EVOLUTION IN ELECTRICAL AND ELECTRONIC ENGINEERING
e-ISSN: 2756-8458

EEEE

Vol. 5 No. 1 (2024) 459-469
https://publisher.uthm.edu.my/periodicals/index.php/eeee

Real-time Pothole Detection using Deep Learning

Zakwan Syukri Mohd Shah[1], Mohd Norzali Haji Mohd[1]*

[1] *Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja, 86400, Johor MALAYSIA*

*Corresponding Author: norzali@uthm.edu.my
DOI: https://doi.org/10.30880/eeee.2024.05.01.058

Article Info

Received: 11 January 2024
Accepted: 15 March 2024
Available online: 30 April 2024

Keywords

Deep Learning, Pothole Detection, YOLOv5

Abstract

The maintenance of infrastructure and the safety of roads are seriously threatened by potholes. Potholes can be quickly located by authorities, allowing for quicker repairs and safer commuter routes. This study uses the YOLO (You Only Look Once) version 5 deep learning model to demonstrate a real-time pothole detection system. Compared to traditional pothole detection approaches, deep learning model detection uses less human power which will contribute less error too. The suggested method makes use of YOLOv5's high accuracy and efficiency to find potholes in recorded or live video streams. The model detection achieves a mAP (mean average precision) of 76% with 1200 annotated public dataset were trained and deployed on YOLOv5 medium sized model. Data collection, preprocessing, model training, and inference are all part of the system's end-to-end pipeline. A diverse dataset of road images annotated with pothole locations is used to train the YOLOv5 model. High detection accuracy and real-time performance are achieved by the proposed approach, as shown by experimental results. Under various lighting conditions, weather conditions, and road types, the system's robustness is assessed. The results show for real-time pothole detection function can be run using a Raspberry Pi but with 2-3 FPS (frame per second) only. With the current FPS, difficulty may occur for real world deployments in high-speed scenario and need an optimization to boost edge computing performance.

1. Introduction

Without a sufficient network of roads, no modern society could be considered complete. But over time, potholes can form on the road due to deteriorating conditions, which increases the risk of accidents, vehicle damage, and higher maintenance expenses. Today, potholes are a major problem in many developed nations. The topography of the road is uneven, there is a lot of traffic and flooding, low-quality materials are used, and there are natural disasters [1]. Frequently difficult to locate, but it's crucial to locate them fast to prevent accidents and reduce the cost of road maintenance. Traditional pothole detection methods, like visual surveys and manual inspection, have drawbacks as well. They are labor-intensive and prone to error. 52,295 potholes were discovered in the state of Selangor and the city of Kuala Lumpur alone, according to data from the Malaysian road traffic navigation app Waze [2].

Machine learning's deep learning branch has shown impressive results in identifying objects and patterns in pictures and videos. The use of deep learning methods for pothole identification has garnered increasing attention in the last several years. According to reference [1], this researcher used three different ways to examine road abnormalities detection: computer vision-based, vibration/sensory method, and 3D construction. Apart from that, a system called Pothole Finder was created employing the YOLO (You Only Look Once) technique in real-time object recognition [2]. Yolo processes an image in one pass, as opposed to traditional techniques that divide detection into multiple phases.

Deep learning has shown promise in detecting potholes in images. The goal is to create a computer-vision system that can accurately identify potholes in real time using video or image data from road or vehicle cameras. Potholes of varying sizes, depths, lighting, and weather conditions ought to be detectable by YOLO algorithms. Several studies have been conducted on pothole detection using deep learning techniques. One recent study [3], shows 95% as the best mean average precision for YOLOv5 with a training loss of 0.02. To reach this level of precision, the author used a total required dataset of 1330 images split into two parts for model training and testing with labels of each image [3]. Another study by [4] proposed a deep learning-based approach that used YOLOv5 with various trained models. The YOLOv5m6 model training has the highest precision compared to others with an mAP@0.5 value of 80.8%.

1.1 Comparison performance of Yolov5

According to [5], YOLOv5s was trained for 350 epochs and deployed on the Xavier NX mobile platform. It achieved a precision of 0.9917, recall of 0.9809, mAP@0.5 of 0.9895, and mAP0.5:0.95 of 0.8272. [6] It runs at 20 frames per second on Xavier NX. produced low precision, accuracy, and mAP values of 0.4, 0.4, and 0.36, respectively. In [7], the author used 465 images in training and achieved 77% accuracy. Note that the accuracy of the YOLOv5 algorithm can be improved by using a larger training dataset. In [8], the author proposed YOLOv5s-M, which achieved high confidence in detecting potholes of small areas with values of 0.782 for precision, 0.721 for recall, and 0.419 for mAP@0.5:0.95. In summary, DenseSPH-YOLOv5 is a high-performance damage detection model that outperforms some of the best detection models in terms of both detection accuracy and speed [9]. Table 1 shows the comparison performance of Yolov5.

