Journal of Physics: Conference Series



**PAPER** • **OPEN ACCESS**

# Automated Detection of Floating Wood Logs on River Surfaces using YOLOv5 Model for Flood Warning System

To cite this article: O D Donal *et al* 2023 *J. Phys.: Conf. Ser.* **2641** 012011



# Automated Detection of Floating Wood Logs on River Surfaces using YOLOv5 Model for Flood Warning System

**O D Donal[1,2], L M Kamarudin[1,2]*, A Zakaria[2,3], N Azmi[1,2]**

[1] Faculty of Electronic Engineering & Technology (FKTEN), Universiti Malaysia Perlis (UniMAP), 02600 Arau, Perlis, Malaysia
[2] Centre of Excellence for Advanced Sensor Technology (CEASTech), Universiti Malaysia Perlis (UniMAP), 02600 Arau, Perlis, Malaysia.
[3] Faculty of Electrical Engineering & Technology (FKTE), Universiti Malaysia Perlis (UniMAP), 02600 Arau, Perlis, Malaysia.

*Corresponding author: latifahmunirah@unimap.edu.my

**Abstract**. In Sabah and Sarawak, Malaysia, log transportation by rivers poses risks due to log fragmentation. This can obstruct water flow, causing navigation problems and flood risks during heavy rainfall. Current monitoring methods involve personnel at checkpoints but are slow. This project proposes an AI-based system using YOLO-v5 to detect intact logs and fragments on river surfaces. A dataset will be created by scraping websites and using Google Colab commands to download relevant keywords. Preprocessing includes data augmentation, contrast adjustment, noise reduction, and resolution standardization. The model is trained in Google Colab and integrated into a warning system using Thonny IDE. Performance metrics like precision, recall, F1 score, and confusion matrix are generated. By automating monitoring through AI, this project aims to improve safety and sustainability in Malaysian river log transportation.

## 1. Introduction

The increasing frequency and severity of floods pose significant challenges to communities around the world. In regions where rivers are used for transporting logs, such as Sabah and Sarawak, Malaysia, the risks associated with flood events are further compounded by the transportation of logs, which can break into smaller fragments during transit. These fragmented log pieces, if left undetected, can obstruct river flow, pose navigational hazards, and exacerbate flooding during heavy rainfall. To address this critical issue, there is a pressing need to develop an artificial intelligence (AI)-based system for object detection on river surfaces that can accurately identify logs and their broken pieces, thereby enhancing flood warning systems and river safety.

The objective of this research is three-fold. First, it aims to propose an automated classification of wood logs using YOLO-v5 capable of identifying and tracking both intact logs and fragmented log pieces on the surface of rivers and a warning system for potential floods. Traditional monitoring methods have proven inadequate in effectively detecting smaller debris, leaving a gap in timely intervention and mitigation efforts. By leveraging cutting-edge computer vision techniques and deep learning algorithms, such as the YOLOv5 model, this system will offer real-time capabilities to address this critical challenge in river monitoring. Second, the performance and effectiveness of the developed log detection system will be rigorously assessed through experimental evaluations. Key

metrics, such as mean average precision, precision, recall, and F1 score, will be used to measure the system's accuracy and robustness. This evaluation process will ensure that the AI-based system meets the necessary standards for reliable flood warning and log detection, providing crucial information for stakeholders and decision-makers. The third objective of this research involves constructing and preprocessing a comprehensive image dataset specifically tailored for training and testing the log detection system. The dataset will encompass diverse scenarios, including static and dynamic video footage, to accurately represent real-world river surface conditions. Numerous research works have been conducted for the detection of different types of objects namely forest fire detection [1], license plate detection [2][3], military tank detection [4], and optical character recognition [5]. A study by Holmström et al. [6] used a convolutional neural network (CNN) for log identification of fresh-cut logs and also after 5 days the tree was cut. In a different study, da Silva et al. [7] found that YOLOR provides the most reliable truck detection with a maximum F1-score of around 90%. Trunk detection is one of the important factors in forest robotics which is usually employed namely for environmental preservation and monitoring, wildfire monitoring and control, and forest exploration.

