Computers, Materials & Continua
DOI: 10.32604/cmc.2023.029277
Article

Tech Science Press

Calf Posture Recognition Using Convolutional Neural Network

Tan Chen Tung[1], Uswah Khairuddin[1], Mohd Ibrahim Shapiai[1], Norhariani Md Nor[2,*], Mark Wen Han Hiew[2] and Nurul Aisyah Mohd Suhaimie[3]

[1]Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Kuala Lumpur, 54100, Malaysia
[2]Faculty of Veterinary Medicine, Universiti Putra Malaysia, Selangor, 43400, Malaysia
[3]Faculty of Bioresources and Food Industry, Besut, 22200, Malaysia
*Corresponding Author: Norhariani Md Nor. Email: norhariani@upm.edu.my
Received: 01 March 2022; Accepted: 20 May 2022

Abstract: Dairy farm management is crucial to maintain the longevity of the farm, and poor dairy youngstock or calf management could lead to gradually deteriorating calf health, which often causes premature death. This was found to be the most neglected part among the management workflows in Malaysia and has caused continuous loss over the recent years. Calf posture recognition is one of the effective methods to monitor calf behaviour and health state, which can be achieved by monitoring the calf behaviours of standing and lying where the former depicts active calf, and the latter, passive calf. Calf posture recognition module is an important component of some automated calf monitoring systems, as the system requires the calf to be in a standing posture before proceeding to the next stage of monitoring, or at the very least, to monitor the activeness of the calves. Calf posture such as standing or resting can easily be distinguished by human eye, however, to be recognized by a machine, it will require more complicated frameworks, particularly one that involves a deep learning neural networks model. Large number of high-quality images are required to train a deep learning model for such tasks. In this paper, multiple Convolutional Neural Network (CNN) architectures were compared, and the residual network (ResNet) model (specifically, ResNet-50) was ultimately chosen due to its simplicity, great performance, and decent inference time. Two ResNet-50 models having the exact same architecture and configuration have been trained on two different image datasets respectively sourced by separate cameras placed at different angle. There were two camera placements to use for comparison because camera placements can significantly impact the quality of the images, which is highly correlated to the deep learning model performance. After model training, the performance for both CNN models were 99.7% and 99.99% accuracies, respectively, and is adequate for a real-time calf monitoring system.

Keywords: Calf posture; machine vision; deep learning; transfer learning

1 Introduction

In Malaysia, the milk production rate has been declining in the recent years and it has been an ongoing issue that needs to be solved or at least mitigated. This is according to a survey by Department Of Veterinary Services Malaysia (DVS) [1], where the self-sufficiency level (SSL) of fresh and imported liquid milk has been declining from 71.55% in 2013 to 61.27% in 2018 (22.3% decrease), in other words, there could only be enough milk produced to feed 6 out of 10 people. On the contrary, the milk consumption in Malaysia has been increasing drastically ever since 2013 to 2018 from 37.6 million litres to 62.8 million litres, which is about 67.0% increase [1]. Government has been planning to increase the SSL level to 100% for fresh milk production, which is estimated to be about 65 million litres per year by 2024 as mentioned in the news by [2]. Indubitably, the number of cows, dairy youngstock or calves would also need to be higher to be able to cater to this huge amount of milk required.

Unfortunately, calves have been experiencing a high amount of mortality rate in the recent years in Malaysia. This is based on the survey done by us in 2018, in Sabah alone, 6 out of 24 dairy farms already shut down after suffering from continuous losses due to expensive farm expenditure, which includes land, dairy cows, and youngstock management. Out of the three factors, youngstock management was found to be the most neglected, as this was reflected by the high number of calf deaths. Based on the survey in Keningau, Sabah, there were up to 22% of the calves born on a dairy farm died. To resolve this problem, one of the most effective methods is to maintain a proper amount of daily feed intake for the calves to maintain their good health.

To estimate the proper feed intake amount, the farmers will need to know the calf weights in order to use a formula provided in a standard Microsoft Excel form known as Computing Complete Daily Feed Ration-CCDFR.6.0.4i, to calculate the appropriate amount. However, it is inconvenient and time-consuming to have to measure their weights daily, and furthermore, it requires the farmers to move the calves onto a weighing machine to weigh them. For the calves, this would induce unnecessary stress and likely to result in health issues due to their frailness as they are just newborn young calves. To overcome this problem, there have been plenty of research on using computer vision to estimate animal weights, not only for cow weight estimation [3–6], but also for pig weight estimation [7], chicken weight estimation [8], and so on.

For such a system to work in an automated way, a posture recognition module is sometimes required to ensure that the images are only captured when the animal is in a standing posture, especially when the size measurements of the animals are important features to be used for weight estimation, such as in the case of calves. Cow or calf weight are closely related to their body size and morphological traits. Therefore, in order for the features extracted from the images to be as consistent as possible, the images should only be captured during standing posture. This is because the distance between the camera and the calf would be different depending on the posture, such as between standing posture and resting posture. A posture recognition system is very simple to be incorporated into an automated weight estimation system because posture recognition can easily and effectively be done using computer vision with deep learning. The recognition result should also be very promising considering the rapid advancement of deep learning in the recent years.

Posture recognition is closely related to calf behaviours, there have been many studies that have worked on calf behaviour recognition for various behaviours such as standing, lying, walking, drinking, and ruminating [9–16]. Some of these studies, especially the newer ones, used machine-vision based deep learning algorithms for calf behaviour recognition, these studies are more relevant to the algorithms being applied in the present work. There was only one study that applied deep learning algorithms based on conventional Convolutional Neural Network (CNN) for image classification of the feeding and standing behaviour [15], while the others applied algorithms designed to accept video inputs rather than image inputs, which made the algorithms much more complicated than conventional CNN-based algorithms [10,13,14,16].

