International Journal of Technology 13(6) 1202-1212 (2022)
Received April 2022 / Revised August 2022 / Accepted September 2022

International Journal of Technology

http://ijtech.eng.ui.ac.id

A Cow Crossing Detection Alert System

Yuan Qin Ong[1], Tee Connie[1*], Michael Kah Ong Goh[1]

[1]*Faculty of Information Science & Technology, Multimedia University, 75450, Melaka, Malaysia*

Abstract. Artificial intelligence is rapidly growing in recent years and has derived several branches of studies such as object detection and sound recognition. Object detection is a computer vision technique that allows the identification and location of objects in an image or video. On the other hand, proper recognition is the ability of a machine or program to receive and interpret dictation or to understand and carry out direct commands. This paper presents a cow crossing detection alert system with object detection and sound recognition capabilities. The proposed system aims to protect the driver from animal-vehicle collision. A data-driven deep learning approach is used for cow detection. Consequently, the cow detection module is integrated with a Raspberry Pi device to perform real time monitoring. The proposed cow crossing detection alert system will send an alert message to relevant units like the road transport department to take further actions if a potential animal-vehicle collision is detected on the road. Experimental results show that the proposed cow detection approach yields a mean average precision 0.5 (mAP 0.5) of 99% in object detection and 100% accuracy in sound recognition, demonstrating the system's practical feasibility.

Keywords: Cow crossing detection; IoT; Object detection; Sound recognition; YOLO

1. Introduction

In Malaysia, collisions between cows and vehicles are significant causes of road accidents, especially in rural areas. Apart from that, the collisions usually take place on the paths of villages (kampong roads). Usually, these paths have very few or no streetlights installed, so the drivers do not have good vision for driving and do not have enough time to react when encountering cows on the roadways. To address this problem, we propose an active roadways alert system that allows relevant parties to take preventive actions when cows are present in the area.

The proposed alert system is equipped with real time object detection and sound recognition modules to monitor roadway conditions. The alert system requires less budget and less effort for maintenance. Furthermore, when the system's camera or microphone detects a cow on roadways, the system will send an alert or notification to the relevant units like the Road Transport Department Malaysia. The relevant units can have informed knowledge about the condition of the roadways and perform further actions such as driving out cows from the highways. To reduce cost, Internet of Things (IoT) technology is deployed to automate the monitoring process (Jonny et al., 2021; Zahari et al., 2021; Munaf et al., 2020). IoTs are, physicalweight physical devices embedded with sensors, microprocessors software, and other technologies connected to the Internet or other communications

[*]Corresponding author's email: tee.connie@mmu.edu.my, Tel.: +606-2523592, Fax.: +606-2318840, doi: 10.14716/ijtech.v13i6.5874

networks to exchange data with other devices and systems. Raspberry Pi is an affordable, small-sized computer that can host operating systems. Therefore, Raspberry Pi is chosen in this research as the IoT device to be installed on the roadway to collect environmental data for cow detection using a camera and mini microphone.

Object detection is an essential component of the proposed system. Recently, YOLO has emerged as a popular object detection algorithm. It relies on neural networks to provide real time object detection. The benefit of using YOLO is that it allows for high object detection speed and high accuracy. In addition, sound recognition can assist the limitation of object detection in dark conditions. Hence, the trained sound recognition model can alert the relevant units when it detects cows on roadways when the camera sensor is out of service, or the visual cue is affected due to bad weather conditions. In this research, YOLO with a custom and sound recognition model is trained and run in Raspberry Pi. The proposed YOLO with a custom model can achieve high accuracy of 99% of mAP 0.5 for cow detection and the sound recognition model can obtain 100% accuracy.

