International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(Research Article)

Check for updates

Computer vision for asphalt cracks detction using YOLOv5

Kenechukwu Sylvanus Anigbogu [1, *], Samuel Ochai Audu-war [2], Tochukwu Sunday Belonwu [1], Okwuchukwu Ejike Chukwuogo [1] and Emmanuel Chibogu Asogwa [1]

[1] *Department of Computer Sciences, Nnamdi Azikiwe University, Awka, Anambra State, Nigeria.*

[2] *Department of Computer Science, Benue State Polytechnic, Ugbokolo, Benue State, Nigeria.*

International Journal of Science and Research Archive, 2023, 10(01), 163–179

Publication history: Received on 16 July 2023; revised on 03 September 2023; accepted on 06 September 2023

Article DOI: https://doi.org/10.30574/ijsra.2023.10.1.0693

Abstract

Recent studies have shown that researchers have proposed various techniques for Pothole detection using data collected from different parts of the world. Automating pothole detection will go a long way in providing safe driving for road users and intelligent transportation systems. This is not only necessary to guarantee safe and adequate performance, but also to adjust to the drivers' needs, potentiate their acceptability, and ultimately meet drivers' preferences in bad roads. This paper presents a computer vision model that assists drivers by detecting and predicting potholes while on the road to curb road accidents. The datasets used in this research were potholes images extracted from kaggle which were classified into two; potholes and normal roads. The object detection algorithm that was used to evaluate the model is YOLO 5. The results from the parallel testing provided good results in detecting and predicting normal roads and potholes. The predicted values were all positive. The two classifiers were all detected perfectly in while testing without being perverse. The system presents its predicted value in percentage, therefore showing the level of adherence to each of the classes detected.

Keywords: Safe driving; Object detection; YOLO 5; Asphalt cracks detection.

1. Introduction

Road cracks and potholes are types of defects in the pavements that can disturb the safety and quality of the roads. The similarity are much as both means a bad occurrences on asphalt that can lead to deep death traps on the roads. Therefore in this research this two terms will be interwoven. A pothole is a natural cave or a hollow on the road surface formed as a result of erosion or aging of asphalt [1]. Potholes pose a lot of dangers for road transport users in many developing countries, especially in Nigeria. The task of maintaining roads and removing these road anomalies is an expensive and tedious one, due to the nature of landmass and climate conditions in Nigeria. It is reported that pothole is the second largest cause of accidents in Nigeria apart from overspeeding and reckless driving with annual reported accidents surpassing 45% [2]. The problem of potholes in Nigeria cannot be eradicated completely by the government but rather how to manage it and drive safely. The roads have been a concern of authorities to avoid unwanted circumstances. These roads are vulnerable to scenarios such as traffic load, weather conditions, age, poor material used for construction, and miserable drainage system, exhibiting two major road failures such as cracks and potholes. Potholes are essentially concave-shaped depressions in the road surface that require attention as they induce awful circumstances such as accidents, unpleasant driving experiences, and malfunctioning of vehicles. Potholes should be dealt with on a priority basis to minimize their contribution towards unfortunate scenarios.

According to the prediction made by WHO (World Health Organization), road accidents will become the fifth leading cause of death in 2030 [3]. The significance of potholes created conspicuous interest for the researchers of the civil community. The developing nations use manual inspection methods to recognize the potholes leading to inaccurate

* Corresponding author: Kenechukwu Sylvanus Anigbogu

estimation as it is highly dependent on individual experience. These manual inspection methods require human interventions that are time consuming and costly. Many technical solutions exist for pothole detection such as scanning based with 3D reconstruction [4], vibration sensor based [5], thermal imaging [6], and computer vision based [7].

2. Material and methods

In this section, we described the materials and methods used for the development of an intelligent mobile application for pothole detection.

2.1. Data Collection

The dataset used in this work were extracted from an online data source known as Kaggle. This dataset consists of images for road potholes along with the annotations. The feature of these datasets i.e., class value has two possible values which are pothole and normal, these driving events are the class labels. The dataset containing a total of 700 000 images with a total size of 10.1GB was used in this work.

2.2. Model Selection

We adopted the Yolo 5 model for this work because it improves object detection by taking advantage of spatiotemporal features and maintaining frames per second above 30. Its timely handgun detection is a crucial problem to improve public safety.

2.3. Data Preprocessing

The local dataset is preprocessed by applying basic image processing techniques such as normalization, resizing, and thresholding. The images are resized to 64x64 pixels to improve the performance of the model. The dataset was already annotated on extraction from kaggle. Figure 1a, b, c and d contain sample images with labels using labelling tool.