In the present work, as the dairy calves in Malaysia are mostly held in their own individual cells, they are not allowed to move around, thus behaviours that require temporal and motion-related features to be captured, such as walking, will not be included in our system. Our deep learning algorithm would only classify whether the calf is standing or lying (referred as "not standing" in this paper). Therefore, we proposed to use conventional CNN-based deep learning algorithm with transfer learning from a modern CNN architecture, ResNet-50 [17], which only accepts images rather than video sequences as inputs. The difference with the study from [15] is that they made their own relatively simple CNN architecture and trained from scratch, rather than using transfer learning from a pretrained model with one of the well-studied architectures, which would arguably perform much better in most cases. As shown in the present work, we achieved a binary classification accuracy close to 100%, with 99.7% and 99.99% accuracies for the first and second camera setups, respectively, compared to 92.61% obtained by the study from [15].

2 Methodology

2.1 Calf Image Database

The images of calves have been collected from a dairy farm in UPM's agriculture park. The specific breed of calves and cows used here is the Holstein-Friesian Cross breed. The images were being collected for 2 months continuously with a 5-s interval in between each frame from the camera feed, with some downtime due to electricity or network connection issues. However, not every image will be used in training the models as most of the images were quite similar, and the training time would also take a very long time. There were two different positions of camera being used to compare the results of using images captured at two different angles. Camera 1 was used for the first month, while camera 2 was used for the second month. The sample images from camera 1 and camera 2 can be referred in Figs. 1 and 2 below.

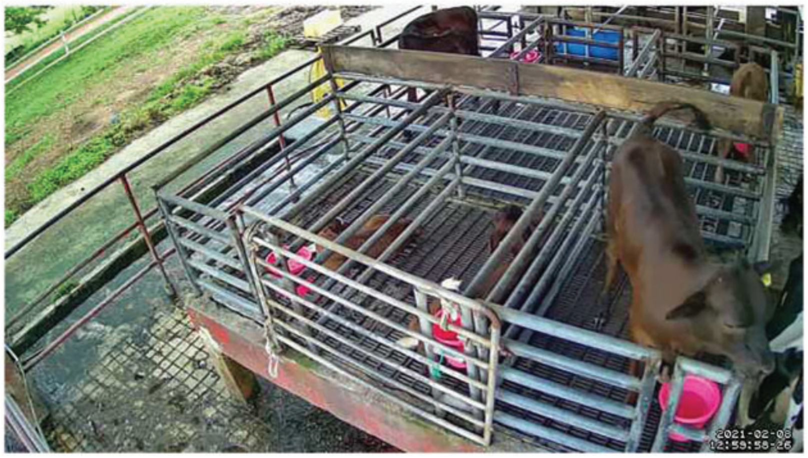

Figure 1: Sample image captured from the first camera position

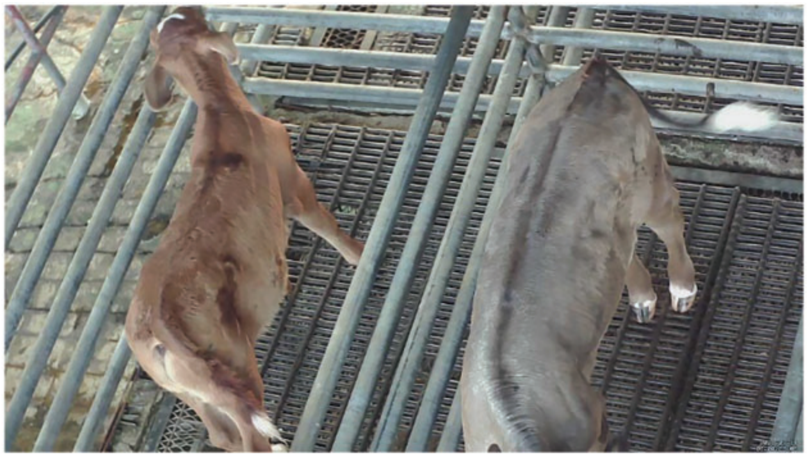

Figure 2: Sample image captured from the second camera position

2.2 Cropping of Individual Calves

As shown above, each image consists of at least two calves clearly visible and partially occluded by the metal bars from the cages. The second camera feed has a limitation of not being able to cover the entirety of the calves, but this angle will still be used as a comparison to determine how much this would affect the model performance. To allow the model to learn the posture of calves properly, two calves were cropped from each of the images: the middle calf and the right-side calf for the first camera; the left-side calf and the right-side calf for the second camera. These individual calf images are then resized to 224 × 224 (width and height) to be used for training.

2.3 Model Architecture for Feature Extraction

There are many common existing CNN-based architectures that have their algorithms pretrained on the ImageNet dataset [18], and most of the common ones were used in the present work for benchmarking purposes. The architectures are listed as follow:

1. DenseNet-201 [19]
2. EfficientNet-B0 [20]
3. EfficientNet-B7
4. Inception ResNet-V2 [21]
5. Inception-V3 [22]
6. MobileNet-V2 [23]
7. MobileNet-V3 Large [24]
8. NASNet Mobile [25]
9. ResNet-50 [17]
10. ResNet-152
11. VGG16 [26]
12. VGG19
13. Xception [27]
14. VGG16 [26]

A total of 13 deep learning models were trained on the images of the first camera setup. Out of all these architectures, only the ResNet-50 model architecture will be further elaborated for the present work, as this is the model that was ultimately selected for our calf posture recognition system due to its arguably simplest architecture compared to the others, while still being as performant as the best performing model, which would be further explained in Section 3.

2.4 Transfer Learning with ResNet-50 Model

For the posture recognition model (can also be considered as classification model in this case) proposed in the present work, an image classification model based on Residual Network (ResNet) [17] model has been used. Transfer learning was used by fine-tuning the pre-trained ResNet-50 model (consisting of 50 trainable neural network layers) on the calf posture recognition task. The ResNet-50 model has been pre-trained on the ImageNet [18] dataset. The ImageNet dataset consists of more than a million images of 1000 categories, including animals, hence the pre-trained ResNet-50 model will make the training converge much faster than purely training from scratch, while still achieving decent performance in the case of calf posture recognition task. The architecture of the model makes use of skip connections throughout its convolutional neural network layers, which is the essence of what makes the training of very deep neural networks feasible and more effective than usual [17]. Skip connections are essentially done by adding the input signal of a layer into the output of the layer, as a means to make the neural network to model the function of $f(x) = h(x) - x$ rather than just the usual $h(x)$ (see Fig. 3 below). This is also known as residual learning.