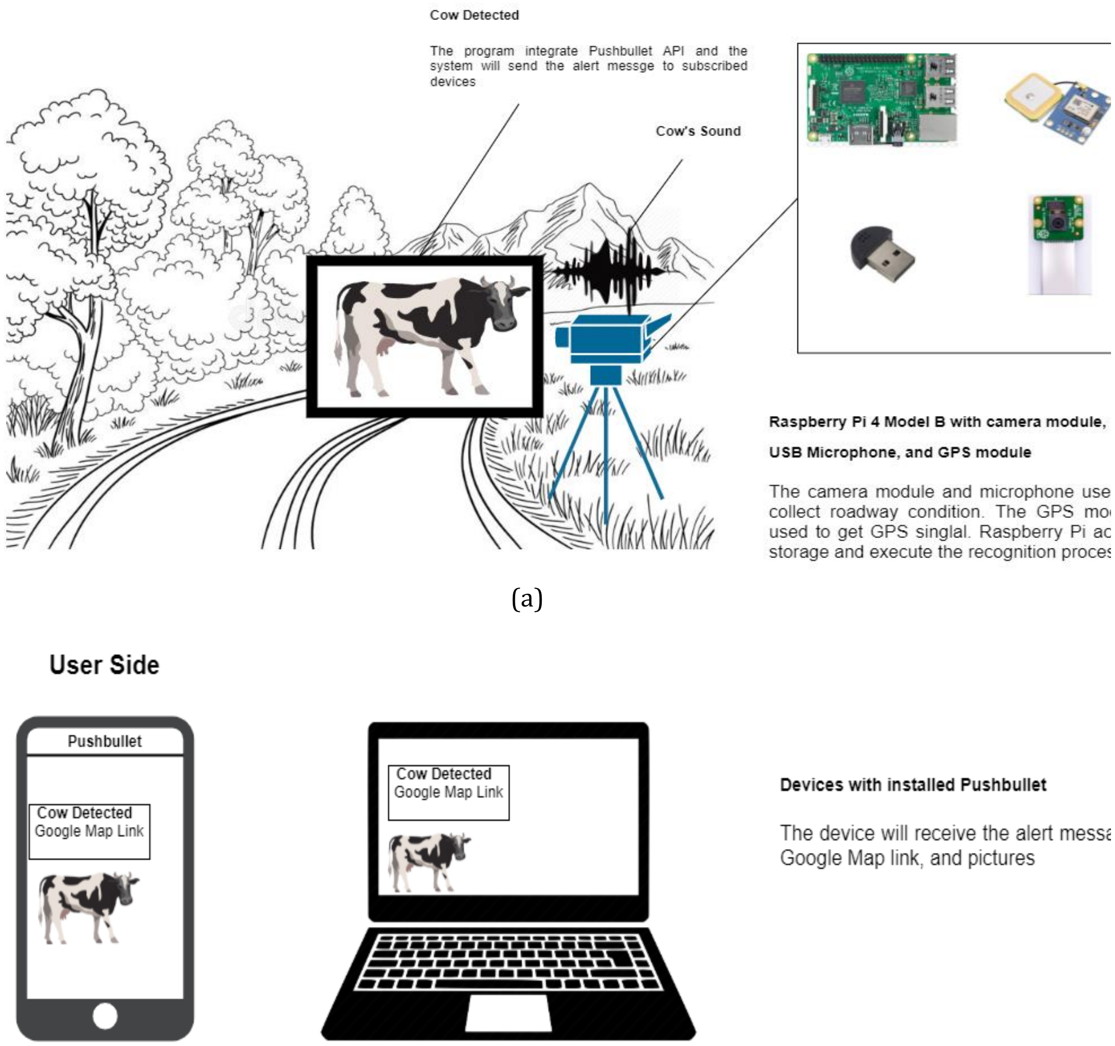
2. Methods

2.1. Cow Crossing Detection Alert System

In this paper, an object detection model is trained with a custom dataset so that it can recognize cows from different angles, such as the front, side, and back. A sound processing module is also developed to analyze the sound signals. The system will capture videos and sounds of the roadway at fixed durations. The cow crossing detection alert system is installed on roadsides to monitor road condition. The system will read the captured video's frame to execute the recognition process using the custom YOLO model. The system also performs sound recognition at the same time. The system will send alert messages to subscribed devices if the detection result indicates the occurrence of a cow in the scene under observation. If the system does not detect the existence of cows on the roadway, it will wait for the following input from the camera and microphone to execute the recognition process.

Some existing techniques or solutions use GPS trackers to track cows to ensure that they do not enter the roadway area. However, there are limitations to such practices. The GPS trackers are not able to track the cow's position correctly and accurately. Malaysia is a tropical country, and many roadways are covered by dense forests that cause the GPS trackers to lose signal to provide correct location data. The proposed system, on the contrary, can fully address this issue. The proposed approach does not rely on the availability of the GPS signal but is based on visual and audio cues instead to monitor the cows.

Five core characters are required in the proposed cow crossing detection alert system: Pushbullet, Raspberry Pi camera module, mini-USB microphone, GPS module, and Raspberry Pi. The Raspberry Pi camera module and mini-USB microphone are used to capture the roadway condition for recognition. Furthermore, Raspberry Pi is used to execute the cow recognition process, and it acts as a storage to save the input video, sound data recorded, and essential data. Besides, the latitude and longitude of the Raspberry Pi device can be used to generate a Google Map link so that relevant units can track the cow's location easily. Next, Pushbullet plays a major role on the user side because it sends the alert message with a map link to the relevant units' devices, such as a computer or smartphone, if the system detects a cow on the roadway. Figure 1 illustrates a scenario of the proposed cow crossing detection system.


Cow Detected
The program integrate Pushbullet API and the system will send the alert messge to subscribed devices
Cow's Sound
Raspberry Pi 4 Model B with camera module,
USB Microphone, and GPS module
The camera module and microphone use
collect roadway condition. The GPS mo
used to get GPS singlal. Raspberry Pi ac
storage and execute the recognition proce
User Side
Pushbullet
Cow Detected
Google Map Link
Cow Detected
Google Map Link
Devices with installed Pushbullet
The device will receive the alert messa
Google Map link, and pictures

(a)

(b)

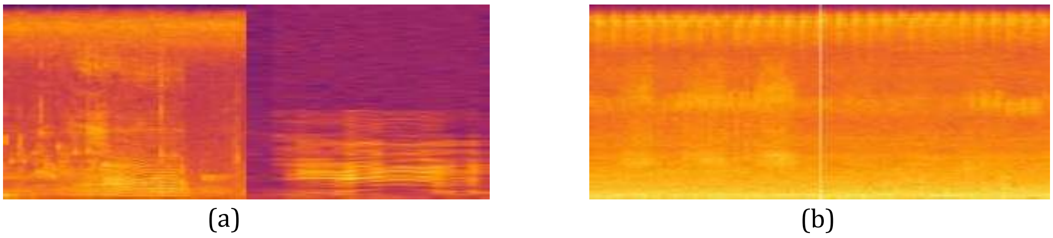
Figure 1 The proposed cow crossing detection alert system

2.2. YOLO: Real Time Object Detection

In this research, YOLOv3 is chosen and integrated with Raspberry Pi. YOLOv3 is an advanced real time object detection algorithm that is faster than other detectors such as Retinanet-50-500, SSD321, and so on (Redmon & Farhadi, 2018). YOLOv3 is based on the CNN architecture to detect objects real time YOLOv3 can interpret images as a structured array of data and find patterns between them using CNNs. The YOLOv3 algorithm performs prediction based on 1x1 convolutions of convolutional layer, and the input image only requires one forward propagation pass across the network to produce high accuracy prediction at a high speed. YOLOv3 employs a novel approach to perform prediction by using a single neural network to process the whole picture. Once the concept has been divided into areas, the network calculates the probabilities associated with each of those parts. The projected possibilities are used to weight the produced bounding boxes. One more important technique used in YOLOv3 is non-max suppression. This technique ensures that each item is detected once and discards any false detections before returning the identified objects and their bounding boxes.