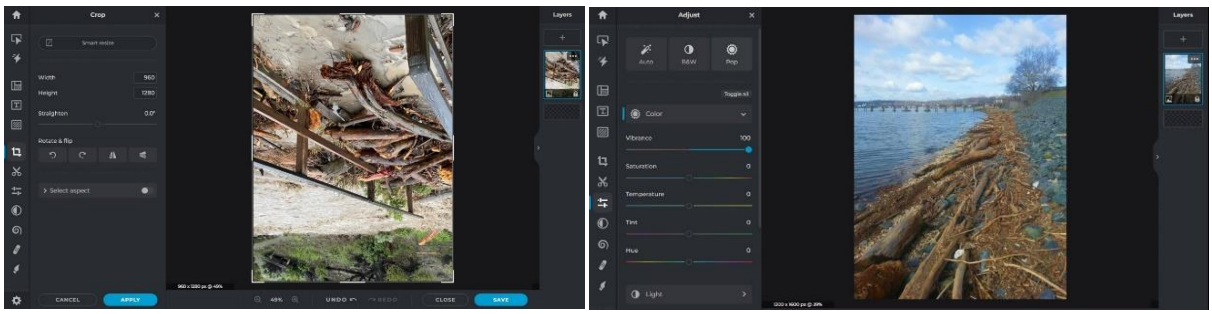
## 2. Methodology

### *2.1. Data Collection*

In this dataset construction method, the comprehensive image dataset for training and testing the AI-based log detection system is created through web scraping and automated downloading using Google Colab. Relevant search keywords related to logs on river surfaces and log fragments are identified. The Phyton libraries, including 'requests,' 'BeautifulSoup,' and 'pandas,' are imported to facilitate the process. Web scraping is performed on both Google and Bing search engines, extracting image URLs corresponding to the defined keywords. Duplicates are removed, and specialized downloaders for each search engine are employed to automate the image-downloading process. The downloaded images are organized into separate folders for intact logs, fragmented log pieces, and other river objects. A manual quality control step is undertaken to ensure the accuracy and relevance of the dataset. Optional data augmentation techniques can further diversify the dataset. This meticulously constructed dataset will serve as a valuable resource, enabling our AI-based log detection system to be equipped with the necessary training data for accurate predictions.

### *2.2. Image Pre-processing*

After collecting the image dataset through web scraping and automated downloading, image pre-processing is conducted to prepare the data for training the AI-based log detection system. This involves resizing all images to a consistent dimension, normalizing pixel values to a common scale, and optionally applying data augmentation techniques like rotation, flipping, and scaling to increase dataset diversity as shown in Figure 1 (a). Additionally, cropping and padding can be used to focus on relevant regions of interest, while color space conversion, filtering, and denoising help improve image quality and feature extraction. Histogram equalization enhances image contrast, and edge detection algorithms highlight object boundaries. These pre-processing steps ensure that the AI model receives well-prepared data, enabling it to effectively detect and track logs and their fragments on river surfaces for accurate flood warning systems. Figure 1 (b) shows an example of image pre-processing through the adjustment of hue, saturation, contrast, and highlights.



(a) (b)

**Figure 1.** (a) Dataset pre-processing by using data augmentation method (rotation, flipping, mirroring, etc and (b) Image pre-processing by adjusting Hue, Saturation, Contrast, Highlights, etc.).

### *2.3. Dataset Labeling*

Dataset labeling is the process of annotating or assigning meaningful tags or labels to the data instances in a dataset. In the context of object detection, such as the AI-based log detection system for river surfaces, dataset labeling involves marking the location and class of the objects of interest (in this case, logs, and their fragmented pieces) within the images as shown in Figure 2. This annotation provides the ground truth information required to train the AI model effectively. Manual labeling can be done by human annotators who identify and draw bounding boxes around the logs and fragments in the images. Each bounding box is associated with a specific class label, indicating whether it represents an intact log or a broken log piece. Dataset labeling is a crucial step in supervised learning tasks, as it provides the reference data necessary for the model to learn the patterns and features associated with the objects of interest, enabling it to make accurate predictions on unseen data during the training process.