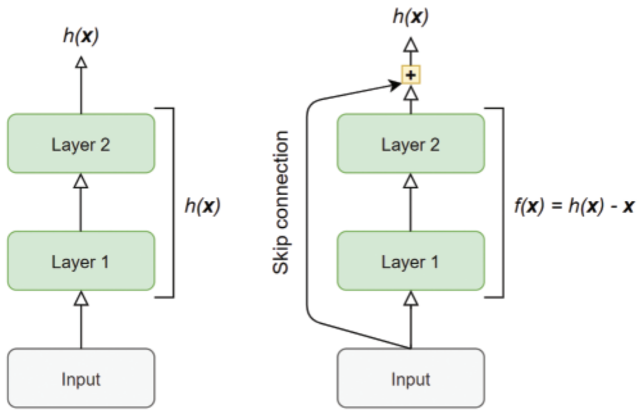

Figure 3: Normal layers (left). A residual block (right)

Our deep learning models were fine-tuned from pretrained models using the transfer learning method, which essentially cuts off the head or top part of the layers of the model and attaches a new set of layers to tailor to our specific posture classification task. The resulting model architecture modified from the original architectures can be seen in Fig. 4 below.

All of the models trained in the present work have the same head layers except for the new pooling layer, which could be average pooling layer (pool size 7 or 5) or global average pooling layer, depending on the number of input parameters passed into the pooling layer. If the number of parameters is small and cannot be fed into the average pooling layer of size 7, then our script would reduce the layer to 5, and if it is still unsuccessful, it would change to global average pooling layer. The effect of this different pooling layers would not make any significant difference as this pooling layer is only responsible for combining the outputs of CNN layers into the input format acceptable by the next fully connected

(FC) layer. For the final selected architecture of ResNet-50, it is using an average pooling layer of pool size 7.

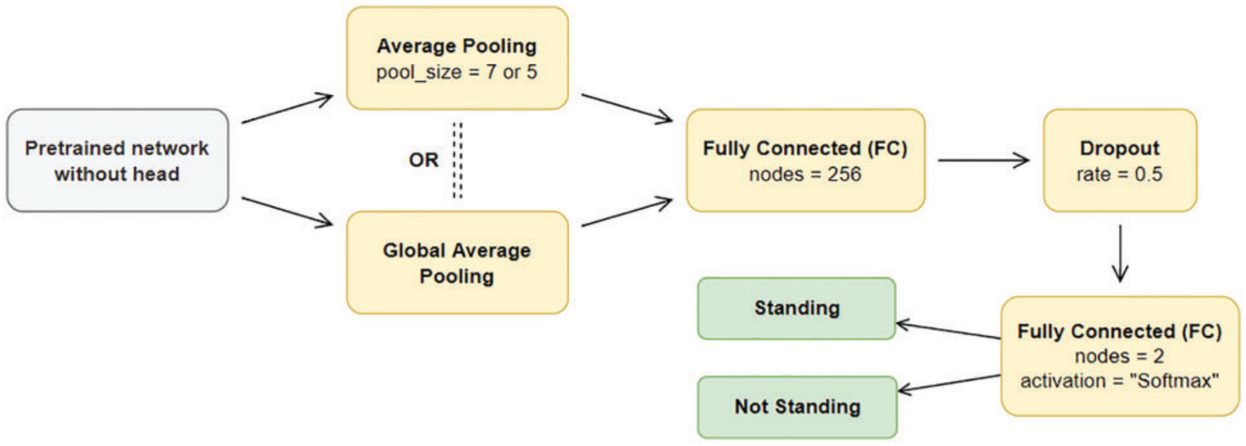

Figure 4: Our model architecture for posture recognition

The FC layer with 256 nodes was added to increase the number of new parameters for the model to learn, whereas the dropout layer with 0.5 dropout rate was added to avoid overfitting and allow better generalization, i.e., the model can still perform well in datasets outside of training set. The final FC layer has only two nodes for the two output classes—"standing" and "not standing". Softmax activation is then used to convert the model outputs into probabilities for each class, and the class with the higher probability will be selected as the final output class or behaviour. The ResNet-50 architecture is then used for the training the two different deep learning models for comparison of two different camera setups (one model for each setup).

2.5 Comparison of Two Models

The ResNet-50 model was fine-tuned specifically for binary classification of the standing posture of calves in this case. However, in the present work, there are two different models used as a comparison based on two different calf image datasets. Both of these datasets were obtained using the same 2D camera but at different angles of about 45 to 80 degrees from the calves, hence the resulting images are quite different, as shown above. Both the models were trained with the exact same hyperparameters and configuration (except for the different images) to make the comparison as fair as possible.

2.6 Machine Specification and Training Hyperparameters

The machine used to train all these models has a processor of Intel i7–8700 (3.20 GHz), with 16 GB RAM, 1 TB hard drive disk, and NVIDIA GTX 1060 (6 GB). All of the scripts and algorithms were developed in Python language. The training time for five epochs for each model was mostly around 20 min or less, with some larger models with many parameters (such as EfficientNetB7) taking around 30 min.

All the models used exactly the same hyperparameters for training, the values can be seen in Tab. 1. The number of training epochs is only 5 because the accuracies and losses seem to have already converged and stopped having any noticeable improvements (less than 0.1%).