2.3. Sound Recognition Model

Apart from object detection, our system also includes a sound recognition model. Audio data analysis is essential to processing and comprehending audio signals obtained from digital devices. Spectrogram plays a vital role in audio analysis because we can extract crucial audio signal features from the spectrogram to train a model. The spectrogram provides a valuable way to understand the audio signal and display the audio signal graphically. Moreover, every audio collected by Raspberry Pi consists of many important features that can be used to predict the sound movement. One of the valuable features that be extracted from an audio signal is spectral centroid. The spectral centroid allows us to quickly locate the centre of mass for a sound (Chauhan, 2020). Next, spectral rolloff is used to measure the shape of the signal. Besides, spectral bandwidth is another helpful feature. Spectral bandwidth is defined as the width of a light band at one-half of its most significant value. In this study, a zero-crossing rate is also used to determine the smoothness of a signal by calculating the number of zero-crossings in a segment of a movement. All the features are fed as input to a neural network to perform classification. Figure 2 illustrates some examples of the spectrogram of the audio signals. Figure 2a shows the cow audio signals, while Figure 2b resemble the audio signals without a cow.



(a) (b)

Figure 2 Spectrogram of audio signals: (a) cow, (b) without a cow

2.4. Fusion of Images and Sound Signals

This study uses score-level fusion to consolidate the audio and visual signals. Score-level fusion combines the match scores produced by different matchers to make a judgement regarding an individual's identification (Ross & Nandakumar, 2009). The cow crossing alert system uses the score-level fusion to determine the existence of cows from two modalities at different fusion levels: image and sound, before sending the alert messages. Figure 3 provides an illustration of how the score-level fusion is applied in the study. In the beginning, both the visual and audio be fed to the respective backbones/models to yield a matching score. The coordinating scores from the two models are joined with the exceptional coordinating score. The system used the SUM operation for score level combination. If the melded coordinating scores (MF) are bigger than the target (T) score, the system will send an alert message. For example, suppose the value for T is set to 1. The YOLO model returns a score of 1 as a cow is detected, but the sound processing module returns a 0 score. In this case, the total is 1 and is not greater than 1. Therefore, the system will not send an alert message because the condition is no met.

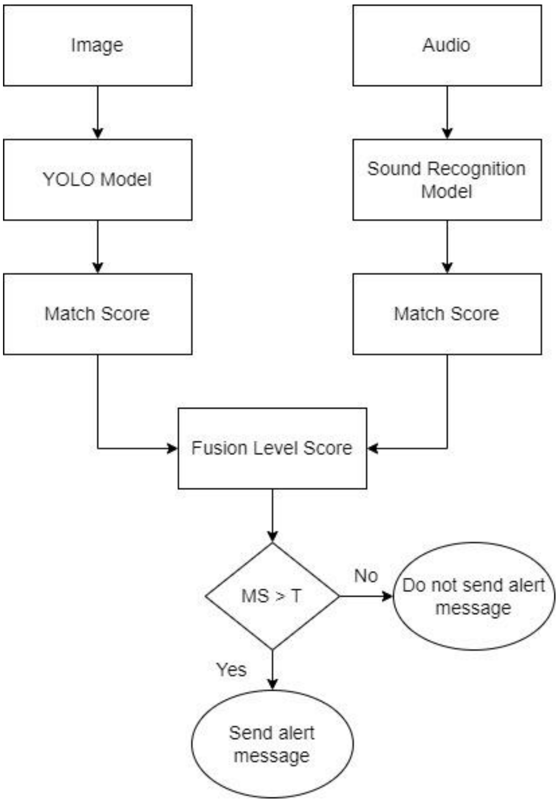


Figure 3 Score level fusion approach

2.5. Platform to Receive an Alert Message

The system will send alert messages to subscribed devices such as computers or smartphones with the Pushbullet application. The system will send an alert message to the devices when one of the conditions are met:

1. A cow is detected in the video, but no cow's sound is detected
2. A cow's sound is detected, but there is no cow seen in the video
3. Both of the video and sound witnessed the occurrence of a cow

The reason for setting the conditions is to allow more possibilities to detect the cows due to the dynamic environment in real-world. The visual input especially is vulnerable to change in illumination. The cow objects might not be visible at night or in bad weather conditions. Therefore, the audio signal can be a complementary information in the proposed cow detection system.

3. Results and Discussion

3.1. Dataset

3.1.1. Image Dataset

In this paper, about 400 cow and 100 vehicle images are obtained from Kaggle. For clarity, this dataset is denoted as Dataset 1. The collected samples include various backgrounds and different views of cows and cars. The collected data are pre-processed by auto-orientation to prevent feeding the model with wrong information. After that, the images are resized to 416 x 416 pixels. The samples are stored in Pascal VOC format for further use. After that, data augmentation is applied on the images to increase the dataset size. The operations performed include random gaussian blur of between 0 and 3 pixels, and salt and pepper noise that are applied to 5 percent of the pixels. This results in a total of 1300 images coined as Dataset 2.