Table 1 *Comparison performance of Yolov5*

Pothole Detection Performance					
Work	Model	Precision	Accuracy	mAP@0.5	mAP@0.5:0.95
[5]	Yolov5s	0.90	0.82	0.9895	0.8272
[6]	Yolov5	0.4	0.4	0.36	0.20
[7]	Yolov5m	0.86	0.77	0.734	0.707
[8]	YOLOv5s-M	0.782	0.721	0.798	0.509
[9]	DenseSPH-YOLOv5	0.8951	0.76.25	0.85	0.8521

2. Methodology

The methodology for detecting potholes using deep learning is explained in this section. The dataset is gathered to train the model, which is the general flow of the suggested methodology. Pothole images are collected from various sources and annotated using tools like labelImg. Image augmentation is typically required to improve the performance of deep models to construct a reliable classifier with a small amount of training data. Images are augmented in a variety of ways, including horizontal flipping, vertical flipping, and rotation, to expand the dataset. A sizable dataset is needed to train pothole detection models more effectively.

YOLOv5 is used as the model for pothole detection. Several pretrained models have been tested on the pre-processed dataset in this study. A deep learning model is then trained using the dataset to find potholes in the road's surface. After training, the model is saved and tested using a different image dataset. The input video stream is run by microcontroller such as Raspberry Pi 5 Model B, which can be using a single camera like the Raspberry Pi Camera or webcam. Fig. 1 shows the pothole detection flowchart.

2.1 Dataset Preparation and Annotation

Datasets are gathered from public datasets such as Google Archive and Roboflow. Annotation process occurred in LabelImg software tool to create bounding boxes of pothole class. Once already done the annotation, the files can be saved as XML in PASCAL VOC format using ImageNet format also supports YOLO and createML formats. A total of 1200 images with split ratio of 0.8:0.2 represent training and valid dataset. Fig. 2 shows images annotation.

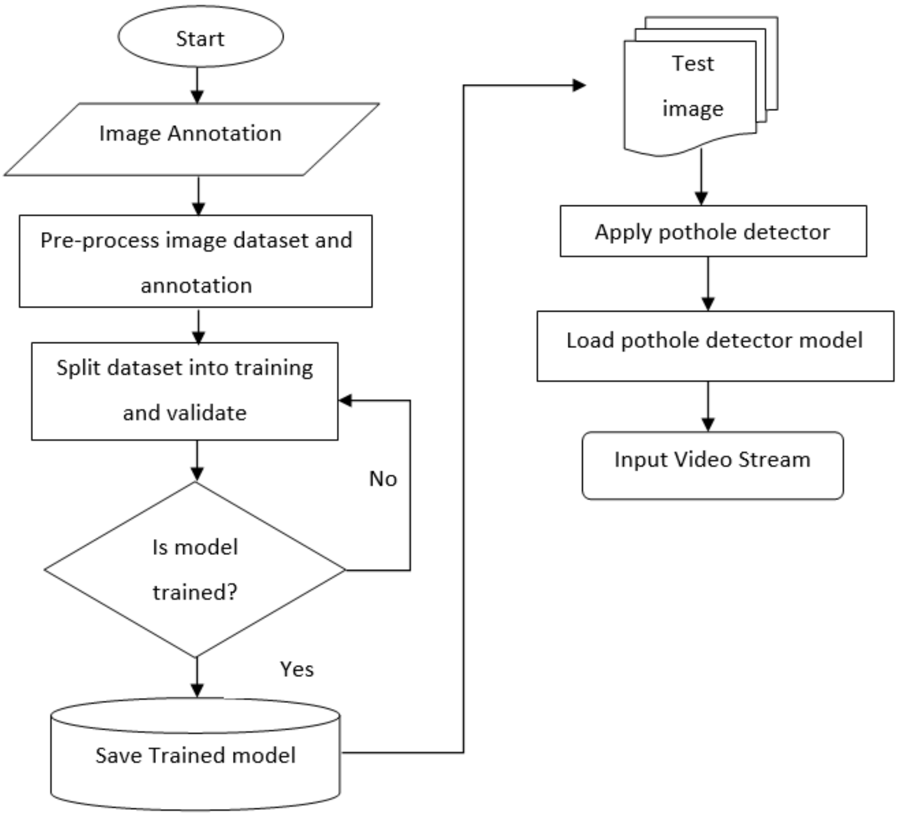


Fig. 1 *Pothole detection flowchart*

Fig. 2 *Images Annotation*

2.2 Deep Learning Model Training

The process is managed by a Jupyter notebook. Since this model is currently being trained, there won't be many processes that need to be regulated. The following steps would be included in the entire implementation process:

- Downloading the YOLO V5 model files
- Preparing the annotated files
- Preparing the train, validation, and test sets
- Implementing the training process
- Executing the inference process using the trained model

2.3 Real-time Implementation

Fig. 3 and Fig. 4 show the hardware being used in this system.

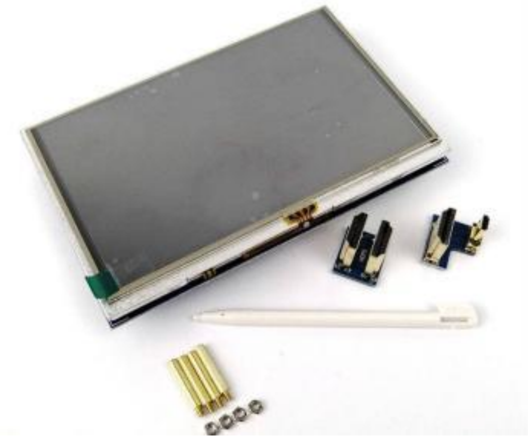


Fig. 3 *5-inch LCD HDMI display*

Hardware Description for 5-inch LCD HDMI display:
1. Resolution: 800 x 480 px.
2. With backlight control switch (slide switch at the back of LCD).
3. Working Voltage/Current: DC 5V / 1A.
4. Power source: Default from Raspberry Pi GPIO, alternatively USB Micro B on the back of display. next to the HDMI socket.
5. It's suitable for Raspberry Pi series.
6. You can download the official firmware and drivers to use it.
7. And you can use integrated firmware to use it (the firmware with driver, is very easy).
8. Dimension: 120mm x 74mm x 7mm

Fig. 4 *Raspberry Pi 5 model B PCB layout*

Hardware description for Raspberry Pi 5 model B PCB layout:
1. SOC: Broadcom BCM2712.
2. CPU: 2.4GHz quad-core 64-bit Arm Cortex-A76 CPU.
3. RAM: LPDDR4X-4267 SDRAM (4GB).
4. WIFI: Dual-band 802.11ac wireless LAN (2.4GHz and 5GHz) and Bluetooth 5.0.
5. Ethernet: Gigabit Ethernet, with PoE+ support (requires separate PoE+ HAT).
6. Thermal management: Yes
7. Video: Yes - Dual 4Kp60 HDMI® display output with HDR support.
8. Audio: Yes
9. USB 2.0: 4 ports (2.0 x2 and 3.0x2)
10. GPIO: 40-pin
11. Power: 5V/5A or minimal of 5V/3A DC power input
12. Operating system support: Linux and Unix
13. Storage: PCIe 2.0 x1 interface for fast peripherals (requires separate M.2 HAT or another adapter)

2.4 Software Implementation

The Raspberry Pi 5 Model B supports deep learning frameworks based on Python, such as PyTorch. These frameworks, which run YOLOv5, offer high-level APIs and tools for developing and deploying deep learning models. Real-time video can be carried out using the Raspberry Pi 5 Model B in conjunction with USB cameras or the Raspberry Pi Camera Module. The deep learning model then processes the captured frames to identify potholes. The Raspberry Pi can handle overall system management and communication tasks.

3. Results and Discussion

3.1 Training Result

The training process was carried out for 50 epochs which composed single class of pothole using 1200 images with split ratio of 80% for train and 20% for validation as shown in Table 2. Mean Average Precision (mAP) is a metric used to evaluate object detection models and are calculated over recall values from 0 to 1. mAP formula is based on the following sub metrics:

Precision measures the accuracy of predictions. It indicates the proportion of true positives (TP) (correctly identified positives) among all positive predictions, including false positives (FP) (incorrectly identified positives).

$$Precision = \frac{T_p}{T_p + F_p} \tag{1}$$

Measures the completeness of predictions. It indicates the proportion of true positives (correctly identified positives) among all actual positives, including false negatives (FN) (missed positives). A high recall value (closer to 1) suggests that the model misses few objects, striving for comprehensiveness.

$$Recall = \frac{T_p}{T_p + F_n} \tag{2}$$

While precision and recall are valuable, they often trade off against each other. A high precision model might miss many true positives, while a high recall model might include many false positives. mAP takes this trade-off into account. It calculates the average precision across all possible recall levels for each object class in the dataset.

