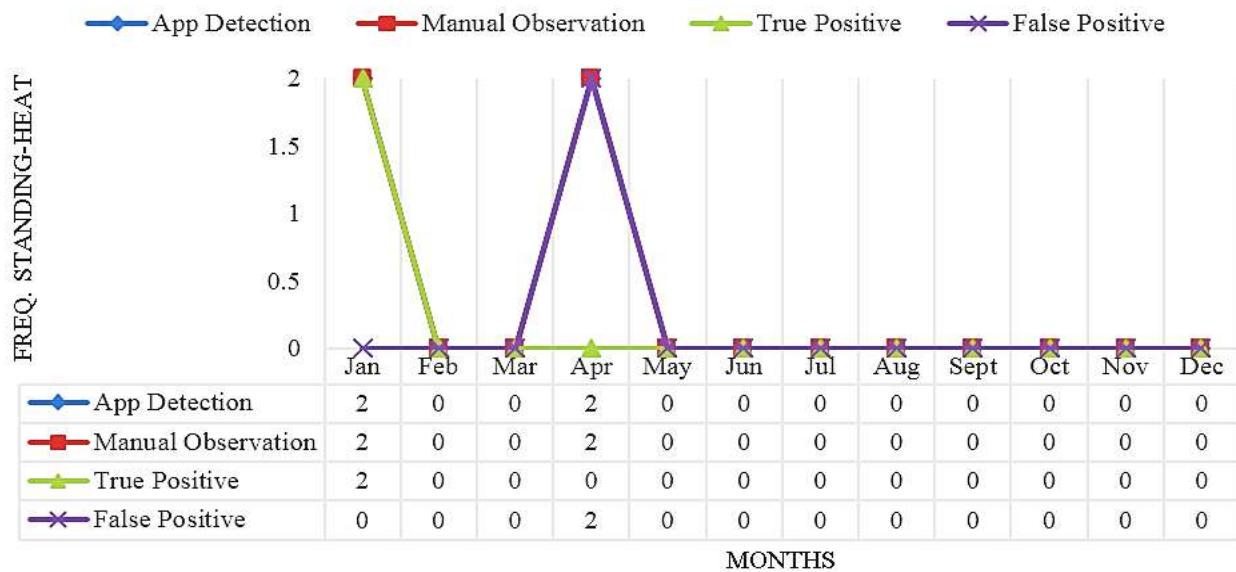


Fig. 11. Summarized Graphical Representation of Database Result for the Frequency of Standing-heat from January to December.

TABLE V. COMPARISON FRAMEWORK OF THE PROPOSED METHOD WITH OTHER RELEVANT WORKS IN TERMS OF PERFORMANCE AND ACCURACY

Author	Tsai and Huang [7]	Andrew, Greatwood, and Burghardt [9]	Yang, Lin, and Peng [10]	Xia <i>et al.</i> [11]	Guo, Zhang, He, Niu, and Tan [12]	Proposed work
Cattle Breed	Holstein	Holstein-Friesian	Holstein	Holstein	Holstein	Holstein-Friesian, Sahiwal
Sensors used	IP Dome Camera	UAV integrated camera	Infrared Camera	Pedometer and reader	Fixed IP Camera	PTZ Camera
Algorithm	Motion Detection, Region Segmentation, Foreground Segmentation, and Blob Analysis	R-CNN Localization, and Tracking	Motion Detection, Region Segmentation, Foreground Segmentation.	Motion Analysis	BSCTF, SVM, Geometric and Optical Flow Feature extraction	Faster R-CNN and SSD Localization and Tracking
Output	Image	Image	Image	Steps	Image	Image, Printed message
Accuracy in object (cattle) detection	-	98.13%	-	-	98.3%	94% (SSD), 50% (FRCNN)
Accuracy in estrus detection	100% (TP) 0.333% (FP)	-	-	91.86%	90.9% (TP) 4.2% (FP)	50% (TP) 50% (FP)
Limitations	Only at daytime, Indoor setup	Only at daytime, Not suitable for marker-less cattle	Shadow appearances	Indoor setup Lack of cattle identification	Indoor setup Lack of cattle identification	Indoor setup, Not suitable for marker-less cattle
Recommendations	Lameness detection	Larger herd identification	Changing type of camera, Cattle identification	None	None	Integrate other machine learning algorithms, implement unsupervised learning in the framework

V. CONCLUSION

In this study, the researchers presented a novel way of detecting estrus for dairy cattle, specifically the Holstein-Friesian and Sahiwal crosses, using the TensorFlow Object Detection API and its pre-trained models such as the Faster R-CNN and Single Shot Detector models with the Inception V2 as the feature extractor. Based from the obtained results, it can be concluded that (1) the Single Shot Detector (SSD) with Inception V2 proved to be effective in visualizing bounding boxes on the single class objects (e.g. "COW") with confidence scores of more than 90%, and (2) the Faster R-CNN with Inception V2 proved to be inaccurate in identifying objects with color indifferences between the subjects and the surface area of the barn obtaining a detection efficiency of 50%. Despite the inaccuracy, the proposed system can detect mounting behaviors of dairy cattle, given that the system will classify only one class (e.g. "COW) as shown in Fig. 8.

VI. FUTURE WORK

This research aims to report the preliminary attempt and provide learnings for other researchers to devise a system which classifies the dairy cattle subjects as well. The researchers also recommend to: (1) implement unsupervised learning techniques and machine learning algorithms within the deep learning framework that enhances the efficiency of cow classification and estrus detection without the aid of cowhide patterns, (2) develop a system that can monitor and detect mounting behaviors of cows on an outdoor setup, (3) define other suitable estrus detection criteria that maximize the camera's performance and line-of-sight, and (4) integrate a notification subsystem to immediately inform the end-users of the estrus events and initiate insemination on the cows.