(a) (b) (c) (d)

Figure 4 Different views of the cow and car

3.1.2. Sound Dataset

The sound datasets are collected from online public datasets such as UnrbanSound. This study collected 150 cow and 150 non-cow sound signals. The no cow sounds contain sounds from the surrounding that people always hear on the roadway, like the sound of air conditioner, car horn, and engine idling. Figure 5 shows some sample sound signals acquired in this study. Features such as Mel-frequency cepstral coefficients (MFCC), Spectral Centroid, Zero Crossing Rate, Chroma Frequencies, Spectral Roll-off are extracted.

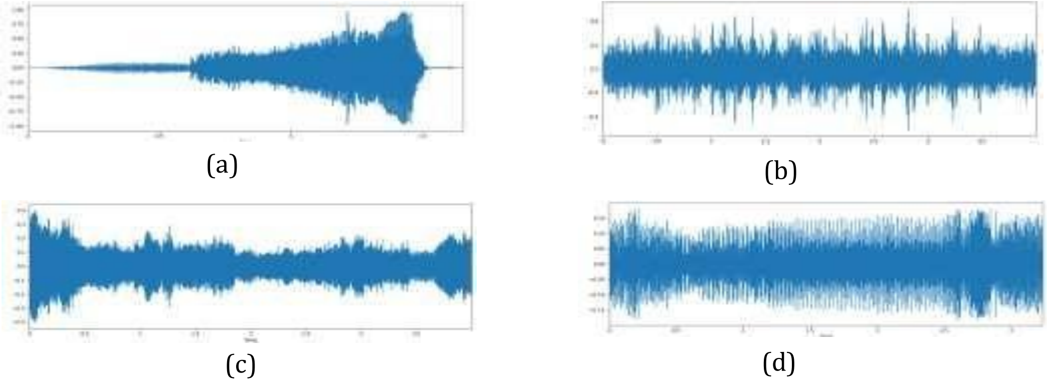

(a) (b) (c) (d)

Figure 5 Samples sound signals of: (a) cow, (b) air conditioner, (c) vehicle horn, (d) engine idling

3.2. Experimental Setting

All the experiments were conducted in Google Colaboratory with the following requirements: GPU: 1xTesla K80, compute 3.7, having 2496 CUDA cores, 12GB GDDR5 VRAM, CPU: 1xsingle core hyper threaded Xeon Processors @2.3Ghz i.e(1 core, 2 threads), RAM: 12.6 GB and 33 GB of disk space.

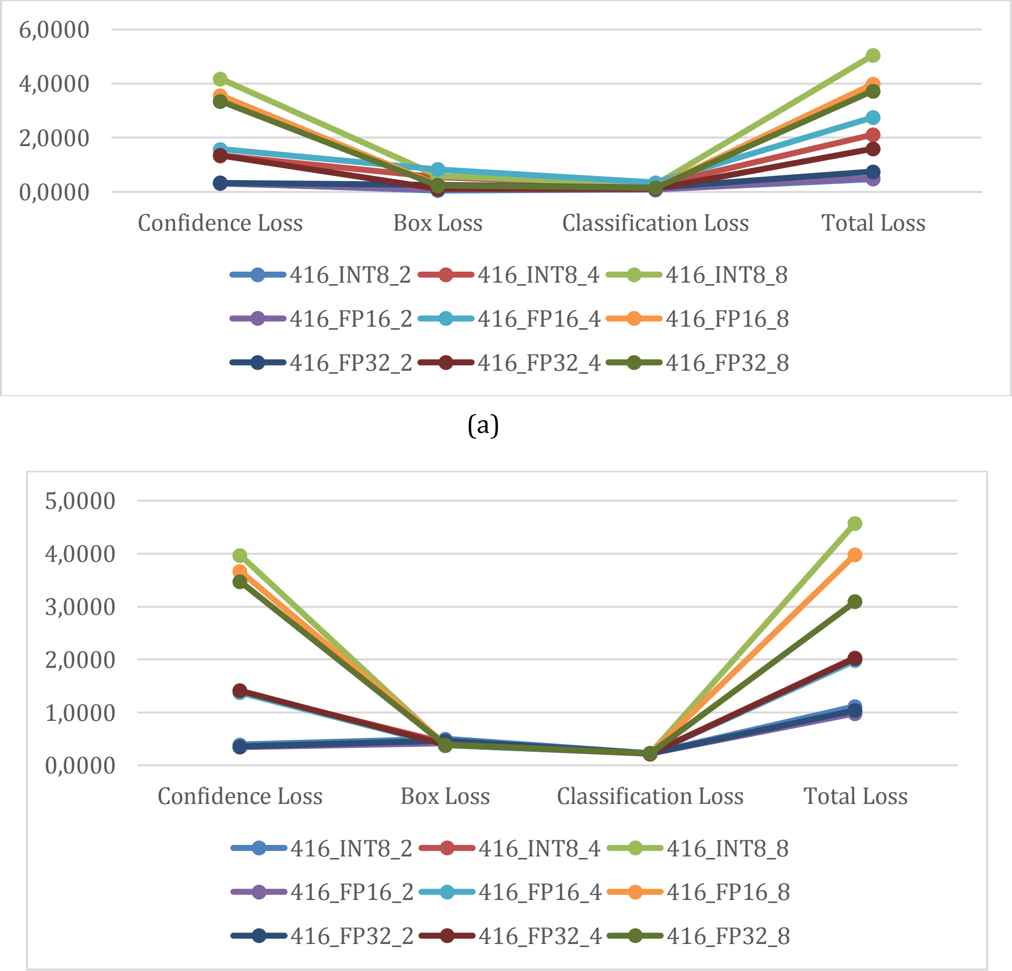
The dataset is split into 80% for training and 20% for testing. For clear indication, each YOLO model is named using the sequence of input size, YOLO quantize model, and batch size, for example 416_INT8_4 denotes a model having an input size of 416 x 416 pixels with an 8-bit quantized version of the model and having a batch size of 4. To further differentiate the expanded dataset (after augmentation) from the original dataset, we name the original dataset and developed dataset as Dataset 1 and Dataset 2, respectively. The hyperparameters used in the experiments are as follows: Epoch: 100, Framework: TensorFlow, IOU Loss Threshold: 0.5, Input Size: 416x416, Initialization Train Learning Rate: 0.0001, Ending Train Learning Rate: 0.00001, YOLO Quantize Mode: INT8, FP16, FP32, Batch Size: 2, 4, 8.