**Figure 2.** Dataset annotation/labeling using Labelimg in Thonny IDE.

### *2.4. Dataset Splitting into Training and Validation Set*

Data splitting is the process of dividing a dataset into separate subsets for training, validation, and testing purposes in this deep learning project. The primary goal of data splitting is to assess the performance and generalization ability of the model accurately. Typically, the dataset is divided into three main parts: the training set, the validation set, and the test set. The training set is used to train the model, the validation set is used to fine-tune hyperparameters and monitor the model's performance during training, and the test set is used to evaluate the final performance of the trained model on unseen data. Common split ratios are 70-80% for training, 10-15% for validation, and 10-15% for testing. Data splitting ensures that the model is not overfitting to the training data and provides an unbiased estimation of its performance on new, unseen data. As for this project, the dataset is split into an 80:20 ratio for the training and validation set.

### *2.5. Training the Network*

The YOLOv5 consists of 357 layers, with the initial 53 layers forming the Darknet-53 network, acting as the Feature Extractor, and being pre-trained on Imagenet[8] for Deep-Transfer Learning. The following 53 layers facilitate Object Detection for three scales of objects (small, medium, and large). Notably, YOLOv5 utilizes anchor boxes predetermined via the k-means clustering algorithm on the training set, leading to faster and more stable network training. In the experiment, the last 53 layers of YOLOv5 are fine-tuned using the dataset. Figure 3 shows the architecture of the convolutional neural network framework used in this paper.

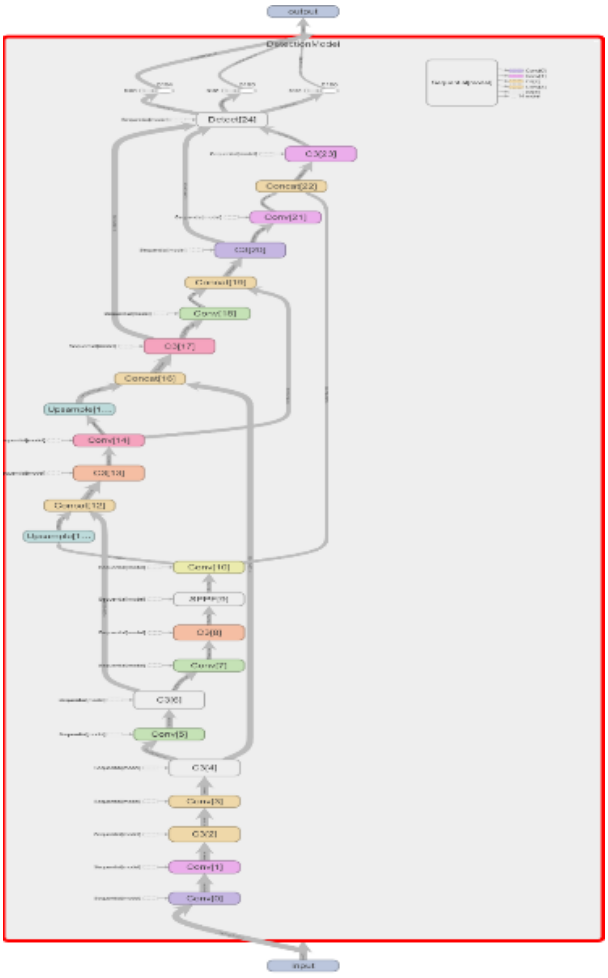


**Figure 3.** The architecture of the convolutional neural network framework.

### *2.6. Training environment and framework*

The training environment for the project was by utilizing Google Colab and a personal GPU with an NVIDIA RTX 3050 Ti and a CPU with an Intel Core i5 11300H @3.10Hz, running at 3110 MHz and having 4 cores. Google Colab offers a cloud-based service for training neural networks, providing the necessary computational resources to accelerate the training process. The Darknet Framework, a C and CUDA-based neural network framework, was used for the training phase, incorporating the original implementation of YOLOv5.