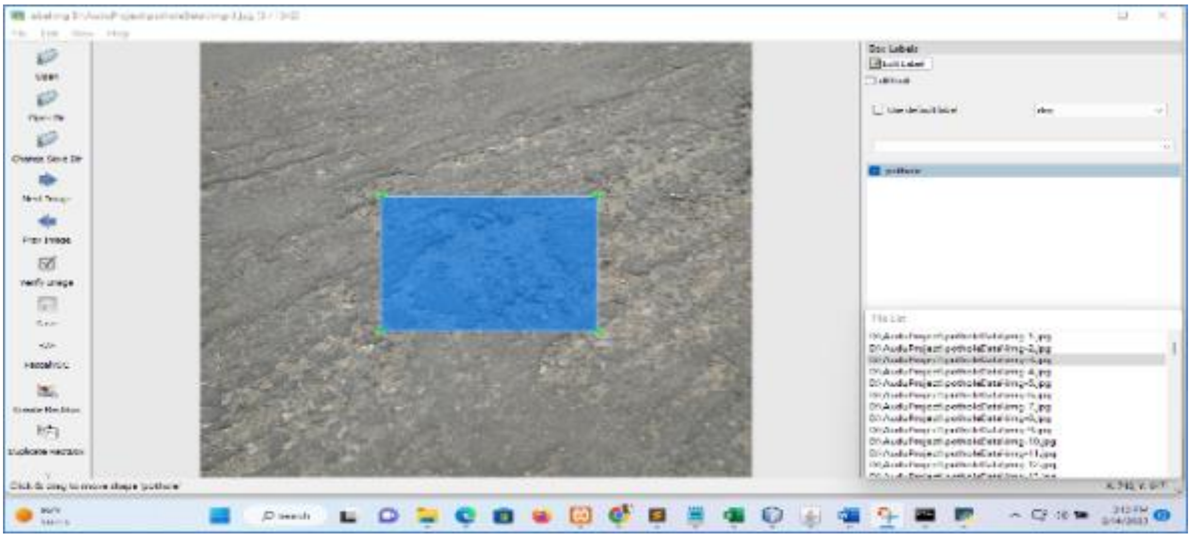


Figure 1a Sample Image Label for Potholes

Figure 1b Sample Image Label for Potholes

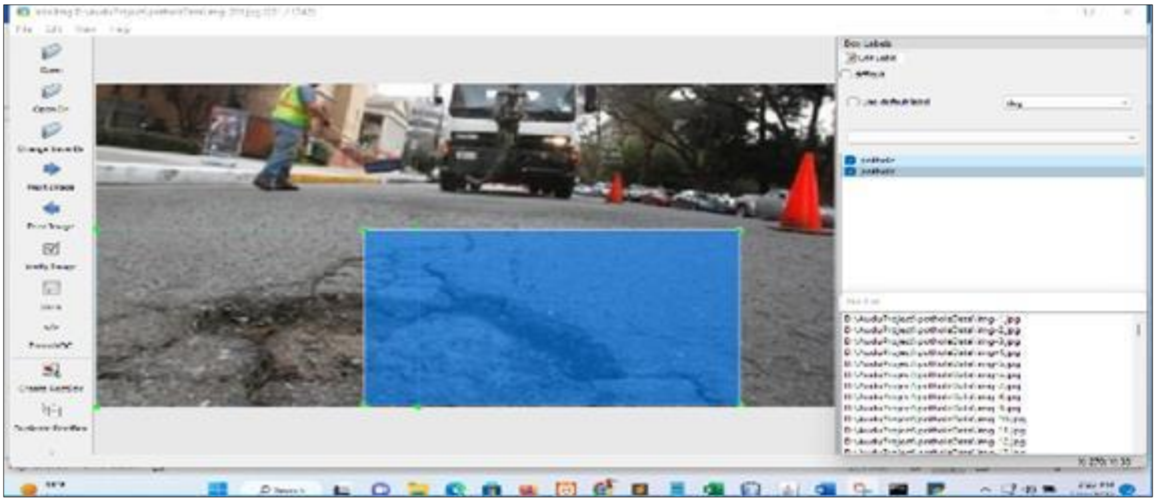


Figure 1c Sample Image Label for Potholes

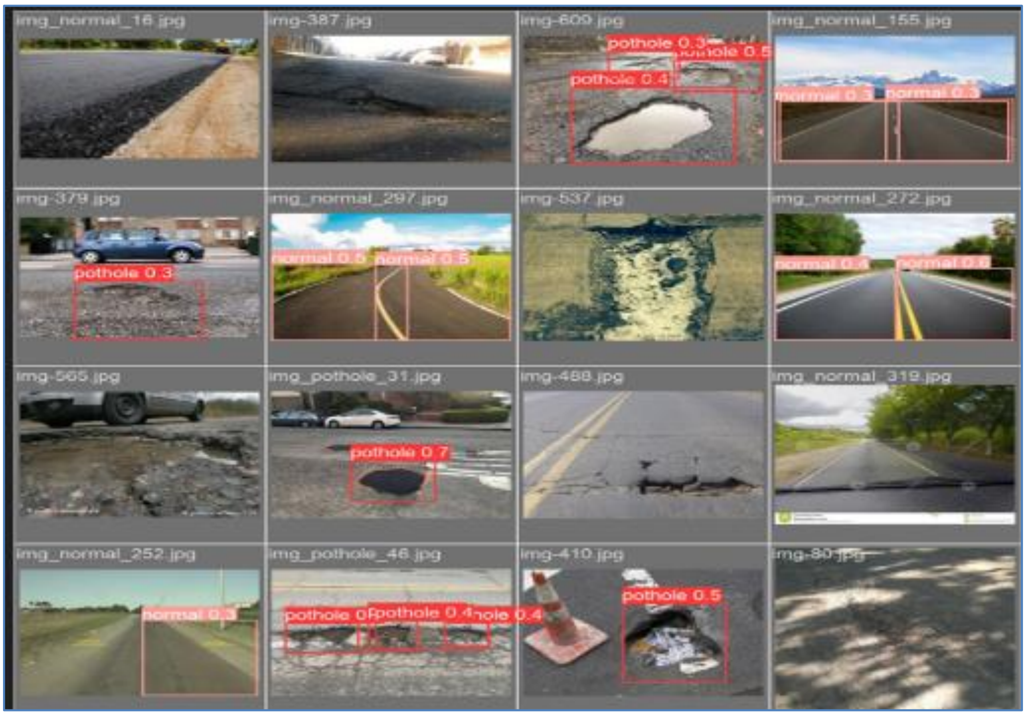
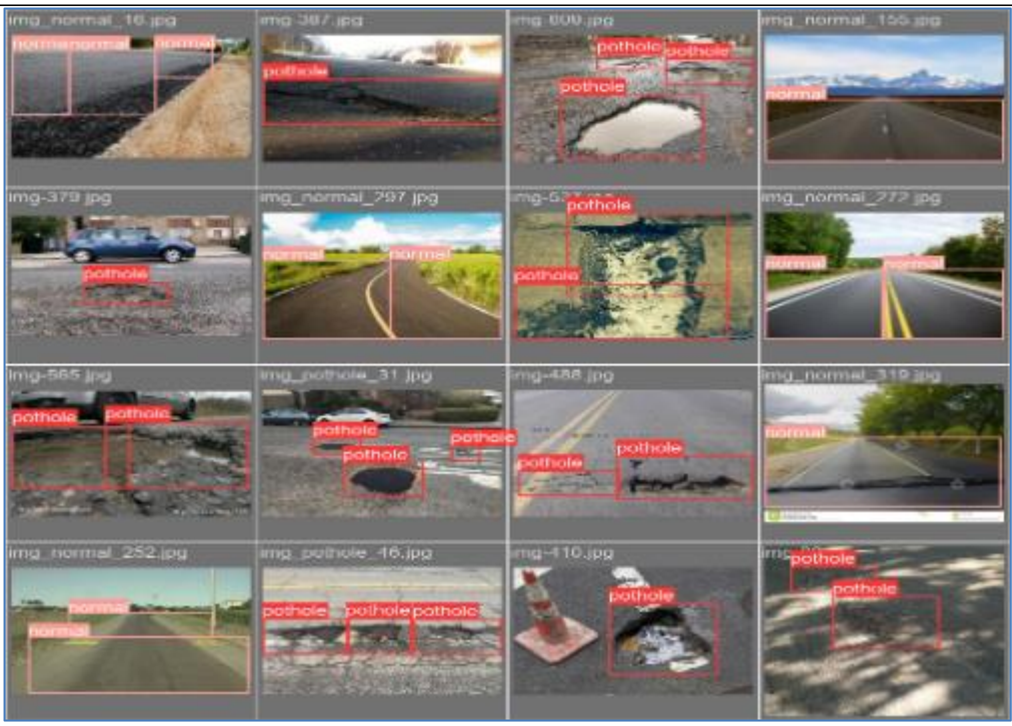
Figure 1d Label Image for normal road