3.3. Experiment Analysis

3.3.1. Experiment Analysis For Object Detection

3.3.1.1. Performance of Custom YOLO Model

We first assess the performance of using a custom YOLO model. Dataset 2 is used for this purpose. The custom YOLO model is trained with different hyperparameters to obtain optimal performance. Figure 6a shows the loss values of using different YOLO models. We observe that 416_INT_2, 416_FP16_2, 416_FP32_2 can get low training loss as compared to other models. Batch size 2 appears to be the best batch size for training the YOLO model. Apart from that, Figure 6b depicts the validation loss values of different YOLO models. We can see that 416_INT_2, 416_FP16_2, 416_FP32_2 still yield the lowest validation loss among the other models.



(a)

(b)

Figure 6 (a) Training and (b) validation of different YOLO models

Table 1 shows the results from all the hyperparameters. We observe that the train total loss of 416_INT8_2 is much lower than the validation total loss, so the 416_INT8_2 is facing overfitting problem although its train total loss is lower than the other hyperparameters. Moreover, 416_FP16_2, 416_FP32_2 can obtain good results in both train total loss and validation total loss, and no overfitting problem.

Table 1 Loss of different hyperparameters

Model	Train confidence loss	Train box loss	Train classification loss	Train total loss	Val confidence loss	Val box loss	Val classification loss	Val total loss
416_INT8_2	0.3278	0.05208	0.095	0.4749	0.3877	0.5011	0.2269	1.115
416_INT8_4	1.341	0.5355	0.2412	2.117	1.380	0.4347	0.2232	2.038
416_INT8_8	4.184	0.5916	0.2721	5.048	3.972	0.3785	0.2246	4.576
416_FP16_2	0.3176	0.1079	0.0947	0.5202	0.3452	0.4132	0.2236	0.982
416_FP16_4	1.574	0.8303	0.3496	2.754	1.378	0.3801	0.2199	1.978
416_FP16_8	3.566	0.2581	0.1591	3.983	3.67	0.4076	0.2228	3.983
416_FP32_2	0.336	0.2614	0.1581	0.7555	0.3512	0.4635	0.2247	1.039
416_FP32_4	1.352	0.118	0.1226	1.597	1.414	0.3886	0.2214	2.024
416_FP32_8	3.356	0.2291	0.1512	3.736	3.476	0.3916	0.226	3.094

The results of using different YOLO models are summarized in Table 2. We observe that the 416_FP32_2 model achieves the best mAP0.5 and FPS among the nine models, which is 1 for mAP0.5 at 13.51 FPS. Although the 416_FP32_4 model can also achieve 13.09 FPS, its speed is still slight lower than 416_FP32_2. On the other hand, the performance of 416_FP16_8 is the worst because it only obtains 0.0076 for mAP0.5 and 5.7 for FPS.

Table 2 Performance of different YOLO models

Model	mAP0.5	FPS
416_INT8_2	1	6.05
416_INT8_4	0.99	7.32
416_INT8_8	1	6.92
416_FP16_2	1	7.17
416_FP16_4	1	6.79
416_FP16_8	0.0076	5.70
416_FP32_2	1	13.51
416_FP32_4	1	13.09
416_FP32_8	1	6.97
416_INT8_2	1	6.05

3.3.1.2. Performance Comparisons Between Models Trained by Dataset 1 and Dataset 2

This experiment aims to evaluate the performance of the proposed methods using Dataset 1 and Dataset 2. The 416_INT8_4 model is applied in the investigation. The total training loss and validate total loss are used to evaluate the model. Figure 7a show the training confidence loss, training box loss, training classification loss, and total loss for Dataset 1. On the other hand, the validation confidence loss, validation box loss, validation classification loss, and validation total loss for Dataset 1 are depicted in Figure 7b. The total validation loss is around 5, but the training loss is only approximately 0.78. This shows that the model may face overfitting, and it is unable to generalize well to new data. The same 416_INT8_4 model is applied to Dataset 2. Figures 8a presents the training confidence loss, training box loss, training classification loss, and training total loss when the model is trained using Dataset 2. The validation confidence loss, training box loss, training classification loss and training total loss are provided in Figure 8b. We observe that the training total loss is around 2.117, higher than the model trained by Dataset 1. However, the validation total loss is only 2.038, which is much lower than the mode trained by Dataset 1. Hence, Dataset 2 is used in the remaining experiments.