Table 1: Hyperparameters used for all model trainings

Hyperparameter name	Value
Image size	224×224
Initial learning rate	1.0×10^{-4}
Batch size	64
Number of epochs	5
Number of iterations (or steps)	1960 (392 per epoch)
Gradient descent strategy	Adam
Loss function	Categorical cross-entropy

2.7 Dataset Splitting for Training and Testing

Before proceeding to training, the dataset was first split into training, validation, and testing set. The ratio of splitting is 67.5%:7.5%:25%. For practical and effective training of deep learning models, a validation set is required to provide an unbiased evaluation of the model's performance during the training process, and is the dataset used by us to strive to tune our model to improve the performance on. Ultimately, after the end of the training phase, then only the testing set is used to provide an unbiased evaluation of the model's performance on something the model has never seen before, and this testing set should not be touched at all during the training phase, this is to ensure the testing result is more representative of a real-world scenario.

3 Results and Discussion

Firstly, different CNN architectures are compared in terms of their performance and inference speed. After that, ResNet50 architecture was selected to proceed to use to compare its performance on two different camera setups. The first ResNet50 model was trained on the images collected from the first camera, while the second model was trained on the images collected using the second camera.

3.1 Model Architecture Performance Comparison

There is a total of 13 CNN architectures to be compared, as mentioned in Section 2.3. Their performance results can be seen in Tab. 2. The table is sorted by their accuracy in descending order. There is a total of 9,272 images and the results show that the average number of incorrect predictions is only 2.1, which makes all of their accuracies extremely close to 100%. This means most of these models are performing very well and is well-suited for real-world calf posture recognition tasks.

Table 2: Model performance results on test set images

Model	Correct predictions	Wrong predictions	Accuracy (%)
EfficientNetB7	9272	0	100
ResNet152	9272	0	100
DenseNet201	9271	1	99.99
MobileNetV2	9271	1	99.99
MobileNetV3Large	9271	1	99.99

(Continued)

Table 2: Continued

Model	Correct predictions	Wrong predictions	Accuracy (%)
ResNet-50	9271	1	99.99
Xception	9271	1	99.99
NASNetMobile	9270	2	99.98
VGG16	9270	2	99.98
VGG19	9270	2	99.98
EfficientNetB0	9269	3	99.97
InceptionV3	9268	4	99.96
InceptionResNetV2	9263	9	99.90
Average	9269.9	2.1	99.9785

ResNet50 performed extremely well and was very close to the top-performing models, which are much larger models (having a lot more parameters in their networks) such as EfficientNetB7 and ResNet52. To further compare their performance, the inference times were also computed, and Tab. 3 shows the total inference time on the test set and the calculated FPS for each model. The FPS is calculated by dividing the total time by the total test set images, which is 9,272.

Table 3: Model inference time on test set, GTX 1060

Model	Total time (s)	Average time (s)	FPS
MobileNetV3Large	16.899	0.002	548.6715
InceptionV3	27.211	0.003	340.7446
EfficientNetB0	28.189	0.003	328.9226
NASNetMobile	30.953	0.003	299.5509
ResNet50	41.833	0.005	221.6432
MobileNetV2	44.676	0.005	207.5387
VGG19	47.139	0.005	196.6949
Xception	48.621	0.005	190.6995
VGG16	59.191	0.006	156.6454
DenseNet201	64.138	0.007	144.5633
InceptionResNetV2	66.005	0.007	140.4742
ResNet152	97.67	0.011	94.93191
EfficientNetB7	148.342	0.016	62.50421
Average	55.4513	0.0060	167.2098

The total elapsed time is the total time required for the model to finish running inference on all test set images, beginning from the first image. The results shown in the Tab. 3 is based on a machine with a processor of i7–8700 and a graphics card of GTX 1060. Note that this inference time is based on a batch size of 64 with an optimized evaluation pipeline, therefore, this does not represent the real-world inference time where each video frame is streamed one-by-one to the algorithm for inference. All of these models were able to achieve a very high frame per second (FPS) during evaluation on test set, with an average of 167 FPS (computed by dividing 1 by 0.006, the average single-frame inference time).

Only the largest models—EfficientNet-B7 and ResNet-152 have noticeably longer inference time, specifically 267% and 176% slower than average, respectively. Nevertheless, most of these inference times are still very acceptable to be deployed in real-world scenario, as their FPS would still at least be significantly higher than 1 FPS. Ultimately, ResNet50 was chosen over MobileNetV3Large which had similar performance, due to the simplicity and the prevalence of ResNet50.

3.2 Analysis of Results on Images from First Camera

Fig. 5 below shows an output image from the first model from camera 1, consisting of boxes for each of the two calves used in the training process, with the labels their postures displayed on top of them. The boxes were also the exact positions where the calves were cropped from the full image to use to train the model.

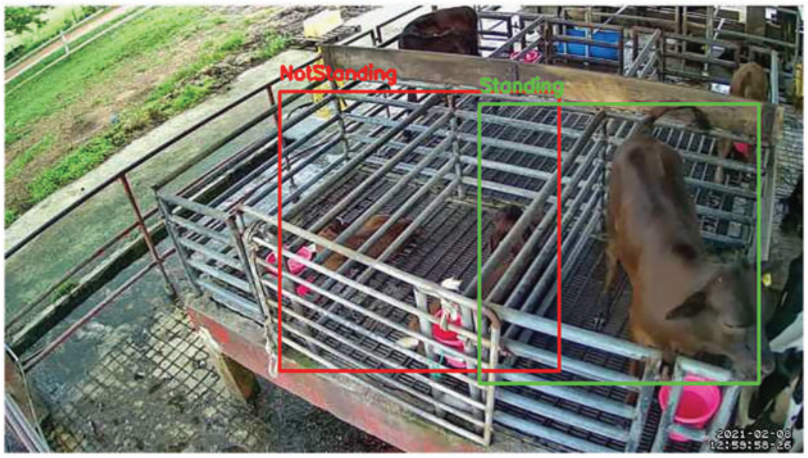

Figure 5: Sample classification output on a full image from the first camera

For the training process, as explained previously, the images are cropped into the size of 224 × 224 for each calf, then they are fed to the Convolutional Neural Network (CNN) layers of the model to learn to classify whether the calves are standing or not standing. After training on the training set, the training results show classification accuracies of close to 100% for both validation set and testing set. The distribution of the number of images in each category can be seen in the Tab. 4 below. It is evident that the number of images for "Not standing" is almost double of the number of images for the "Standing" posture, hence it is not a balanced dataset.