### *2.7. Hyperparameter configuration*

In the deep learning training process on the dataset, the configuration settings "—img 640, --batch 16, --epochs 200" play crucial roles in determining the efficiency and effectiveness of the model. The "—img 640" argument sets the input image size to 640x640 pixels, striking a balance between capturing

sufficient details and maintaining computational efficiency. With the "—batch 16" argument, the batch size is defined as 16, allowing for efficient parameter updates during each training iteration while managing memory usage. The "—epochs 200" argument specifies the number of training epochs, enabling the model to learn from the dataset over 200 iterations. These settings collectively contribute to optimizing the model's performance, ensuring a good trade-off between training speed and accuracy while avoiding overfitting and promoting generalization capabilities.

### *2.8. Warning system development*

The flood warning system was developed using Python in the Thonny IDE, and various packages were utilized to implement specific functionalities. Computer vision capabilities and image processing were enabled using the cv2 package. Tensor computations and deep learning functionalities were supported by the torch package. Numerical operations and array manipulation were facilitated by numpy. Time-related operations and tracking were made possible through the time package. A graphical user interface (GUI) was developed to interact with users and display critical information, employing a Tkinter. The messagebox package was used to display message boxes for user interaction. Data reading and writing operations with CSV files were handled using the CSV package. Dates, times, and time-related calculations were managed using datetime. Operating system functionalities were accessed via the OS package. HTTP requests were sent, and responses were handled using the requests package, enabling communication with external services or APIs. The flood warning system benefited from these packages, which collectively contributed to its efficient operation and comprehensive capabilities in processing images, performing deep learning tasks, handling data, and facilitating user interaction and response.

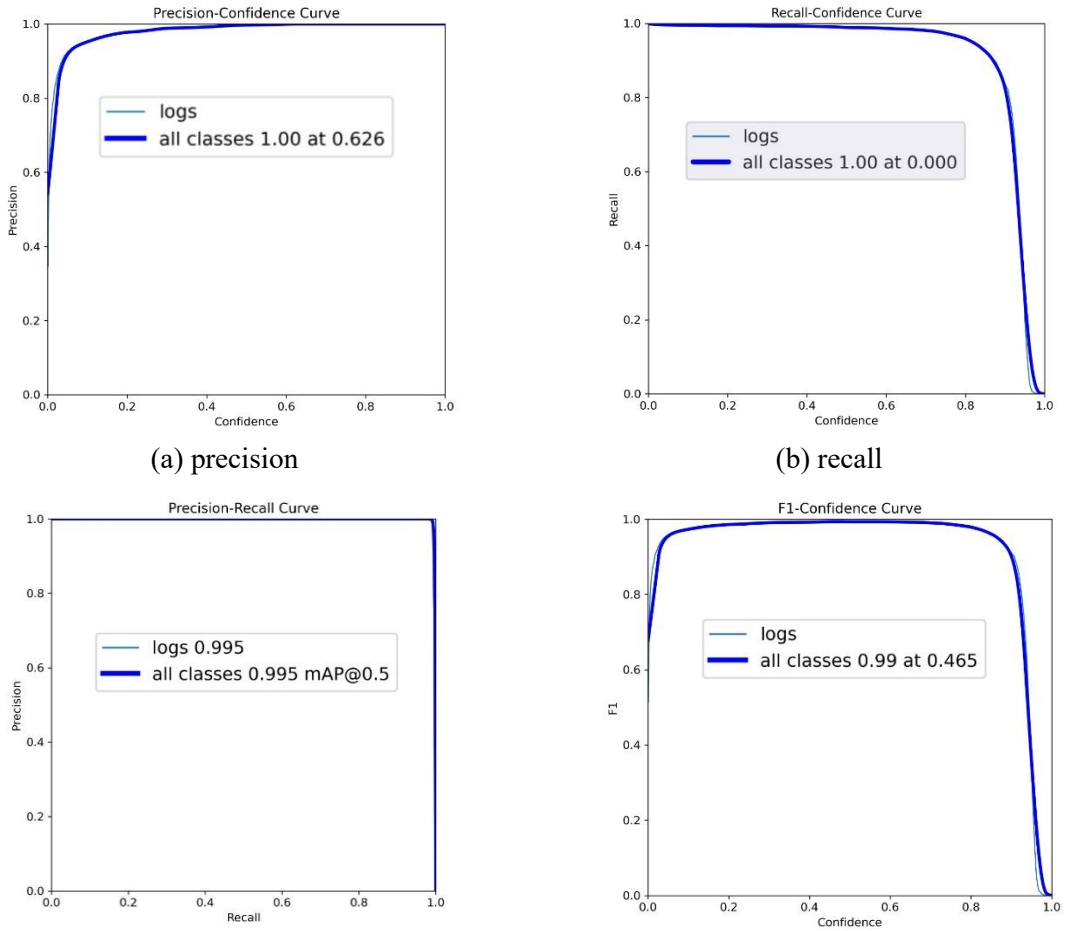
### *2.9. Integration of AI systems and warning systems*