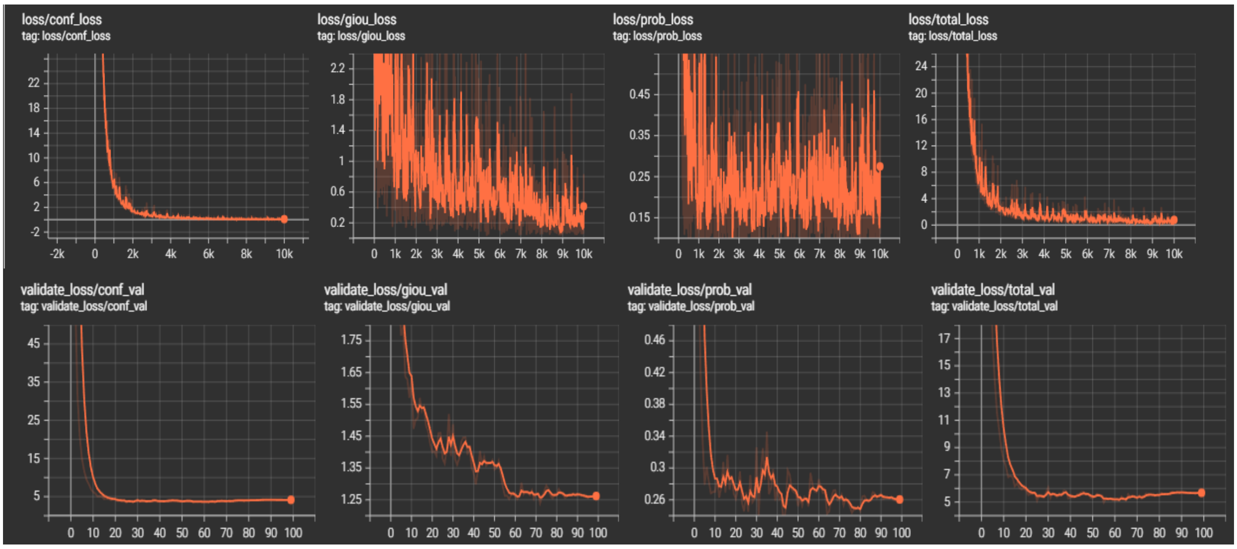


Figure 7 Confidence loss, box loss, classification loss and total loss for training (first row) and validation (second row) on Dataset 1

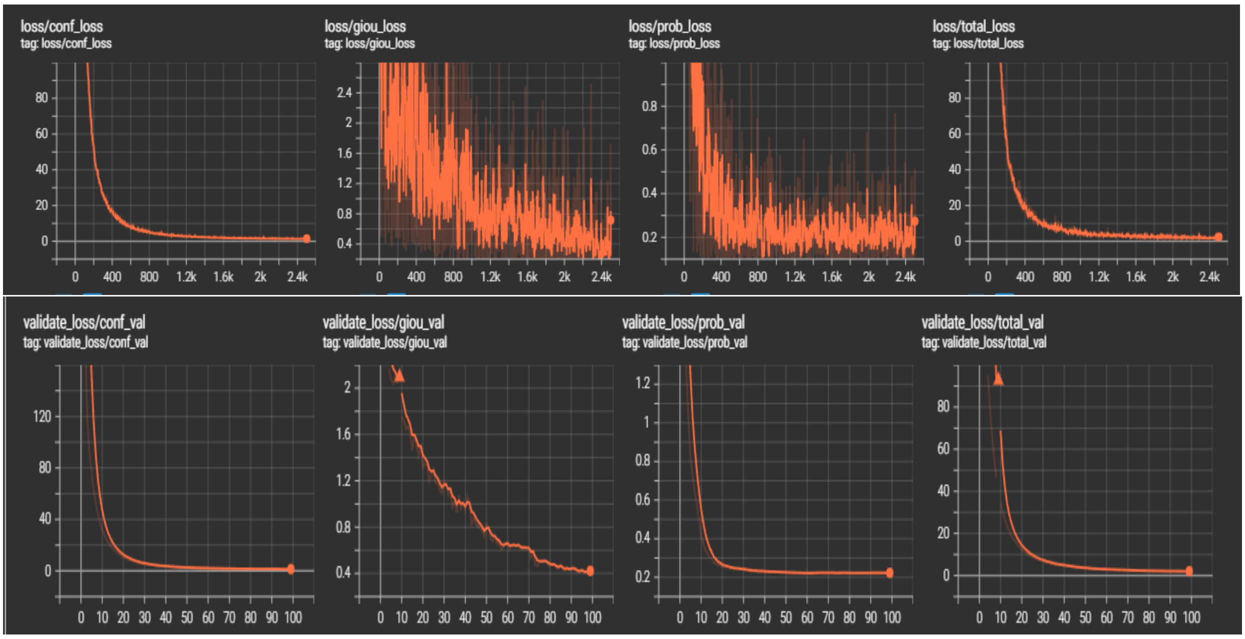
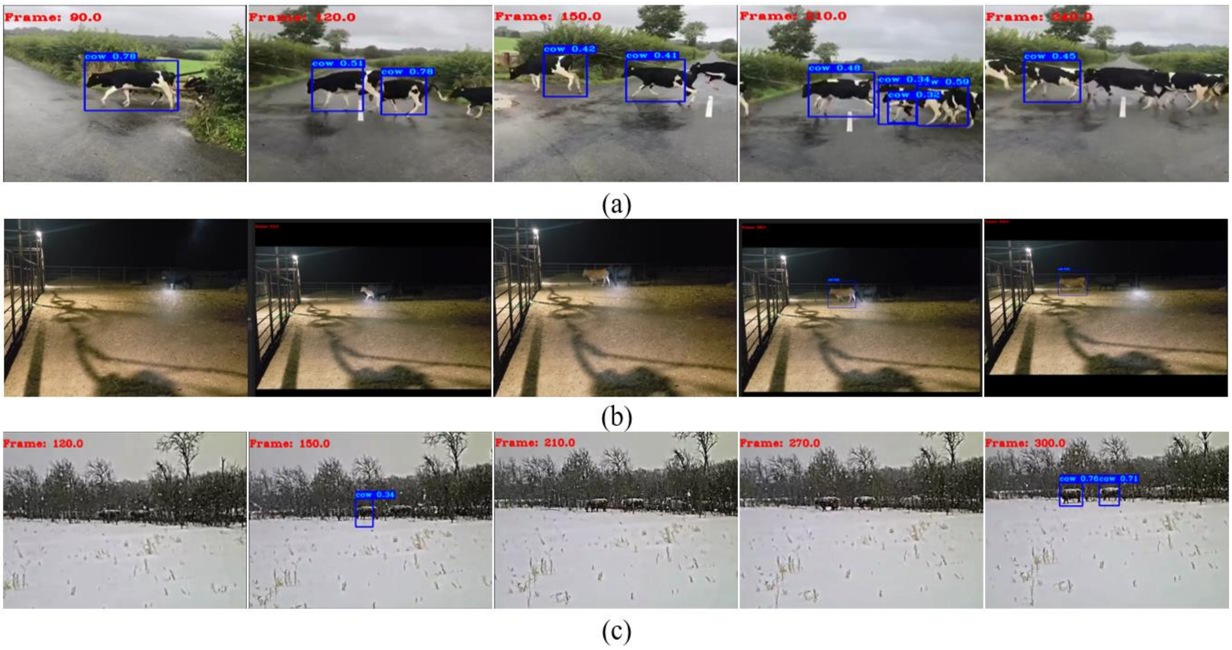