ACKNOWLEDGMENT

The authors would like to express their deepest gratitude to Mr. Arcadio Francisco De Belen Jr., Farm Foreman, and the personnel of De Belen Dairy Farm for allowing the researchers to use the facilities, and to implement and deploy the project on the barn; and to the University Research and Development Services (URDS) of the Technological University of the Philippines, Ermita, Manila for the project funding.

REFERENCES

- [1] T. J. Parkinson, "Infertility in the Cow Due to Functional and Management Deficiencies," *Veterinary Reproduction and Obstetrics*, vol. 10, pp. 361-407, 2019.
- [2] M. Mária, P. Strapák, I. Szencziová, E. Strapáková and O. Hanušovský, "Several Methods of Estrus Detection in Cattle Dams: A Review," *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*, vol. 66, no. 2, p. 619 – 625, 2018.
- [3] J. B. Roelofs, C. Krijnen and E. v. E.-v. d. Kooij, "The effect of housing condition on the performance of two types of activity meters to detect estrus in dairy cows," *Theriogenology*, vol. 93, pp. 12-15, 2017.
- [4] R. W. Rorie, T. R. Bilby and T. D. Lester, "Application of electronic estrus detection technologies to reproductive management of cattle," *Theriogenology*, vol. 57, pp. 137-148, 2002.
- [5] L. G. S. Bersales, "Philippine Statistics Authority," January-December 2018. [Online]. Available: <https://psa.gov.ph/livestock-poultry-iprs/cattle/inventory>. [Accessed 2019].
- [6] S. Reith and S. Hoy, "Review: Behavioral signs of estrus and the potential of fully automated systems for detection of estrus in dairy cattle," *The Animal Consortium* 2017, vol. 12, no. 2, pp. 398-407, 2018.
- [7] D.-M. Tsai and C.-Y. Huang, "A Motion and Image Analysis Method for Automatic Detection of Estrus and Mating Behavior in Cattle," *Computers and Electronics in Agriculture*, vol. 104, pp. 25-31, 2014.
- [8] S. Talukder, K. L. Kerrisk, L. Ingenhoff, P. C. Thomson, S. C. Garcia and P. Celi, "Infrared technology for estrus detection and as a predictor of time of ovulation in dairy cows in a pasture-based system," *Theriogenology*, vol. 81, no. 7, pp. 925-935, 2014.
- [9] W. Andrew, C. Greatwood and T. Burghardt, "Visual Localisation and Individual Identification of Holstein Friesian Cattle via Deep Learning," in 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 2017.
- [10] C.-J. Yang, Y.-H. Lin and S.-Y. Peng, "Develop a Video Monitoring System for Dairy Estrus Detection at Night," in 2017 International Conference on Applied System Innovation (ICASI), Sapporo, Japan, 2017.
- [11] T. Xia, C. Song, J. Li, C. Li, G. Xu, F. Xu, J. Liu, G. M. P. O'Hare and Q. Zhou, "Research and Application of Cow Estrus Detection Based on the Internet of Things," in 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference, Guangzhou, China, 2017.
- [12] Y. Guo, Z. Zhang, D. He, J. Niu and Y. Tan, "Detection of cow mounting behavior using region geometry and optical flow characteristics," *Computers and Electronics in Agriculture*, vol. 163, 2019.
- [13] Porto, S. MC, C. Arcidiacono, U. Anguza and G. Cascone, "A computer vision-based system for the automatic detection of lying behaviour of dairy cows in free-stall barns," *Biosystems Engineering*, vol. 115, pp. 184-194, 2013.
- [14] J. O'Rourke and G. T. Toussaint, "Pattern Recognition," in *Handbook of Discrete and Computational Geometry*, Florida, CRC Press LLC, 2017, pp. 1421-1450.
- [15] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama and K. Murphy, "Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017.
- [16] J. S. Velasco, C. G. Pascion, J. W. Alberio, J. Apuang, J. S. Cruz, M. A. Gomez, B. J. Molina, L. Tuala, A. Thio-ac and R. L. J. Jr., "A Smartphone-Based Skin Disease Classification Using MobileNet CNN," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, pp. 2632-2637, 2019.
- [17] F. Al-Azzo, A. M. Taqia and M. Milanova, "Human Related-Health Actions Detection using Android Camera based on TensorFlow Object Detection API," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, pp. 9-23, 2018.
- [18] J. S. Velasco, N. M. Arago, R. M. Mamba, M. V. C. Padilla, J. P. M. Ramos and G. C. Virrey, "Cattle Sperm Classification Using Transfer Learning Models," *International Journal of Emerging Trends in Engineering Research*, vol. 8, pp. 4325-4331, 2020.
- [19] X. Ma, S. Pan, Y. Li, C. Feng and A. Wang, "Intelligent welding robot system based on deep learning," in 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019.
- [20] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2016.
- [21] D. A. D. E. Wei Liu, C. Szegedy, S. E. Reed, C.-Y. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," *European Conference on Computer Vision 2016*, vol. 9905, pp. 21-37, 2016.
- [22] V. N. and S. A., "Pre-processing," in *Studies in Systems, Decision and Control*, New York City, Springer, 2020, pp. 89-120.
- [23] Porcu, Python for Data Mining Quick Syntax Reference, New York City: Appress, 2018.
- [24] Y. He, C. Zhu, J. Wang, M. Savvides and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2883-2892, 2019.
- [25] O. B., Elementary Differential Geometry, Amsterdam, Netherlands: Elsevier Inc., 2006.