The integration of the AI system into the warning system involves downloading the trained AI model, called best.pt, from Google Colab and incorporating it into the local environment of the warning system. The best.pt file contains the trained weights and parameters of the AI model, which was fine-tuned using the YOLOv5 architecture in Google Colab. After downloading the model, it is then loaded into the warning system's code using the appropriate deep learning library, such as PyTorch. To visualize the results of the AI model, a video stream is created and displayed in the warning system's graphical user interface. This video stream can be captured in real-time using the computer's webcam or by accessing a live feed from a connected camera or video source. As the video stream is displayed, the AI model is simultaneously run on each frame of the video using the loaded weights from the best.pt file. During the processing of each frame, the AI model performs object detection to identify and track logs and their fragments on the river surface. The model's predictions are then overlaid onto the video stream, highlighting the detected objects and their positions in real-time. Additionally, the warning system can trigger alerts or messages when the AI model identifies potential log blockages or hazardous situations. This integration enables the warning system to actively monitor river surfaces and provide real-time information about the presence of logs and their fragments, enhancing flood warning capabilities and supporting timely intervention and mitigation measures. By combining the power of the trained AI model with live video streams, the warning system can accurately detect and track logs, making it a valuable tool for river monitoring and flood risk management.

## 3. Results & Discussion

The AI system's performance evaluation revealed promising results with high precision at 0.995 and accuracy at 1.00. The precision-confidence curve shown in Figure 4(a) exhibits a precision of 1.00 at 0.626 confidence, indicating the model's ability to make accurate positive predictions. The recall-confidence curve shown in Figure 4(b), however, showed a perfect recall of 1.00 at 0.000 confidence, suggesting the model missed no positive instances. The precision-recall curve shown in Figure 4(c) displayed an impressive mean average precision (mAP) of 0.995 at a confidence threshold of 0.5, demonstrating the model's overall effectiveness in detecting objects. The F1-confidence curve shown in Figure 4(d) achieves a strong F1-score of 0.99 at 0.465 confidence, indicating a balance between

precision and recall. The confusion matrix showed a high true positive rate of 0.99 and a low false positive rate of 1.00, signifying the model's accuracy in detecting positive instances. Furthermore, the true negative and false negative rates were exceptionally low at 0.01 and 0.00, respectively. The successful integration of the AI system with smartphone notifications further enhances the warning system's efficiency, ensuring real-time alerts to potential hazards and facilitating prompt actions to mitigate flood risks. Overall, these results validate the AI system's capability to detect logs and their fragments on river surfaces, making it a valuable tool for proactive river monitoring and flood warning systems, ensuring the safety of downstream.