Figure 8 Confidence loss, box loss, classification loss and total loss for training (first row) and validation (second row) on Dataset 2

3.3.1.3. Prediction Results via Visualization

Due to its favourable performance, the 416_FP32_2 model is selected and loaded in Raspberry Pi to perform real time prediction. The model is tested using videos with different weather conditions such as snowing, sunny days or nights. These videos are collected from online resources such as Youtube. The reason to assess real-time prediction is that videos can evaluate the system's ability to detect effectively against dynamic factors such as weather and the brightness of surroundings. Figure 9 illustrate the cow detection results for the different weather conditions. Overall, the occurrences of cows in the various scenes can be detected accurately by the system.



(a)

(b)

(c)

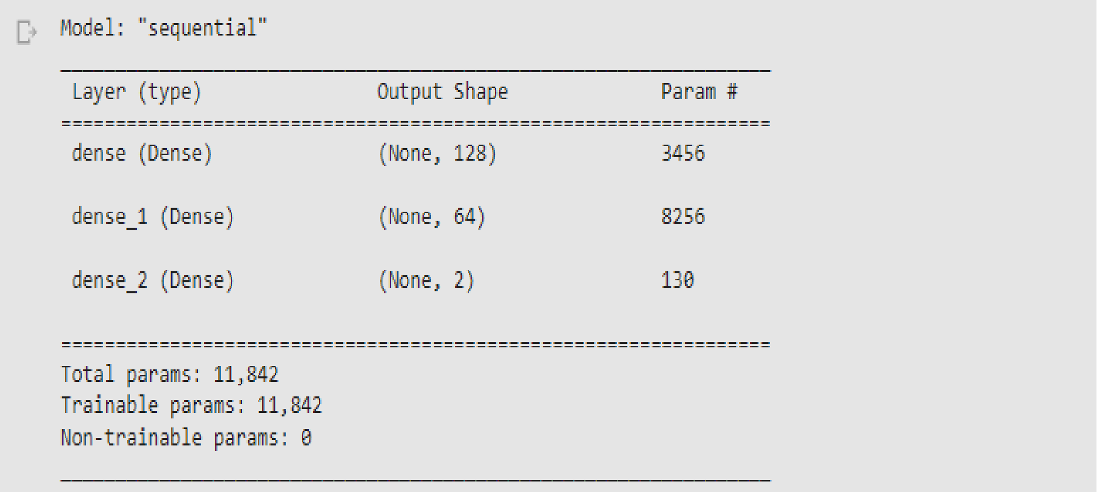
Figure 9 Samples detection results on video visualization at different weathers

3.3.2. Experiment Analysis of Sound Recognition

Next, experiments are conducted to evaluate the proposed sound recognition system. The hyperparameters used in the tests are as follows: Input Shape: 26; Epoch: 100; Model Design: Sequential model with same layers; Batch Size: 32, 64, 128; Optimizer: ADAM, SGD.