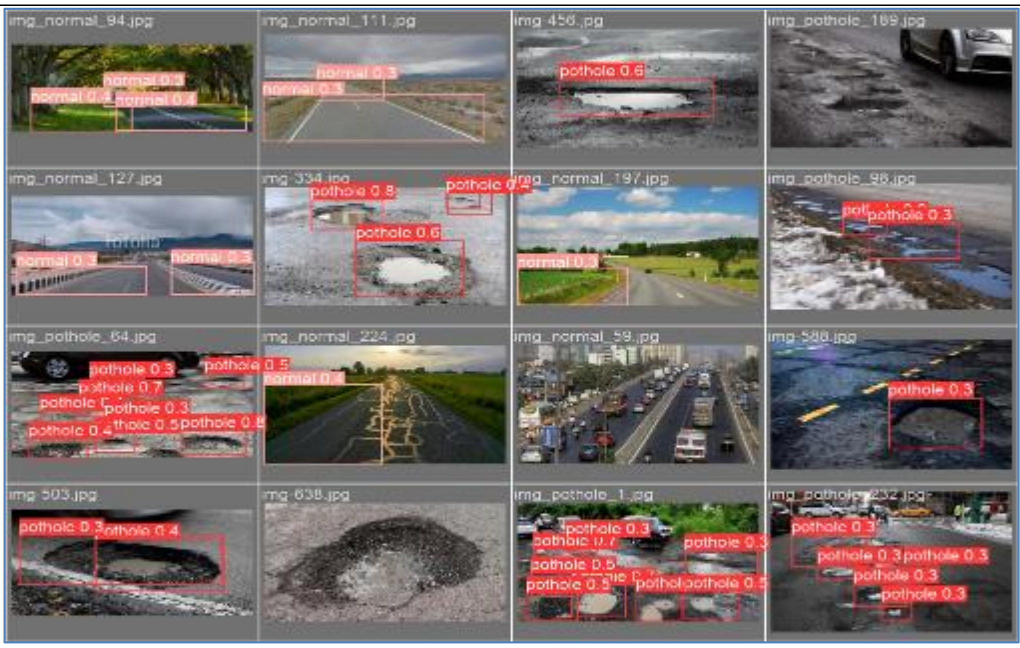
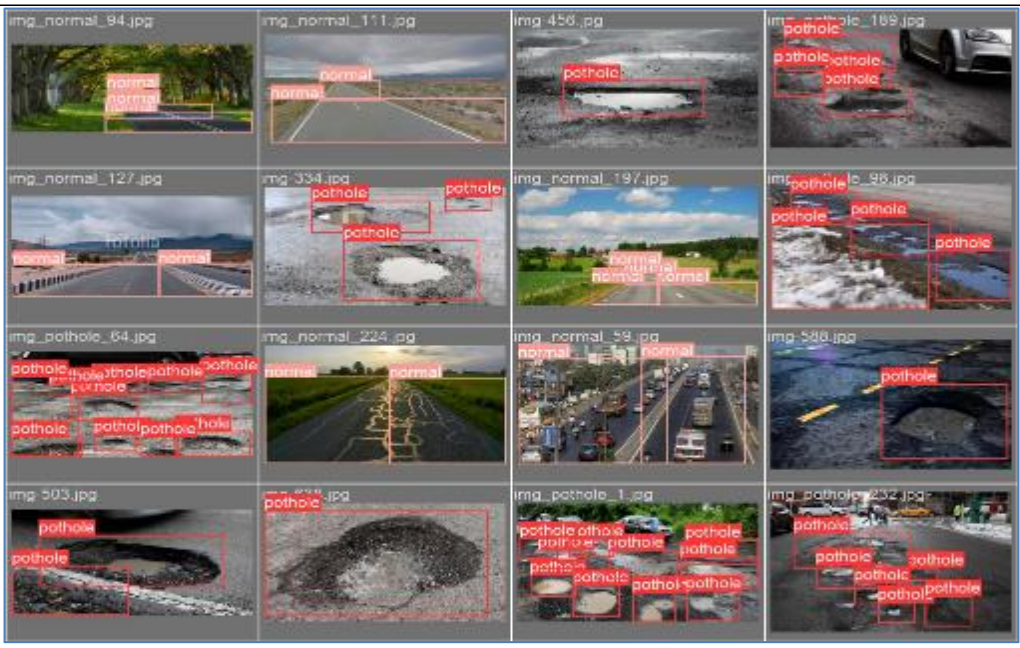
2.4. Programme Module

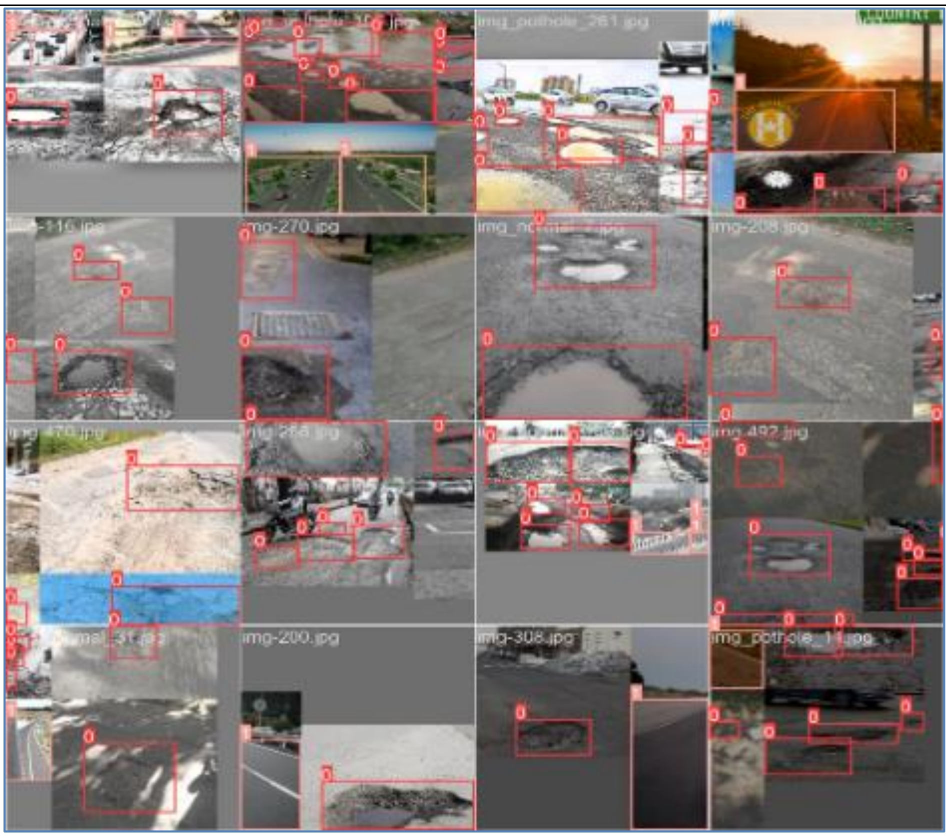
The dataset is labeled with Python programming language and Jupyter Notebook. 80% of the dataset was used for training and 20% for testing.