Table 4: Number of images for each split and posture

	Standing	Not standing
Training set	8,542	16,490
Validation set	944	1,837
Testing set	3,116	6,156

The confusion matrix in Fig. 6 below shows the classification results on the test set, with only one false positive, i.e., actual label is "Not standing" but classified as "Standing". Fig. 7 shows the only incorrect prediction made by the first model from camera 1, this can be justified by the reason that it is more challenging than most of the other images to classify, because this image was captured during the moment the calf was in the motion of standing up, as shown by the calf's right leg which was halfway through reaching a complete standing posture. This is hard to determine even from the perspective of human eyes.

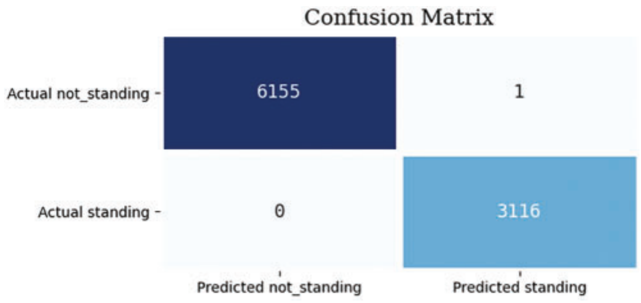

Figure 6: Confusion matrix for results on test set images by the first model from camera 1

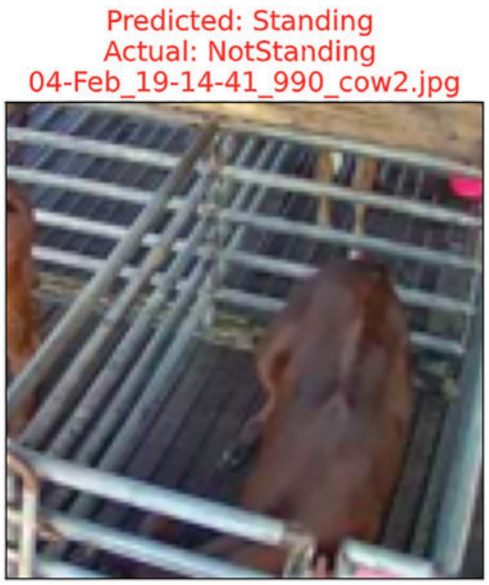

Figure 7: The incorrect prediction by the first model from camera 1

The results show that the first model from camera 1 achieved nearly 100% accuracy on the test set with only one incorrect prediction, even though the illumination conditions in the images were not consistent, as there were also dark images captured during nighttime being included in the dataset. This proved that the images obtained from camera 1 were able to demonstrate distinguishing features for the model to learn the general representations of the calf's postures.

The composite image in Fig. 8 below shows some of the sample prediction results from the trained the first model from camera 1. The labels above each of the image below show the predicted labels VS the actual labels, as well as the respective image filenames.