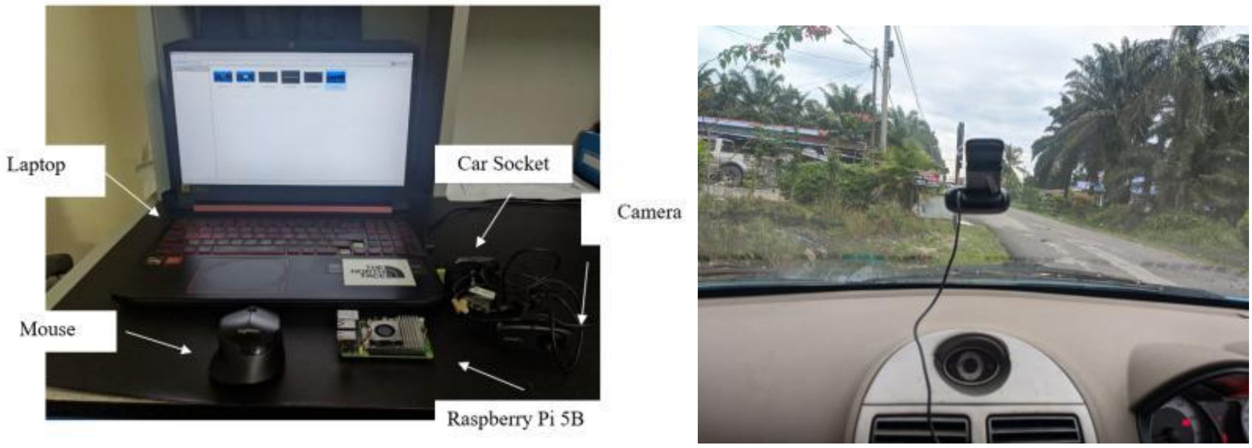
$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \tag{3}$$

Table 2 *Train result for 40-50 epoch*

epoch	precision	recall	mAP_0.5	mAP_0.5:0.95
40	0.80004	0.73025	0.75988	0.40725
41	0.74084	0.66871	0.71108	0.3549
42	0.7444	0.64267	0.72431	0.37566
43	0.73221	0.6401	0.6751	0.3273
44	0.76358	0.68123	0.74947	0.39497
45	0.78683	0.6832	0.73538	0.37848
46	0.81055	0.67352	0.74184	0.38756
47	0.76613	0.69666	0.73234	0.38151
48	0.76371	0.69409	0.73337	0.37753
49	0.80704	0.66324	0.72961	0.36873

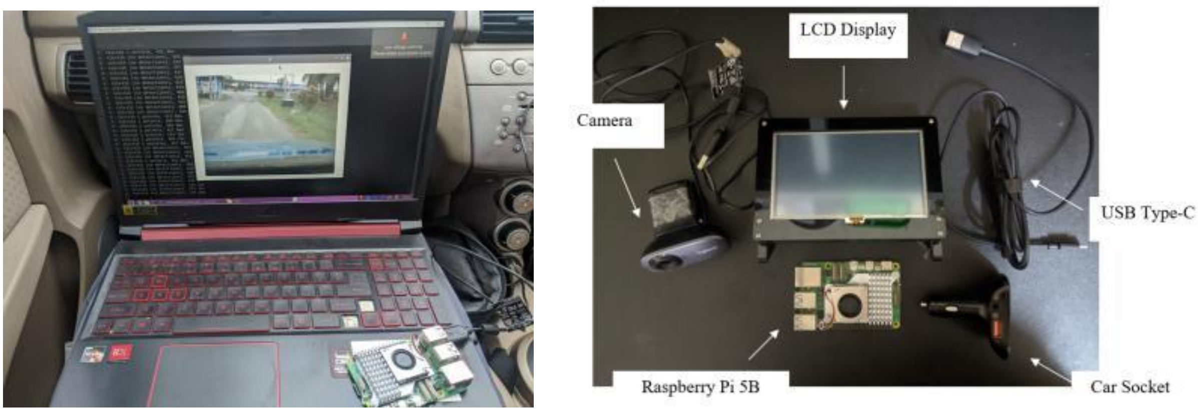
Epoch 40 as shown in Table 2 achieves the highest overall values among 50 epochs with precision of 0.80004 proves the model predicts good positive results. For recall the model correctly identifies 0.73025 which means it achieves about 73% of actual positive cases. 0.75988 for mAP 0.5 provides balance between precision and recall and it also demonstrates a good ability to make accurate positive prediction.