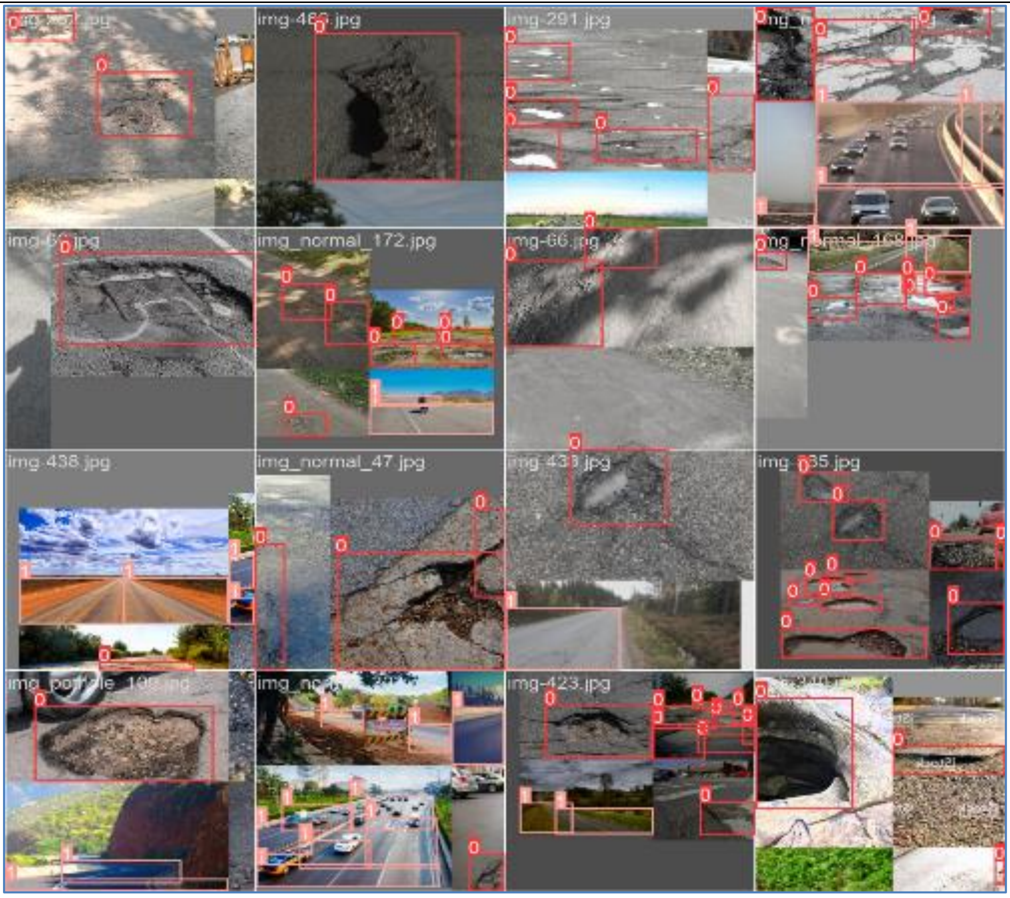
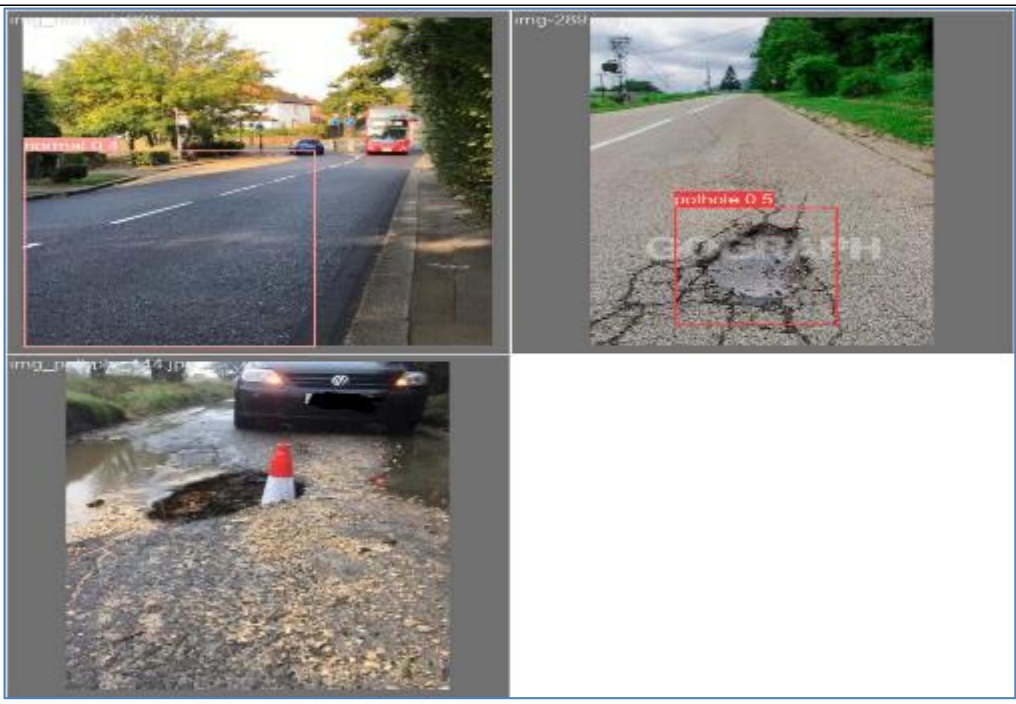
2.5. Datasets

Table 1 Sample Image dataset

S/N	Image Train	Class Nominal
1		['pothole',' normal']
2		['normal', 'pothole']

3		['normal', 'pothole']
4		['normal', 'pothole']

5		['0: pothole','1: normal']
6		['0: pothole','1: normal']

7		['0: pothole','1: normal']
8		['normal', 'pothole']

9	 .	['normal', 'pothole']

3. Results and Discussions

The experiment of the classification model was done with the training set, which was used to build the model. The test set is now used for detecting and predicting the result with the class labels as well as predicting a new class label with their respective class. The model results and analysis are presented hereunder.