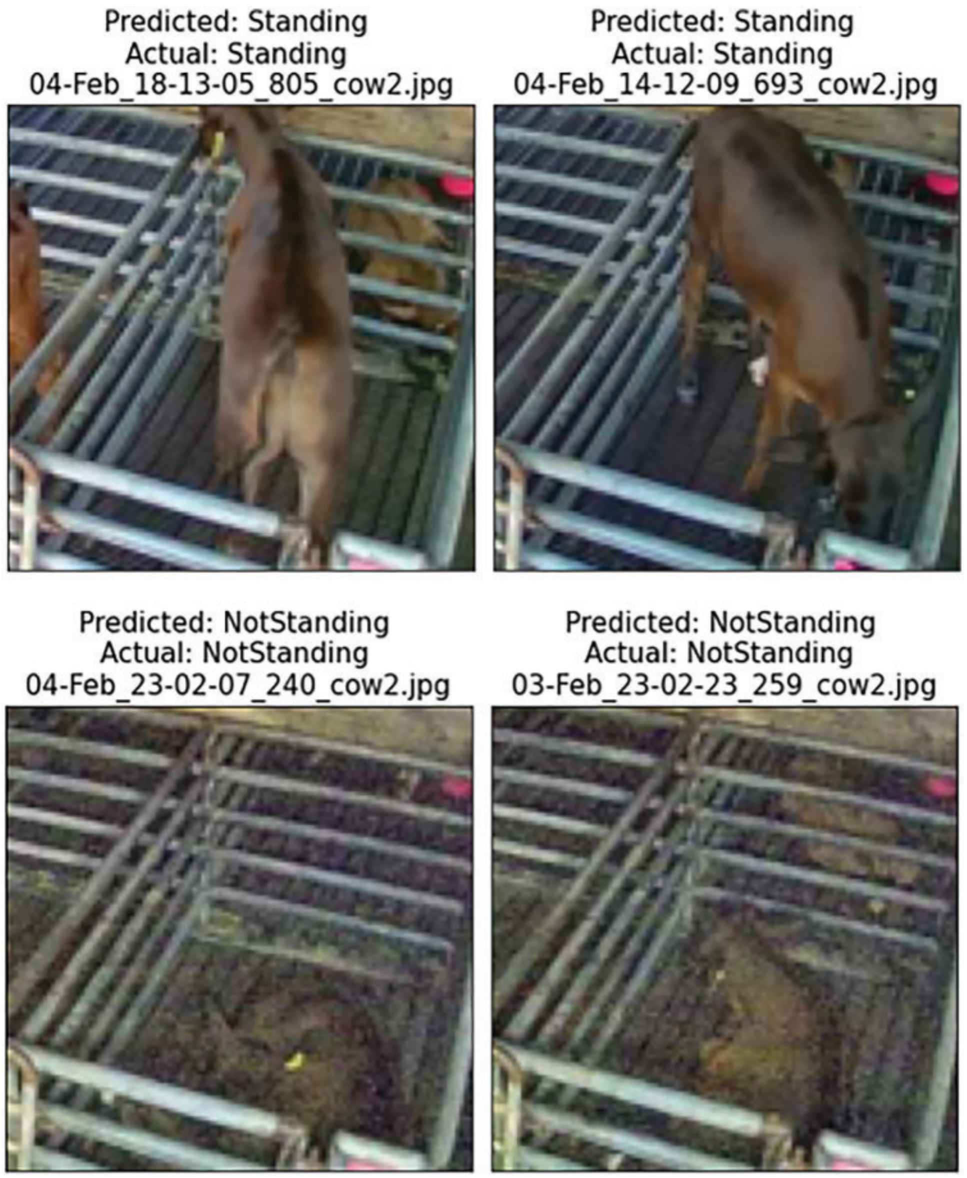

Figure 8: Sample prediction results by the first model from camera 1 on the test set images

3.3 Analysis on Results on Images from Second Camera

The image in Fig. 9 below shows a sample output by the second model from camera 2 on a full image from camera 2. Similar to the previous model, the individual calves were cropped at each position from the image to feed to the model for inference.

After training on the training set, the training results also show classification accuracies of close to 100% for both validation set and testing set, similar to the first model from camera 1. The distribution of the number of images in each category, however, is much more balanced this time (refer Tab. 5 below), which should result in better performance in most cases, but not for the dataset this time, and this will be shown later.

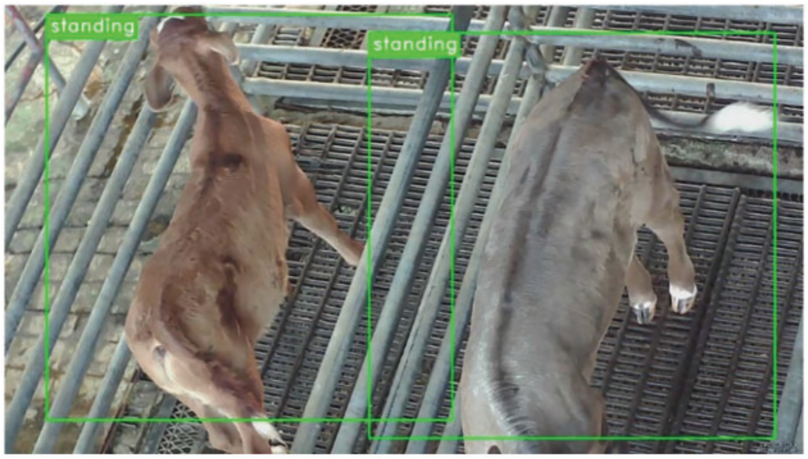

Figure 9: Sample prediction on a full image for camera 2

Table 5: Number of images for each split and posture

	Standing	Not standing
Training set	12,377	12,474
Validation set	1,357	1,404
Testing set	4,596	4,608

Fig. 10 below shows the confusion matrix of the results on the test set images. The result clearly shows that the second model from camera 2 has noticeably worse performance than the first model from camera 1, with accuracy of 99.7%, precision score of 99.7%, and recall score of 99.6%. A lower precision score implies that the model generates more false positives ("Actual not standing" but predicted as "standing"), while a lower recall score implies that the model generates more false negatives ("Actual standing" but predicted as "not standing"). There is a total of 31 incorrect predictions, which is significantly more than the only one incorrect prediction of the first model from camera 1 on test set.

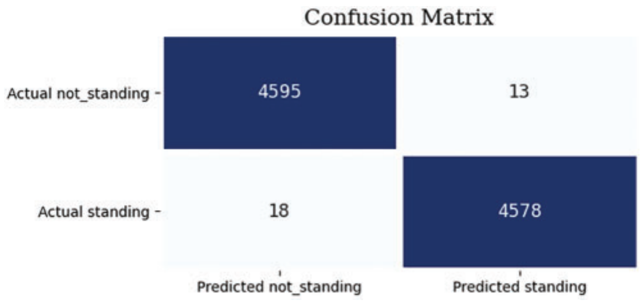

Figure 10: Confusion matrix for results on test set images by the second model from camera 2

This can be further verified by the top losses or top incorrect predictions from the the second model from camera 2 shown in Fig. 11. Almost every image from the top losses comes from the second calf (right-side calf in the full image), where the head or torso of the calf was not captured clearly in the image. Meanwhile, the camera was able to capture the entire body of the first calf most of the time, resulting in less incorrect predictions than for the second calf. These images would also pose difficulties even for humans to correctly judge the posture of the calves as standing or not standing. Therefore, it is imperative to ensure that the camera feed could capture the entire body of the calves, covering from their head to torso in order to allow the deep learning models to learn to make correct predictions. In other words, the calf postures from the images should at least be discernible for human eyes before deciding to use the specific camera position for the machine vision system.