3.2 Hardware Setup for Real-time Prototype



(a) **(b)**

Fig. 5 *(a) Set up Raspberry Pi 5 using laptop as a display; (b) Set up camera on windscreen*



(a) **(b)**

Fig. 6 *(a) Setup of raspberry pi and laptop in a car; (b) Setup of raspberry pi using 5-inch LCD HDMI display*

(a) **(b)**

Fig. 7 *(a) Raspberry pi and webcam position; (b) Setup raspberry pi using 5-inch LCD HDMI display*

3.3 Real-time Pothole Detector

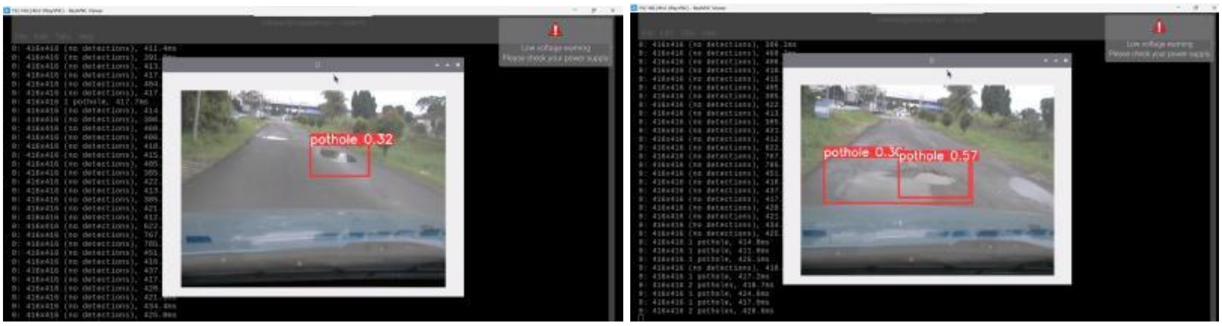
The reason for using low image size is to get the output display run with faster inference time and lowering latency. The bounding boxes appear to be labelled as “pothole” with accuracy rate of that detected potholes. The FPS value can be calculated by formula shown below:

$$number\ of\ FPS\ =\frac{1}{inference\ time}$$

$$number\ of\ FPS=\frac{1}{417\ \times 10^{-3}} \tag{4}$$

$$number\ of\ FPS=2.398$$

Average FPS values were within range of 2-3 FPS with inference time around 300 milliseconds to 400 milliseconds. The accuracy rate shows low value when the pothole detected further away from the car compared to the near one as shown in Figure 8 (a) with 57% and Figure 2.8 (b) with 32%. The bounding box with 57% accurate rate is higher because it detects smaller confidently and this is due to dataset having many similar dimensions annotation.



(a) (b)

Fig. 8 *(a) Screenshot of pothole detector on VNC viewer; (b) Screenshot of pothole detector on VNC viewer*

3.4 Outcome of YOLOv5m Implementation

After successfully trained detection model of YOLOv5 medium version with pothole dataset, the trained model will be calling as inference file. The inference file contains best epoch generated among 50 epochs being trained. For inference, put those weights along with a conf specifying model confidence (higher confidence required makes less predictions), and a inference source. Source can accept a directory of images, individual images, video files, and also a device's webcam port. For this results in Figure 9 (a) and (b), it shows that the picture with large and clear pothole will have more accuracy compared to smaller one and less clear visible. To improve this issue, the dataset can be improved by annotate more images with dynamic sizes, coordinations of pothole in images and with different kind of lighting sources.

(a) (b)

Fig. 9 *Figure description (a)*, *(b) Result after prediction from saved trained model*

The Fig. 10 shows a graph of a F1-Confidence curve, which is a line plot that shows the relationship between the confidence level of a model and its accuracy. In this case, the model is YOLOv5m, with the annotated pothole dataset used. The x-axis of the graph is the confidence level, which ranges from 0.0 to 1.0. The y-axis of the graph is the F1 score, which is a measure of the model's accuracy. The F1 score is a harmonic mean of the model's precision and recall.