3.1. Image Trained Image

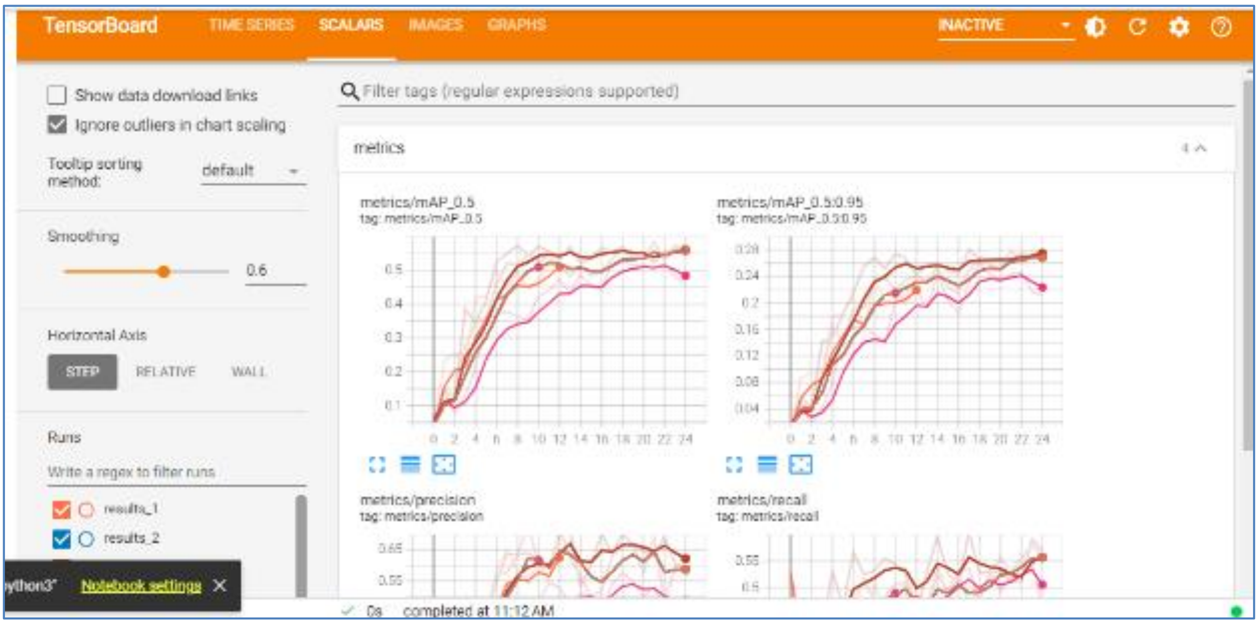


Figure 2a Tensorboard- Image trained Modules

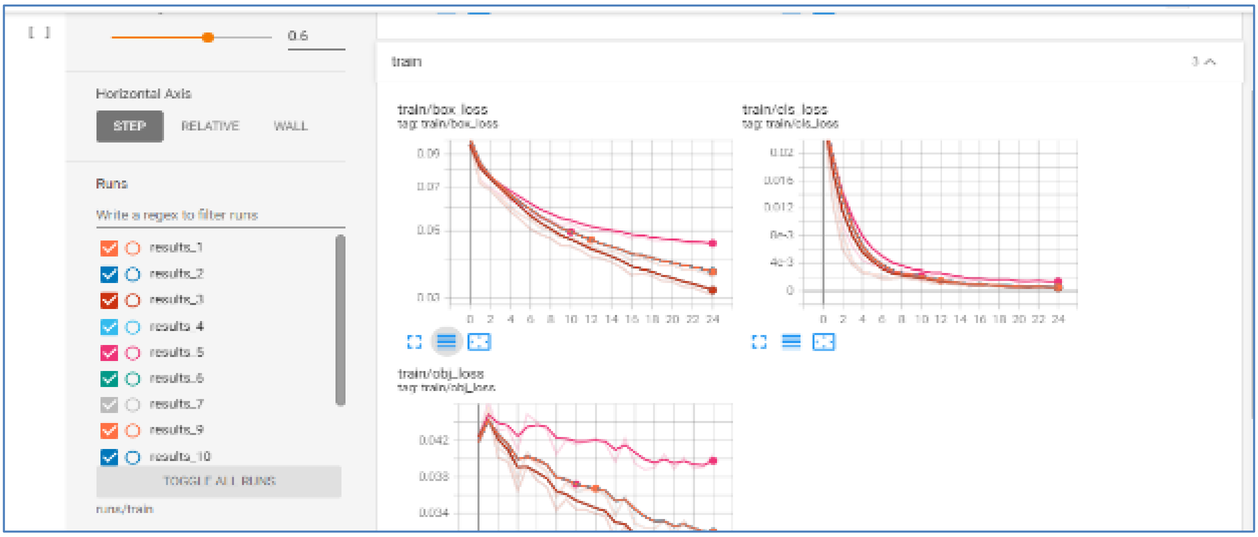


Figure 2b Tensorboard- Image trained Modules

Figure 2c Tensorboard- Image trained Modules

3.1.1. Model Training Sample Data set using YOLOv5

Figure 3 Sample YOLOv5 performance training process

Figure 3 show the level of model percentage accuracy assign to them, which presents the level of detection and prediction from the training image.

3.2. Performance Evaluation of the model

- Plotting labels to runs/train/results_11/labels.jpg...
- Image sizes 640 train, 640 val
- Using 2 dataloader workers
- Logging results to **runs/train/results_11**
- Starting training for 25 epochs...

Epoch	**GPU_mem**	**box_loss**	**obj_loss**	**cls_loss**	**Instances**	**Size**	
0/24	6.09G	0.09652	0.04241	0.02531	44	640: 100% 34/34 [00:34<00:00, 1.03s/it]	
	Class	Images	Instances	P	R	mAP50	**mAP50-95**: 100% 3/3 [00:02<00:00, 1.28it/s]
	All	67	157	0.173	0.0877	0.0513	0.0151
Epoch	**GPU_mem**	**box_loss**	**obj_loss**	**cls_loss**	**Instances**	**Size**	
1/24	7.99G	0.07188	0.04532	0.01243	53	640: 100% 34/34 [00:35<00:00, 1.03s/it]	
	Class	**Images**	**Instances**	**P**	**R**	**mAP50**	**mAP50-95**: 100% 3/3 [00:01<00:00, 1.80it/s]
	All	67	157	0.2	0.451	0.148	0.0474
Epoch	**GPU_mem**	**box_loss**	**obj_loss**	**cls_loss**	**Instances**	**Size**	
2/24	7.99G	0.06824	0.0401	0.005949	32	640: 100% 34/34 [00:33<00:00, 1.02it/s]	
	Class	**Images**	**Instances**	**P**	**R**	**mAP50**	**mAP50-95:** 100% 3/3 [00:01<00:00, 2.03it/s]
	All	67	157	0.148	0.317	0.126	0.0417
Epoch	**GPU_mem**	**box_loss**	**obj_loss**	**cls_loss**	**Instances**	**Size**	
3/24	7.99G	0.06281	0.03964	0.003697	27	640: 100% 34/34 [00:36<00:00, 1.08s/it]	
	Class	**Images**	**Instances**	**P**	**R**	**mAP50**	**mAP50-95**: 100% 3/3 [00:02<00:00, 1.49it/s]

	Class	**Images**	**Instances**	**P**	**R**	**mAP50**	**mAP50-95**: 100% 3/3 [00:02<00:00, 1.49it/s]
	All	67	157	0.433	0.424	0.386	0.141
Epoch	**GPU_mem**	**box_loss**	**obj_loss**	**cls_loss**	**Instances**	**Size**	
4/24	7.99G	0.05793	0.03653	0.002606	41	640: 100% 34/34 [00:34<00:00, 1.03s/it]	
	Class	**Images**	**Instances**	**P**	**R**	**mAP50**	**mAP50-95**: 100% 3/3 [00:01<00:00, 2.07it/s]
	All	67	157	0.355	0.53	0.344	0.149
Epoch	**GPU_mem**	**box_loss**	**obj_loss**	**cls_loss**	**Instances**	**Size**	
5/24	7.99G	0.05444	0.03919	0.002354	34	640: 100% 34/34 [00:34<00:00, 1.02s/it]	
	Class	**Images**	**Instances**	**P**	**R**	**mAP50**	**mAP50-95**: 100% 3/3 [00:01<00:00, 1.91it/s]
	All	67	157	0.421	0.464	0.411	0.183
Epoch	**GPU_mem**	**box_loss**	**obj_loss**	**cls_loss**	**Instances**	**Size**	
6/24	7.99G	0.05078	0.03765	0.001724	45	640: 100% 34/34 [00:31<00:00, 1.07it/s]	
	Class	**Images**	**Instances**	**P**	**R**	**mAP50**	**mAP50-95**: 100% 3/3 [00:01<00:00, 2.04it/s]
	All	67	157	0.548	0.566	0.525	0.217
Epoch	**GPU_mem**	**box_loss**	**obj_loss**	**cls_loss**	**Instances**	**Size**	
7/24	7.99G	0.04866	0.03694	0.001718	41	640: 100% 34/34 [00:32<00:00, 1.06it/s]	

	Class	**Images**	**Instances**	**P**	**R**	**mAP50**	**mAP50-95**: 100% 3/3 [00:02<00:00, 1.49it/s]
	All	67	157	0.433	0.424	0.386	0.141
Epoch	**GPU_mem**	**box_loss**	**obj_loss**	**cls_loss**	**Instances**	**Size**	
4/24	7.99G	0.05793	0.03653	0.002606	41	640: 100% 34/34 [00:34<00:00, 1.03s/it]	
	Class	**Images**	**Instances**	**P**	**R**	**mAP50**	**mAP50-95**: 100% 3/3 [00:01<00:00, 2.07it/s]
	All	67	157	0.355	0.53	0.344	0.149
Epoch	**GPU_mem**	**box_loss**	**obj_loss**	**cls_loss**	**Instances**	**Size**	
5/24	7.99G	0.05444	0.03919	0.002354	34	640: 100% 34/34 [00:34<00:00, 1.02s/it]	
	Class	**Images**	**Instances**	**P**	**R**	**mAP50**	**mAP50-95**: 100% 3/3 [00:01<00:00, 1.91it/s]
	All	67	157	0.421	0.464	0.411	0.183
Epoch	**GPU_mem**	**box_loss**	**obj_loss**	**cls_loss**	**Instances**	**Size**	
6/24	7.99G	0.05078	0.03765	0.001724	45	640: 100% 34/34 [00:31<00:00, 1.07it/s]	
	Class	**Images**	**Instances**	**P**	**R**	**mAP50**	**mAP50-95**: 100% 3/3 [00:01<00:00, 2.04it/s]
	All	67	157	0.548	0.566	0.525	0.217
Epoch	**GPU_mem**	**box_loss**	**obj_loss**	**cls_loss**	**Instances**	**Size**	
7/24	7.99G	0.04866	0.03694	0.001718	41	640: 100% 34/34 [00:32<00:00, 1.06it/s]	