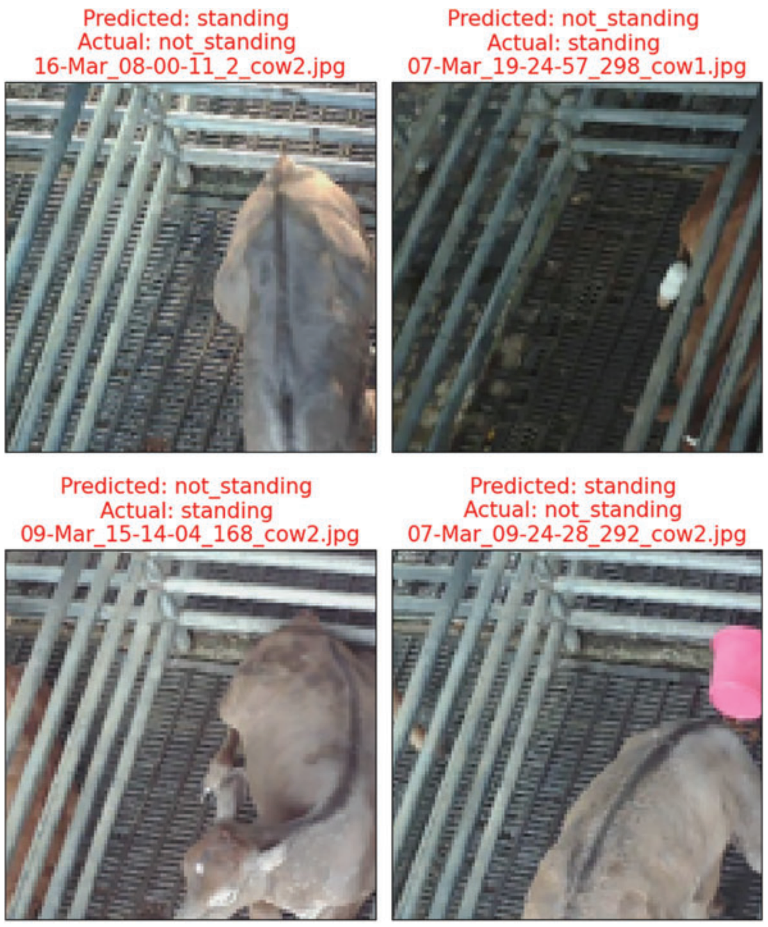

Figure 11: (Continued)

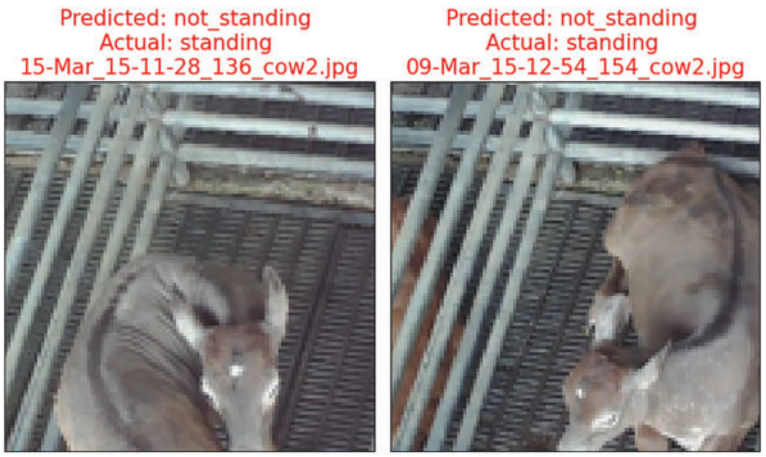

Figure 11: Top 6 incorrect predictions by the second model from camera 2 on test set images

4 Conclusion

From the analysis of the models trained on the images from the first and second camera, it is evident that the quality of the images is very crucial to enable the deep learning models to learn the accurate features of the calf postures to be able to make correct predictions.

The ResNet50 model used in the present work was also proved to outperform existing CNN-based models that accept image inputs, with 99.7% and 99.99% accuracies for the first and second camera setups, respectively, especially compared to 92.61% accuracy obtained by the study from [15]. The study used a much simpler custom-made CNN-based model architecture, rather than the ResNet50 architecture used in the present work, which showed superior performance.

The calf posture recognition module plays an important role in automated machine vision systems as it serves as the "switch" to control whether to proceed to the next phase of feature extraction for any relevant tasks such as weight estimation. The proposed deep learning model could solve this problem without much overhead as the ResNet-50 model is relatively lightweight but still performs exceptionally well on calf posture recognition. This module is also easy to be integrated into any machine vision system as it only requires the animal images to be able to work, ultimately improving the overall effectiveness of dairy farm management.

In the future, more functionalities could also be incorporated into such machine-vision based systems to further improve or expand them, such as adding a cow identification module using a similar technology to that of a vehicle re-identification system [28] to help detect the presence and the identity of the calves. Object tracking [29] can also be added to track the location of the calves instead of cropping the calves at a predetermined location, this could improve the quality of the cropped calf images to be fed to the deep learning model. Feed intake amount estimation and weight estimation modules are also useful to be included.

Acknowledgement: The authors would like to acknowledge Putra Agriculture Park (Previously known as UPM Agriculture Park) for allowing us to work at the calf rearing area and provide assistance as well as advice on calf rearing practices in the park.

Funding Statement: This project is funded under the Malaysian Young Researchers grant scheme (MRUN-MYRGS) Vote number: 5539500 (Universiti Putra Malaysia) Title: Precision surveillance system to support dairy young stock rearing decisions (NMN).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] D. of V. S. M. DVS, "Malaysia: Perangkaan Ternakan 2018/2019," 2019. [Online]. Available: http://www.dvs.gov.my/dvs/resources/user_1/2019/BP/PerangkaanTernakan/1_Malaysia_Perangkaan_Ternakan_.pdf.

[2] E. Jessinta, "Going Dutch to boost fresh milk production," 2019. https://www.nst.com.my/news/nation/2019/06/495493/going-dutch-boost-fresh-milk-production (accessed Jun. 04, 2020).

[3] Z. H. Pradana, B. Hidayat and S. Darana, "Beef cattle weight determine by using digital image processing," in *ICCEREC 2016 - Int. Conf. on Control, Electronics, Renewable Energy, and Communications 2016, Conf. Proc.*, Bandung, Indonesia, pp. 179–184, 2017. https://doi.org/10.1109/ICCEREC.2016.7814955.

[4] S. Tasdemir, A. Urkmez and S. Inal, "Determination of body measurements on the Holstein cows using digital image analysis and estimation of live weight with regression analysis," *Computers and Electronics in Agriculture*, vol. 76, no. 2, pp. 189–197, 2011.