The curve in the graph shows that the F1 score of the model increases as the confidence level increases. This means that the model is more accurate when it is more confident in its predictions. The graph also shows the F1 score for the only class which is pothole at a confidence level of 0.432. In this case, the F1 score is 0.76. This means that the model is 76% accurate at identifying objects with a confidence level of 43.2%.

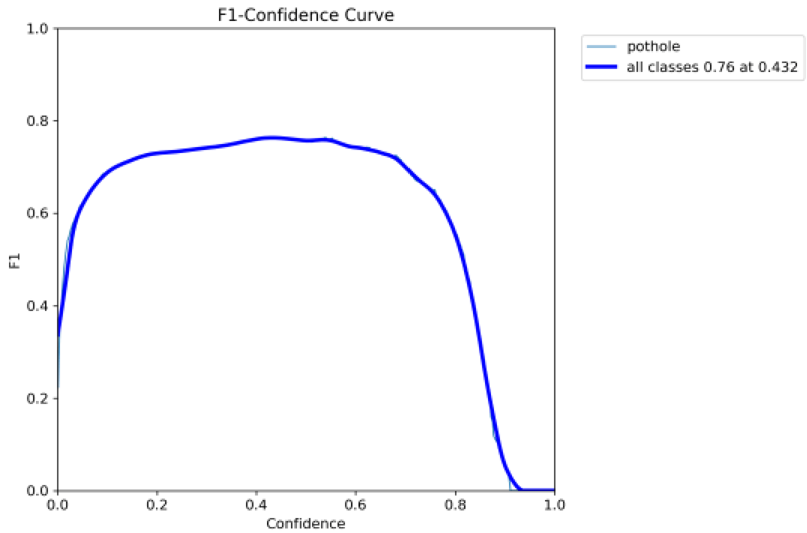


Fig. 10 *F1-Confidence Curve*

Fig. 11 shows confusion matrix for classification task of deep learning pothole model. The rows of the matrix represent the true labels of the data points, while the columns represent the predicted labels. The numbers in the cells of the matrix represent the number of data points that belong to a certain true class and were predicted to belong to a certain predicted class.

To evaluate the performance of the model, some key metrics need to be calculated. Accuracy: This is the proportion of data points that were correctly classified. In this case, the accuracy is (77 + 23) / 100 = 100%. Precision is the proportion of data points that the model predicted to be potholes that were potholes. In this case, the precision is 77 / 100 = 77%. Recall is the proportion of data points that are potholes that the model correctly classified as potholes. In this case, the recall is 77 / 77 = 100%.

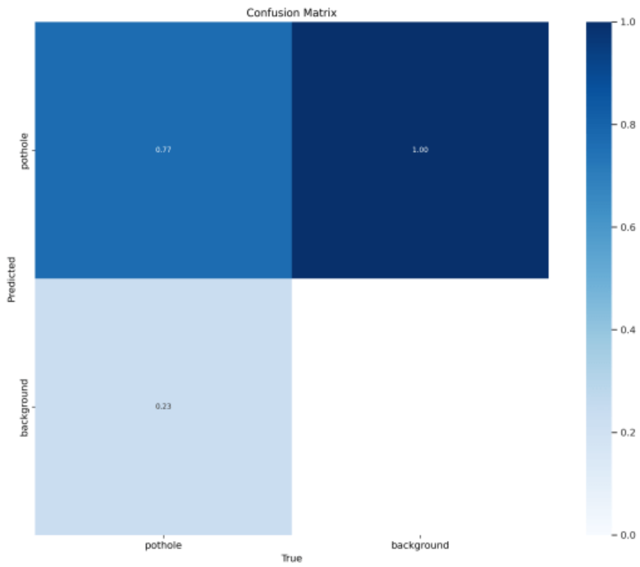


Fig. 11 *Confusion Matrix*

Fig. 12 shows precision-confidence curve for a pothole detection model used to visualize a model's precision (the ability to correctly identify potholes) and its confidence (the ability to identify all potholes). The confidence level of 0.904, the model has a precision of 1.0 (perfect) for that class. The model's performance will be better when the curve is higher.