Figure 4 A log output of the model

Figure 4 presents a log output YOLOV5 model that is being trained on pothole and normal road for object detection. Here's a breakdown of what each column represents:

- **Epoch:** This refers to which epoch (iteration) of the training process the model is currently on.
- **GPU_mem**: This indicates the amount of memory (in gigabytes) that the GPU is currently using to train the model.
- **box_loss, obj_loss, and cls_loss**: These three columns refer to the losses for the bounding boxes, objectness, and class predictions, respectively. These values are used to optimize the model during training.
- **Instances**: This represents the total number of instances (objects) detected in the training data.
- **Size:** This is the size of the images being used for training, in pixels.
- **Class, Images, Instances, P, R, mAP50, mAP50-95**: These columns are related to the evaluation of the model's performance on a validation set. P stands for precision, R for recall, and mAP for mean average precision. These are metrics used to evaluate the accuracy of object detection models. The numbers in the table represent the precision, recall, and mAP50 and mAP50-95 scores for all classes combined, based on the validation set.

Overall, this log output provides a summary of the training progress and performance of the object detection model.

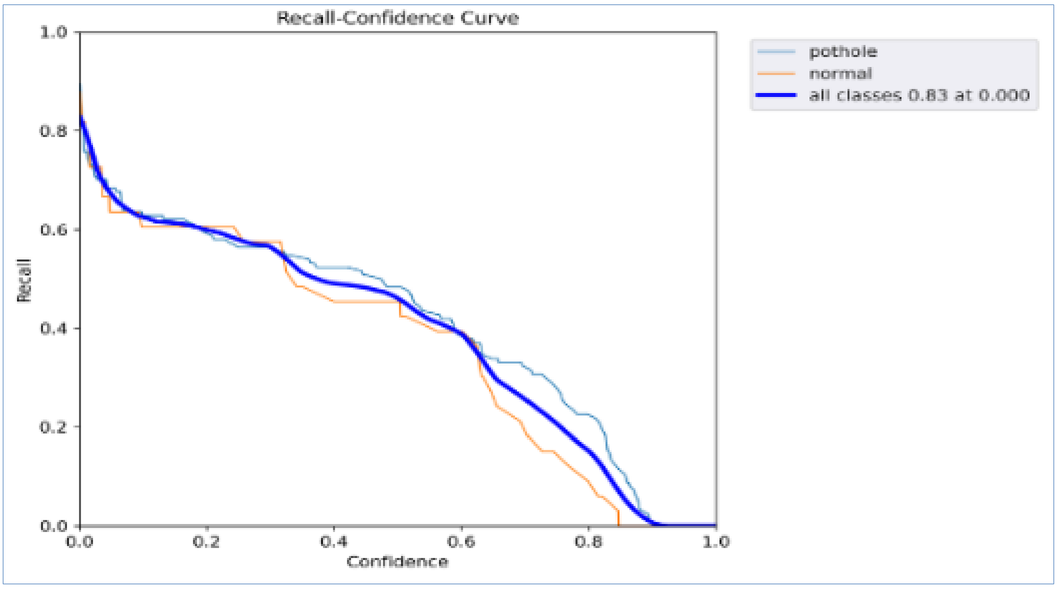


Figure 5 Model Performance for recall

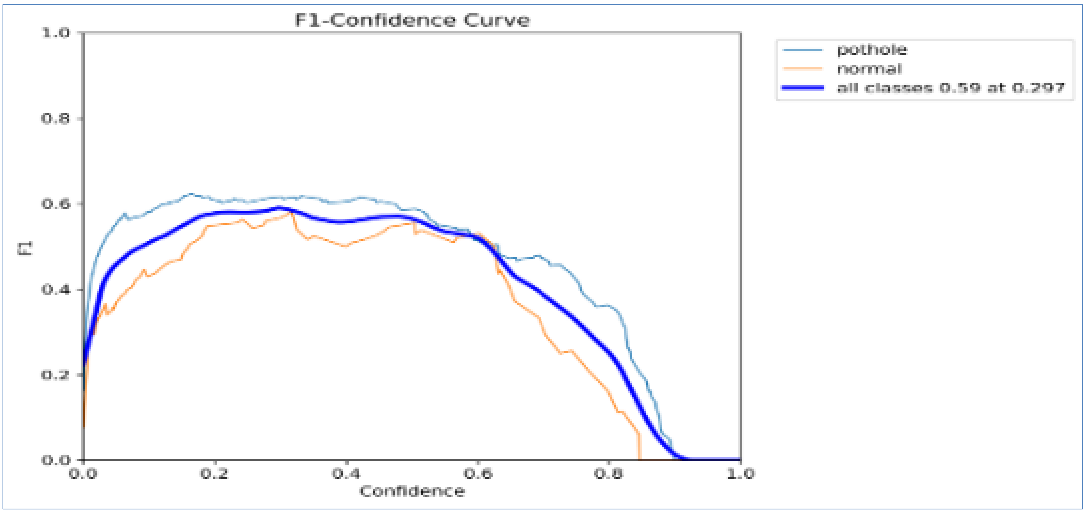


Figure 6 Model Performance for F1 score