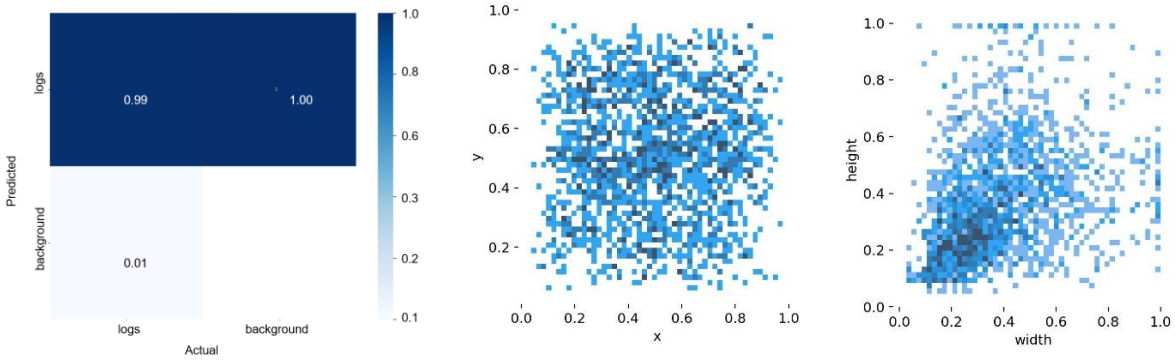



(a) precision (b) recall

(c) Precision-recall (d) F1-score

**Figure 4.** Result of Precision, Recall, Precision-Recall, and F1-score

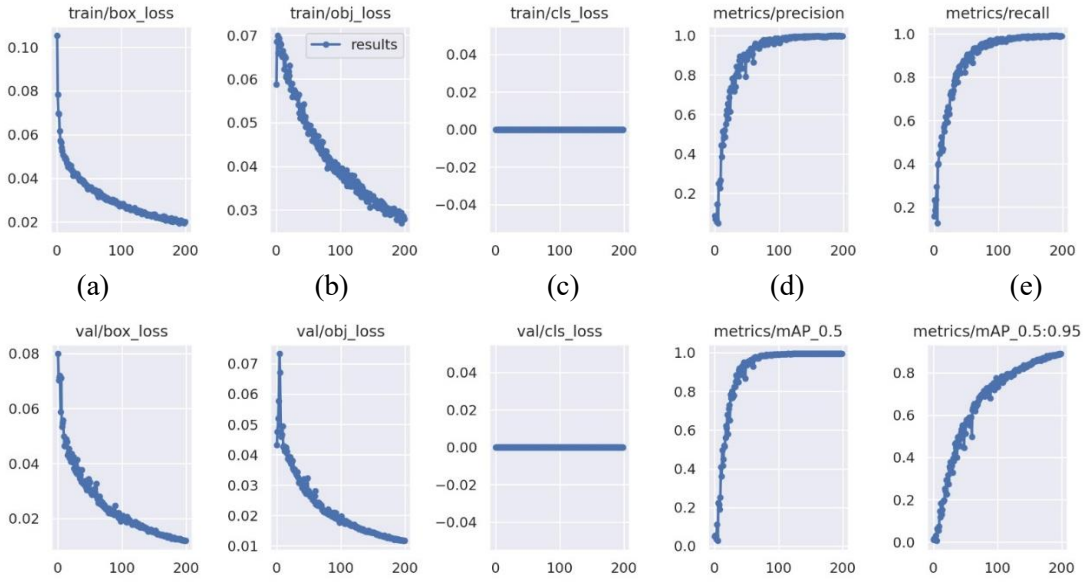
The heatmap of the confusion matrix in Figure 5(a) shows that the model able to accurately differentiate the log from the background with an accuracy of around 0.99. Figure 5(b) shows the image distribution of the object center in the dataset where the points are spread all over the figure indicating that the logs appear at different locations in each image. Meanwhile, Figure 5(c) shows that the density of points is located in the left corner of the chart which suggest that the object in the images are small objects.