[5] M. Haile-Mariam, O. Gonzalez-Recio and J. E. Pryce, "Prediction of liveweight of cows from type traits and its relationship with production and fitness traits," *Journal of Dairy Science*, vol. 97, no. 5, pp. 3173–3189, 2014.

[6] Y. Kuzuhara, K. Kawamura, R. Yoshitoshi, T. Tamaki, S. Sugai *et al.,* "A preliminarily study for predicting body weight and milk properties in lactating Holstein cows using a three-dimensional camera system," *Computers and Electronics in Agriculture*, vol. 111, pp. 186–193, 2015. https://doi.org/10.1016/j.compag.2014.12.020.

[7] Z. Li, C. Luo, G. Teng, T. Liu, Z. Li *et al.,* "Estimation of Pig weight by machine vision: A review," in *7th Int. Conf. on Computer and Computing Technologies in Agriculture (CCTA)*, Beijing, China, pp. 42–49, 2013.

[8] A. K. Mortensen, P. Lisouski and P. Ahrendt, "Weight prediction of broiler chickens using 3D computer vision," *Computers and Electronics in Agriculture*, vol. 123, pp. 319–326, 2016. https://doi.org/10.1016/j.compag.2016.03.011.

[9] J. Q. Gu, Z. H. Wang, R. H. Gao and H. R. Wu, "Cow behavior recognition based on image analysis and activities," *International Journal of Agricultural and Biological Engineering*, vol. 10, no. 3, pp. 165–174, 2017.

[10] J. McDonagh, G. Tzimiropoulos, K. R. Slinger, Z. J. Huggett, M. J. Bell *et al.,* "Detecting dairy cow behavior using vision technology," *Agriculture (Switzerland)*, vol. 11, no. 7, pp. 1–8, 2021.

[11] Y. Guo, D. He and L. Chai, "A machine vision-based method for monitoring scene-interactive behaviors of dairy calf," *Animals*, vol. 10, no. 2, pp. 1–14, 2020.

[12] Ö. Cangar, T. Leroy, M. Guarino, E. Vranken, R. Fallon *et al.,* "Automatic real-time monitoring of locomotion and posture behaviour of pregnant cows prior to calving using online image analysis," *Computers and Electronics in Agriculture*, vol. 64, no. 1, pp. 53–60, 2008.

[13] B. Jiang, X. Yin and H. Song, "Single-stream long-term optical flow convolution network for action recognition of lameness dairy cow," *Computers and Electronics in Agriculture*, vol. 175, article 105536, 2020. https://doi.org/10.1016/j.compag.2020.105536.

[14] D. Wu, Y. Wang, M. Han, L. Song, Y. Shang *et al.,* "Using a CNN-LSTM for basic behaviors detection of a single dairy cow in a complex environment," *Computers and Electronics in Agriculture*, vol. 182, article 106016, 2021. https://doi.org/10.1016/j.compag.2021.106016.

[15] B. Achour, M. Belkadi, I. Filali, M. Laghrouche and M. Lahdir, “Image analysis for individual identification and feeding behaviour monitoring of dairy cows based on convolutional neural networks (CNN),” *Biosystems Engineering*, vol. 198, pp. 31–49, 2020. https://doi.org/10.1016/j.biosystemseng.2020.07.019.

[16] X. Yin, D. Wu, Y. Shang, B. Jiang and H. Song, “Using an EfficientNet-LSTM for the recognition of single cow’s motion behaviours in a complicated environment,” *Computers and Electronics in Agriculture*, vol. 177, article 105707, 2020. https://doi.org/10.1016/j.compag.2020.105707.

[17] K. He, X. Zhang, S. Ren and J. Sun, “Deep residual learning for image recognition,” in *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016. https://doi.org/10.1109/CVPR.2016.90.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh *et al.,* “ImageNet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[19] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. - 30th IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2017*, Honolulu, HI, USA, 2017. https://doi.org/10.1109/CVPR.2017.243.

[20] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *36th Int. Conf. on Machine Learning, ICML 2019*, Long Beach, CA, USA, 2019. https://doi.org/10.48550/arXiv.1905.11946.

[21] C. Szegedy, S. Ioffe, V. Vanhoucke and A. A. Alemi, “Inception-v4, inception-ResNet and the impact of residual connections on learning,” in *31st AAAI Conf. on Artificial Intelligence, AAAI 2017*, San Francisco, CA, USA, 2017. https://doi.org/10.48550/arXiv.1602.07261.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 2818–2826, 2016. https://doi.org/10.1109/CVPR.2016.308.

[23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018. https://doi.org/10.1109/CVPR.2018.00474.

[24] A. Howard, M. Sandler, B. Chen, W. Wang, L. C. Chen *et al.,* “Searching for mobilenetv3,” in *Proc. of the IEEE Int. Conf. on Computer Vision*, Seoul, South Korea, 2019. https://doi.org/10.1109/ICCV.2019.00140.

[25] B. Zoph, V. Vasudevan, J. Shlens and Q. V. Le, “Learning transferable architectures for scalable image recognition,” in *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018. https://doi.org/10.1109/CVPR.2018.00907.

[26] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd Int. Conf. on Learning Representations, ICLR 2015-Conf. Track Proc.*, San Diego, CA, USA, 2015. https://doi.org/10.48550/arXiv.1409.1556.

[27] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. - 30th IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2017*, Honolulu, HI, USA, 2017. https://doi.org/10.1109/CVPR.2017.195.

[28] W. Sun, X. Chen, X. Zhang, G. Dai, P. Chang *et al.,* “A Multi-feature learning model with enhanced local attention for vehicle Re-identification,” *Computers, Materials and Continua*, vol. 69, no. 3, pp. 3549–3561, 2021.

[29] F. Bi, X. Ma, W. Chen, W. Fang, H. Chen *et al.,* “Review on video object tracking based on deep learning,” *Journal of New Media*, vol. 1, no. 2, pp. 63–74, 2019.