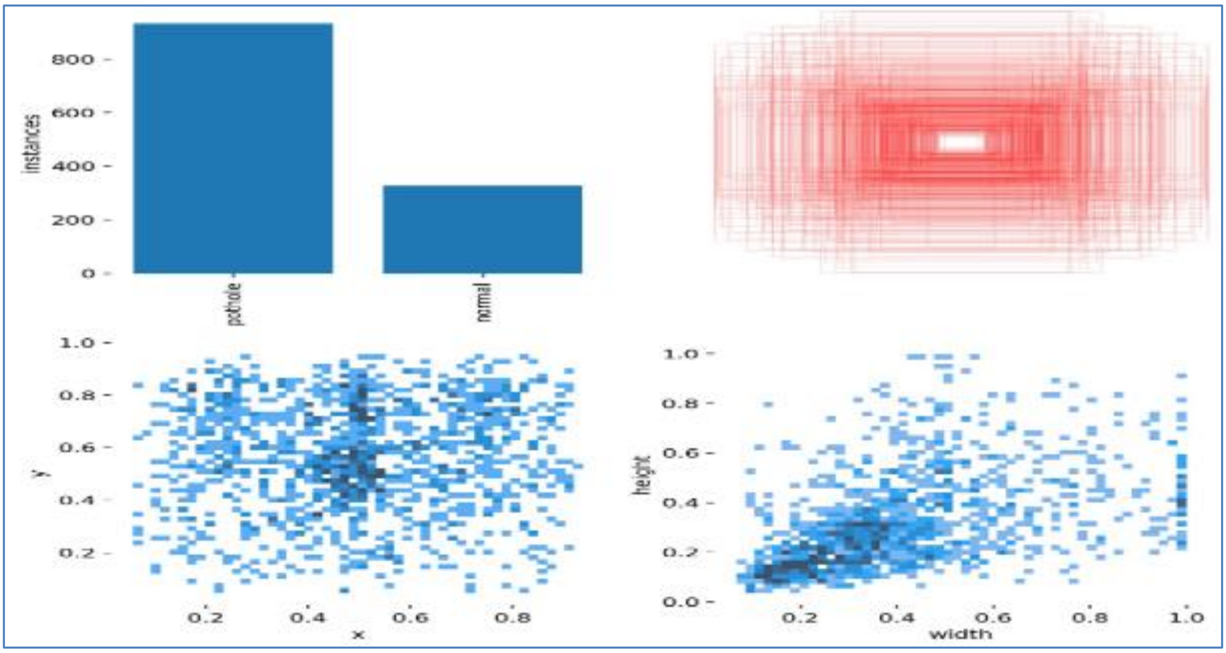


Figure 7 Model Performance for class Labels

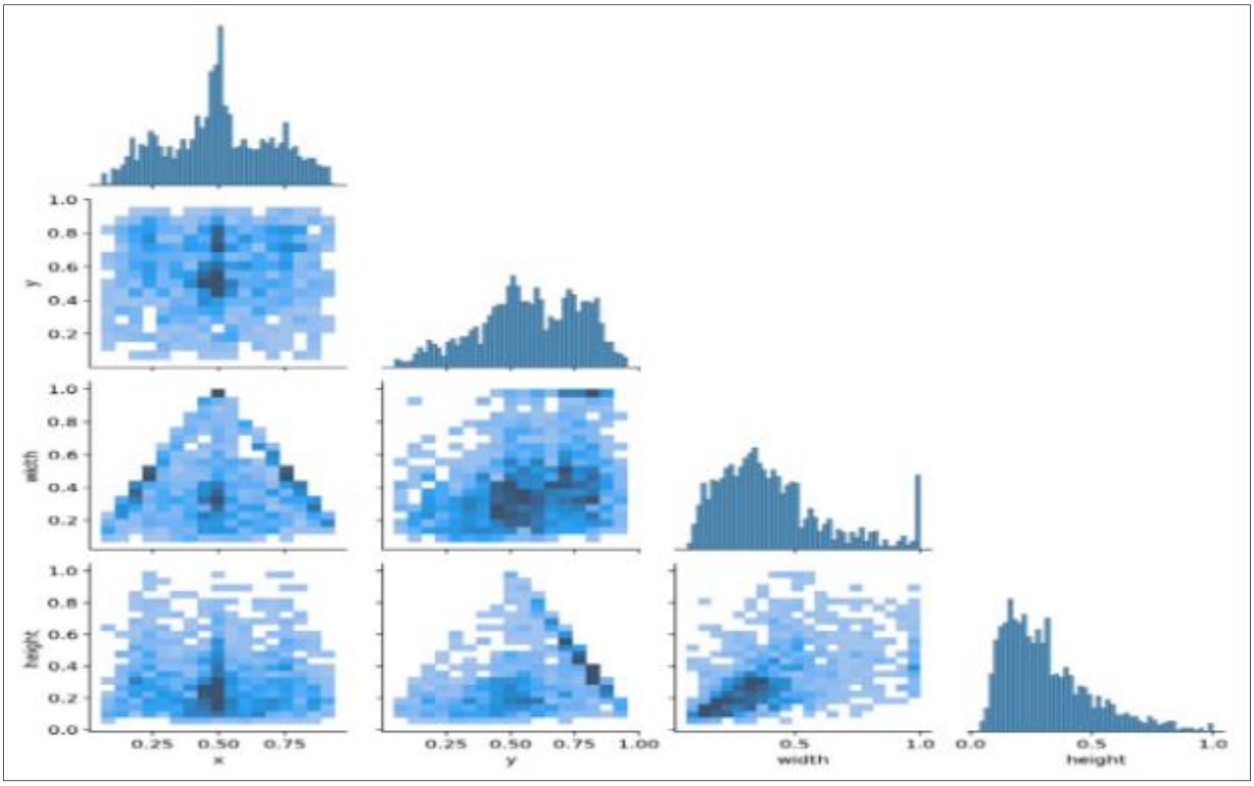


Figure 8 Model Performance for class Labels Corrolograms

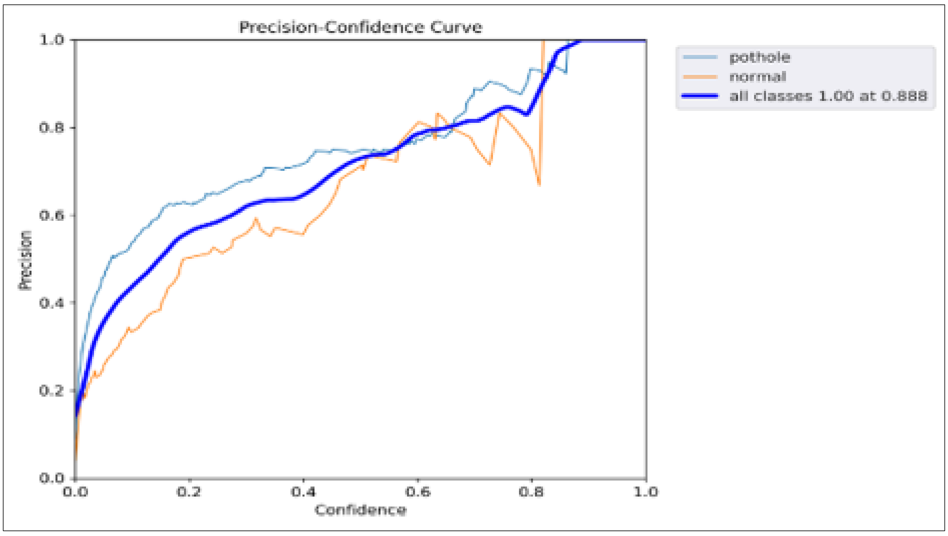


Figure 9 Model Performance for class Precision

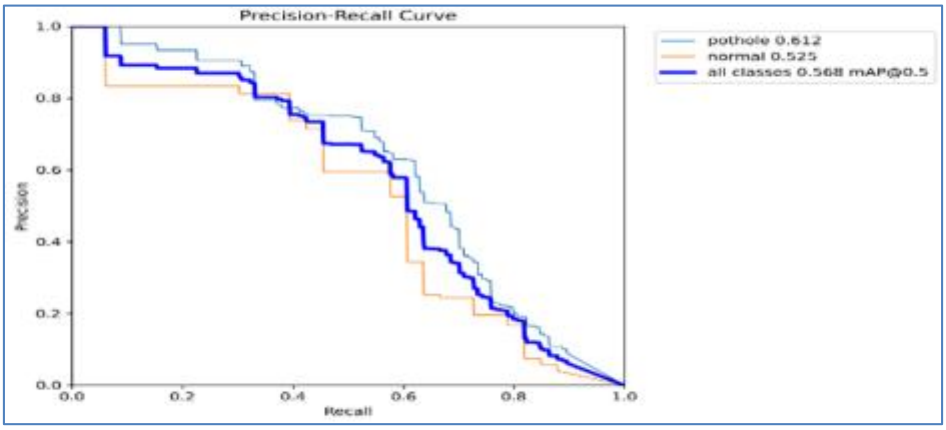


Figure 10 Model Performance for class Precision

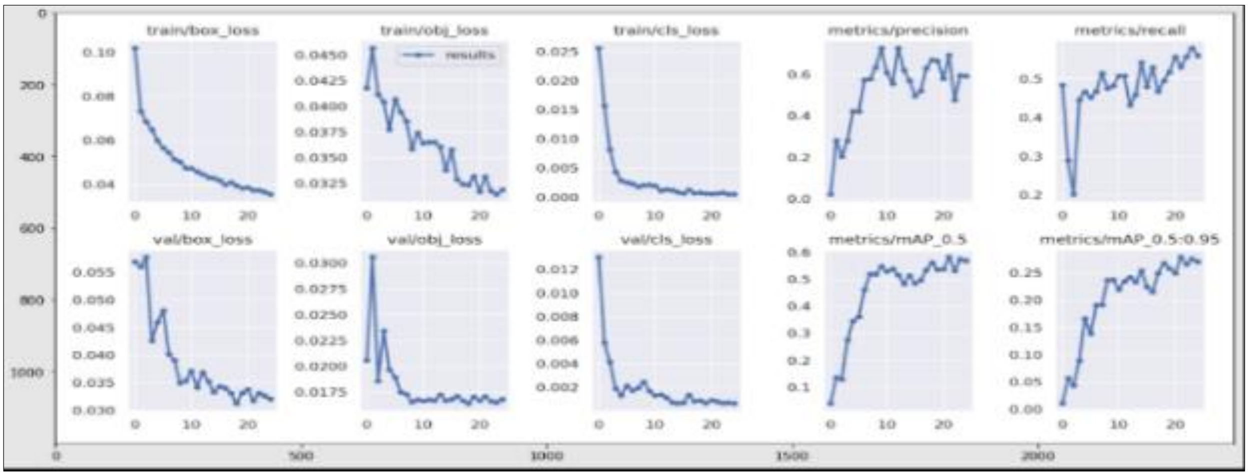


Figure 11 Model Performance for overall Results

3.3. Classification Report for model evaluation metrics with YOLOV5

Table 2 Details by categories of classification model

Class	Images	Instances	P	R	mAP50	mAP50-95: 100% 3/3 [00:01<00:00, 1.65it/s]
All	67	157	0.594	0.57	0.568	0.28
Normal	67	33	0.525	0.576	0.525	0.276
Pothole	67	124	0.663	0.565	0.612	0.287

3.4. Contengency table/confusion matrix

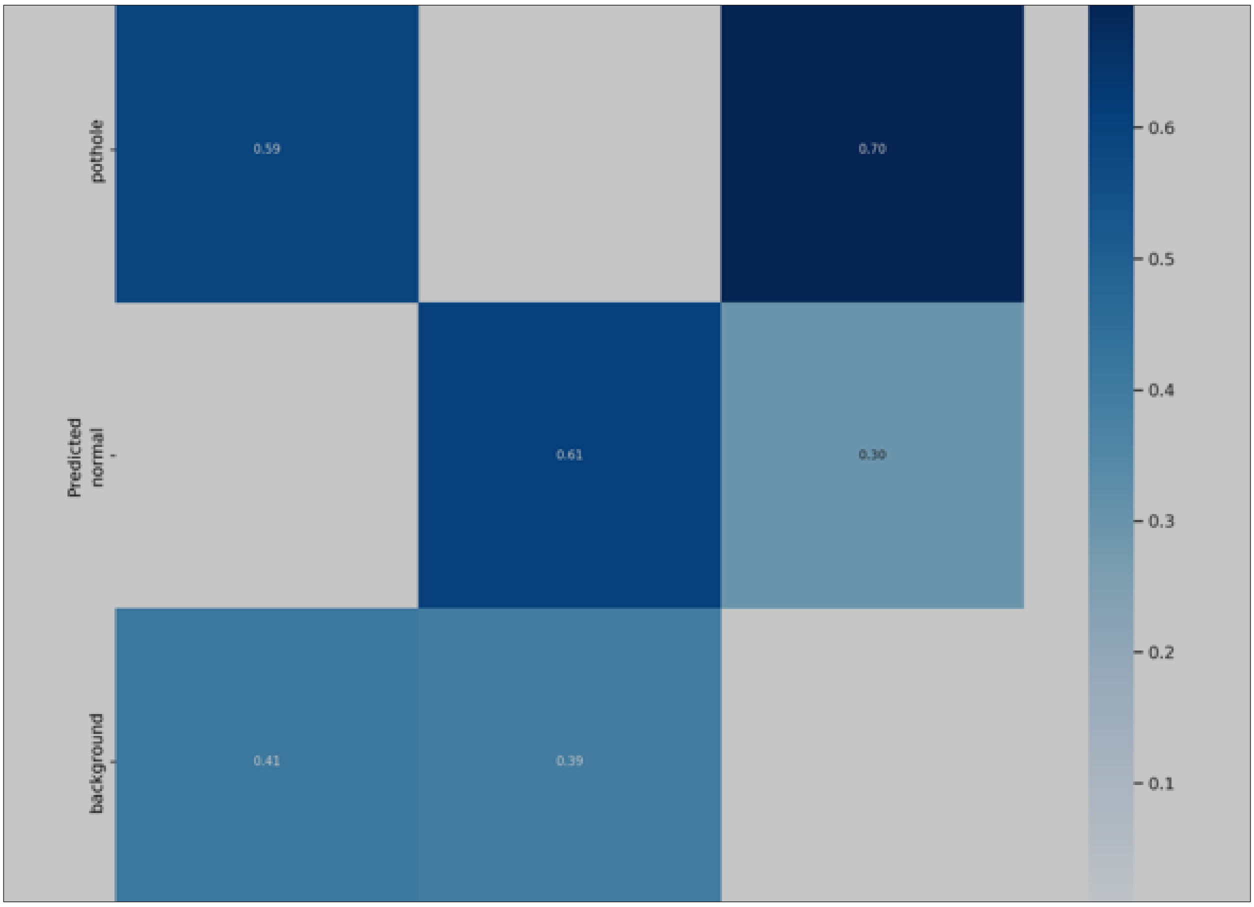


Figure 12 Contingency Table

4. Conclusion

This work presented a unique set of data with real time data collected from Nigeria and dataset from Kaggle which is now proven to be trainable with good predictions that can be adopted by researchers working on computer vision for pothole detection in Nigeria and indeed Africa. Among many other models that has been adopted for this related study, YOLOv5 has proven to be a good model with good prediction and perfect detection for the two classes of data trained in this research. This research will go a long way in deploying an embedded system or mobile applications for pothole detection and prediction in