3.3.3 Experiment Results of Different Hyperparameters

In this section, the different hyper-parameters are tested to evaluate the performance of each setting. The model design used in experiments is depicted in Figure 10. A shallow neural network model consisting of 11, 842 parameters is used to avoid overfitting due to the small datasets.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 128)               3456

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 2)                 130

=================================================================
Total params: 11,842
Trainable params: 11,842
Non-trainable params: 0
_________________________________________________________________
```

Figure 10 The architecture of the sound recognition model

Figure 11a shows that the 26_SGD_128 model yields the worst performance because both the training and validation losses are the highest among the six combinations. On the other hand, 26_ADAM_32, 26_ADAM_64, and 26_ADAM_128 provide low training loss and validation loss. However, the 26_ADAM_64 model is able to achieve a 0.002 validation loss, which is the lowest among all. Therefore, the performance of 26_ADAM_64 is the best in the sound recognition model experiments. The precision, recall, F1-score, and accuracy of each model are presented in Figure 11b. We observe that 26_ADAM_64 and 26_ADAM_128 can obtain better performance, which can yield a score of 1 in all the evaluation metrics.

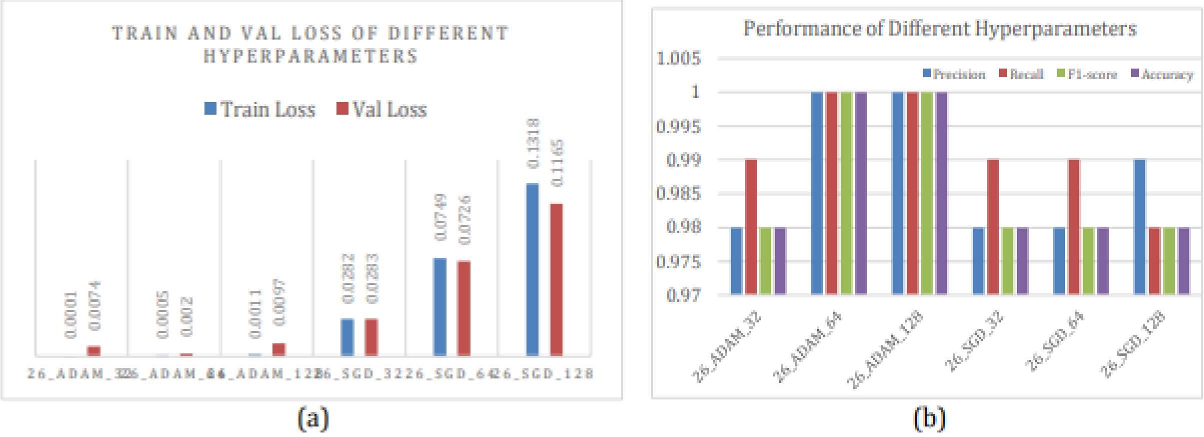


Figure 11 Training, validation, and performance of different hyperparameters

3.3.4. Fusion of visual and audio models

Table 3 depicts the performance of the proposed fusion approach. The testing samples are chosen randomly from the image and sound datasets. The fusion of visual and audio models undoubtedly achieves a good result of 100%. The good performance is expected as the audio model alone can obtain such an appealing outcome. The sound of cows can be easily differentiated from the other background noise/sound. Therefore, the experiment shows that the audio cue is an essential complement to the proposed cow detection system. The cow images may be easily affected by appearance changes due to lighting or pose modifications. Hence, the audio input resembles important cue to provide complementary information in the detection system.

Table 3 Performance of the proposed fusion method

Model	Accuracy
Visual model (YOLO)	98%
Audio model (ANN)	100%
Fusion of visual and audio models	100%

4. Conclusions

This paper presents a cow crossing detection alert system. A custom YOLO model and sound recognition models have been trained and are ready to be deployed into Raspberry Pi to monitor the condition of the roadway. Comprehensive experiments have been conducted to evaluate the robustness of the proposed system. Experimental results show that the proposed method can reliably be used for cow detection across different weather conditions. In the future, efforts will be devoted to improving the FPS of the detection such as using YOLO Tiny to train and detect the video because YOLO Tiny has a much smaller number of convolution layers than YOLOv3, so the hardware requirement is lower than YOLOv3, and increase the FPS of detection

Acknowledgements

This project is supported by TM R&D Fund, 2022 (Grant no. MMUE/220023). The authors would also like to thank the providers of the Kaggle cow and vehicle datasets, and also the UnrbanSound dataset for sharing their databases.

References

Chauhan, N.S., 2020. Audio Data Analysis Using Deep Learning with Python (Part 1). Kdnuggets. Available online at https://www.kdnuggets.com/2020/02/audio-data-analysis-deep-learning-python-part-1.html, Accessed on March 8, 2022

Jonny, Kriswanto, Toshio, M., 2021. Building Implementation Model of IoT and Big Data and Its Improvement. *International Journal of Technology*, Volume 12(5), pp. 1000–1008

Munaf, D.R., Piliang, Y.A., 2020. Sociotechnological Perspective on the Development of Lunar and Martian Infrastructures Made of Concrete Materials. *International Journal of Technology,* Volume 11(3), pp. 587–598

Redmon, J., Farhadi, A., 2018. YOLOv3: An Incremental Improvement. *ArXiv Preprint,* Volume 2018, pp. 1–6

Ross, A., Nandakumar, K., 2009. Fusion, Score-Level. *Encyclopedia of Biometrics*, Volume 2009, pp. 611–616

Zahari, N.Z., Fong, N.S., Cleophas, F.N., Rahim, S.A., 2021. The Potential of Pistia stratiotes in the Phytoremediation of Selected Heavy Metals from Simulated Wastewater. *International Journal of Technology,* Volume 12(3), pp. 613–624