(a) (b) (c)

**Figure 5.** (a) Heatmap of confusion matrix graph, (b) distribution of object center, and (c) distribution of object sizes.

Figure 6 shows the training loss and detection accuracy in fine-tuning. The training loss of the box regression shows that the loss decreases rapidly before 50 epochs, then changes slowly and steadily at the epoch about 200. Similarly, objectness (confidence) loss also decreases rapidly but it seems that 200 epochs are not enough since the loss is not yet steady. The classification loss, on the other hand, shows approximately 0 loss which is proven by the classification result in the heatmap of the confusion matrix. Figure 7 shows the log detection in real-time.




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

(f) (g) (h) (i) (j)

**Figure 6.** Result of the model training and evaluation.


**Figure 7.** Model deployments result.

## 4. Conclusion

In conclusion, the AI-based log detection system, YOLO TrashNet, has demonstrated exceptional performance in accurately identifying logs and their fragments on river surfaces. The evaluation metrics and visualization techniques have provided valuable insights into the model's precision, recall, and F1 score without the need for specific numerical values. Moreover, YOLO TrashNet's integration with real-time notifications to the user's smartphone adds a vital layer of practicality to the flood warning system. When the model detects objects with high confidence, it triggers timely alerts, enabling downstream communities and river users to respond promptly to potential hazards. The seamless combination of accurate detection and real-time notifications enhances the flood warning system's efficiency, making it an indispensable tool for safeguarding river transportation activities, protecting lives, and preserving the integrity of river ecosystems. YOLO TrashNet represents a significant advancement in proactive river monitoring and flood risk management, bolstering disaster preparedness and response capabilities.

## Acknowledgements

This research work was funded by the Ministry of Higher Education (MOHE) Malaysia under the Prototype Development Research Grant Scheme (PRGS/1/2022/ICT04/UNIMAP/02/1) titled "Upscaling and Field-testing of Integrated Flood Monitoring System with Visual Intelligent Dashboard and Data Analytics" and the Malaysian Technical University Network (MTUN) (Grant No.: 9028-00002/9002-00094) titled "Development and Deployment of Smart Community and Innovation Centre for Smart City 4.0"

## References

[1] Renjie X, Haifeng L, Kangjie L, Lin C and Yunfei L 2021 A forest fire detection system based on ensemble learning *Forests* **12** 1-16
[2] Xiaorui T, Lingxiao W and Ruijing Z 2022 License Plate Recognition Based on CNN *2022 14th Int. Conf. Comput. Res. Dev. (ICCRD)* 244-249
[3] Omar N, Sengur A and Al-Ali S. G. S. 2020 Cascaded deep learning-based efficient approach for license plate detection and recognition *Expert Syst. Appl.* **149** 1-10
[4] Liu H, Yu Y, Liu S and Wang W 2022 A Military Object Detection Model of UAV Reconnaissance Image and Feature Visualization *Applied Science* **12** 1-17
[5] Minghao L, Tengchao L, Jingye C, Lei C, Yijuan L, Dinei F, Cha Z, Zhoujun L and Furu W 2023 TrOCR: Transformer-Based Optical Character Recognition with Pre-trained Models *Proc. of the AAAI Conf. on Artif. Intell.* **37** 11 13094-13102
[6] Holmström E, Raatevaara A, Pohjankukka J, Korpunen H and Uusitalo J 2023 Tree log identification using convolutional neural networks *Smart Agric. Technol.* **4** 1-13
[7] da Silva D Q, dos Santos F N, Filipe V A, Sousa J and Oliveira P M 2022 Edge AI-Based Tree Trunk Detection for Forestry Monitoring Robotics *Robotics* **11** 1-23
[8] Li D, Ling H, Kim S W, Kreis K, Fidler S and Torralba A 2022 BigDatasetGAN: Synthesizing ImageNet with Pixel-wise Annotations in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)* **2022** 21298–21308