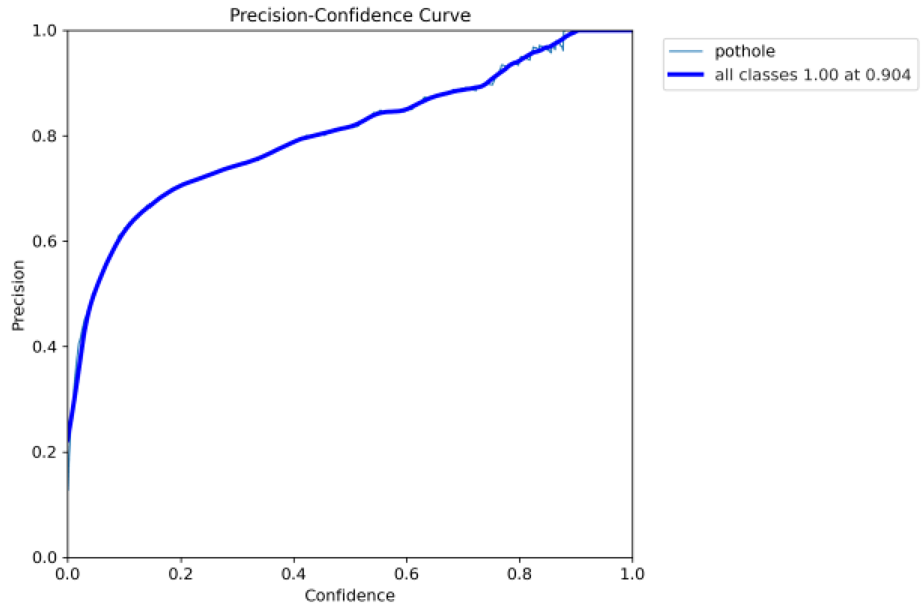


Fig. 12 *Precision-Confidence Curve*

Fig. 13 shows precision is the fraction of positive predictions that are correct, and recall is the fraction of true positives that are correctly identified. precision-recall curve would be a straight line at the top left of the graph, meaning that the model would correctly identify all potholes (high recall) and never make any false positives (high precision). As the model is tuned to be more sensitive to potholes (to increase recall), it will also start to identify more false positives (which will decrease precision).

The specific curve in the image shows that the model has a precision of 0.76 at a recall of 0.5. For example, of every 10 potholes identified by the model, 7.6 are potholes and 2.4 are false positives. The higher the area under curve, the better the model's performance.

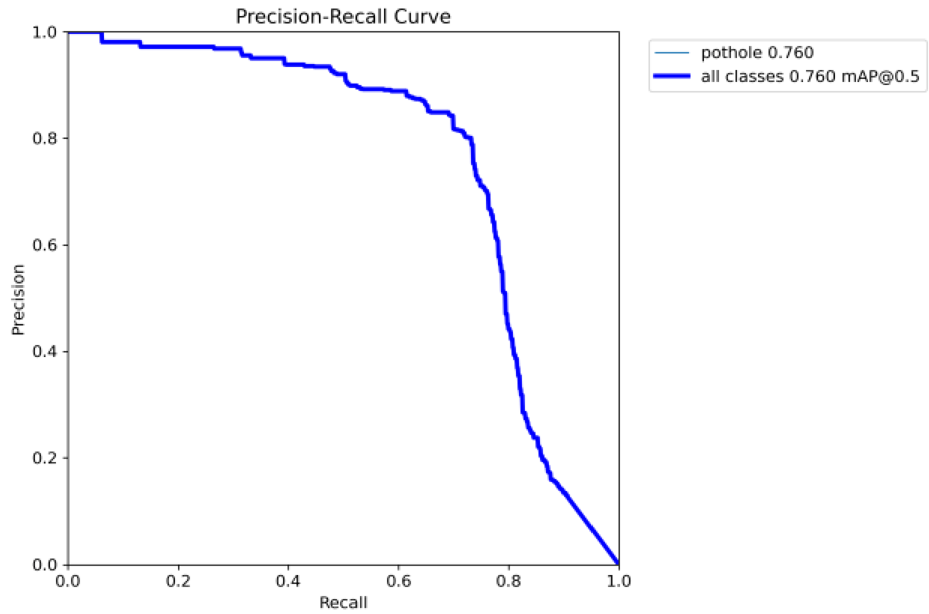


Fig. 13 *Precision-Recall Curve*

Fig. 14 shows a graph of a recall-confidence curve for potholes class of 0.87 at 0.000. The curve itself shows how the recall changes as the model's confidence increases. In this case, the curve starts at 0 recall for a confidence of 0 and gradually increases to reach a maximum recall of 0.87 at a confidence of 0.000. This means that the model achieves its highest recall rate (identifying 87% of actual potholes) when it is very confident in its predictions (confidence score of 0.000).

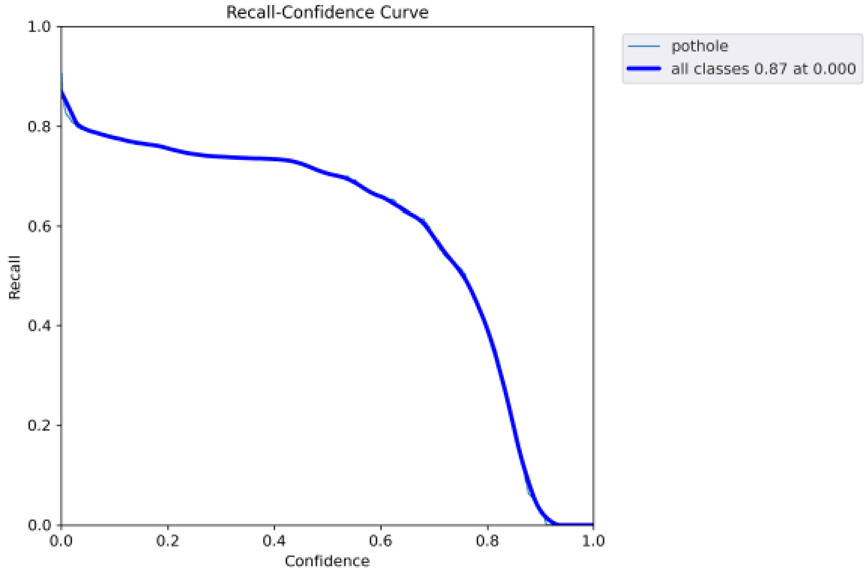


Fig. 14 *Recall-Confidence Curve*

4. Conclusion

The Raspberry Pi 5 B successfully executes a pothole detection project using the Yolov5m model trained on preexisting data. Yolov5m, chosen for its stability and compact inference file, optimizes deep learning model deployment on the Pi by minimizing file size. This efficiency ensures rapid startup, responsiveness, and smooth operation without overheating or excessive power consumption. Given the Pi's processing limitations, the default Pytorch inference file is converted to OpenVino format, streamlining the computational graph and accelerating inference performance.

Precision reduction from 32-bit to 8-bit maintains accuracy while achieving an impressive 3 FPS, in stark contrast to Pytorch's suboptimal sub-1 FPS performance. Power is sourced from a car socket supplying a maximum 3.5A at 5V, meeting the 3A requirement for optimal processor functionality. Adjusting display camera resolutions further enhances inference speed. Overall, this project adeptly addresses Raspberry Pi constraints, delivering a highly effective pothole detection system.

Acknowledgement

The author would like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussien Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the completing of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Zakwan, Mohd Norzali;* ***data collection:*** *Zakwan;* ***analysis and interpretation of results:*** *Zakwan, Mohd Norzali;* ***draft manuscript preparation:*** *Zakwan. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Hijji, M., et al., *6G Connected Vehicle Framework to Support Intelligent Road Maintenance Using Deep Learning Data Fusion.* IEEE Transactions on Intelligent Transportation Systems, 2023: p. 1-10.

[2] Sathya, R., B. Saleena, and B. Prakash. *Pothole Detection Using YOLOv3 Model*. in *2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*. 2023.

[3] Asad, M.H., et al., *Pothole Detection Using Deep Learning: A Real-Time and AI-on-the-Edge Perspective.* Advances in Civil Engineering, 2022. **2022**: p. 9221211.

[4] Her, A.Y., et al. *Real-time pothole detection system on vehicle using improved YOLOv5 in Malaysia*. in *IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society*. 2022.

[5] Achirei, S.D., et al. *Pothole Detection for Visually Impaired Assistance*. in *2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP)*. 2021.

[6] Darapaneni, N., et al. *Pothole Detection Using Advanced Neural Networks*. in *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. 2021.

[7] Deepa, D., et al. *Pothole Detection using Roboflow Convolutional Neural Networks*. in *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*. 2023.

[8] Miao Ren, Xianfeng Zhang, Xiao Chen, Bo Zhou, Ziyuan Feng, YOLOv5s-M: A deep learning network model for road pavement damage detection from urban street-view imagery, International Journal of Applied Earth Observation and Geoinformation, Volume 120, 2023, 103335, ISSN 1569-8432, https://doi.org/10.1016/j.jag.2023.103335.

[9] Roy, A.M. and J. Bhaduri, *DenseSPH-YOLOv5: An automated damage detection model based on DenseNet and Swin-Transformer prediction head-enabled YOLOv5 with attention mechanism.* Advanced Engineering Informatics, 2023. **56**: p. 102007.