automobiles by the industries. Therefore there is need to train more road characteristics to help improve safe driving in the world. Our future research will be on deploying the results into mobile applications or embedded systems for intelligent transportation system.

Compliance with ethical standards

Acknowledgments

I would like to thank all the contributors who have participated in this research to make it a success. I have acknowledged the contributions made by my co-authors and the reviewers before the manuscript was sent for publication.

Disclosure of conflict of interest

All authors of this manuscript agreed and contributed significantly to the success of this research without conflict of interest.

References

[1] Approach Aminu Musa, Mohamed Hamada, and Mohammed Hassan3 (2022). A Theoretical Framework Towards Building a Lightweight Model for Pothole Detection using Knowledge Distillation. SHS Web of Conferences 139, 03002 (2022) ETLTC2022 https://doi.org/10.1051/shsconf/202213903002.

[2] Nduka, P. (2021). The Causes of Potholes On Roads, Their Effects And Control Methods, Engineering All: Engineering (Science & Technology), Technical Posts (Diy), And Business Investments In Nigeria, Aug. 20, 2021. https://www.engineeringall.com/the-causes-ofpotholes-on-roads-their-effects-and-control-methods/.

[3] W. health, Statistics, Injury Deaths Rise in Rank, 2008, https://www.who.int/violence_injury_prevention/key_facts/VIP_key_fact_3.pdf/.

[4] K. C. Wang (2004), Challenges and Feasibility for Comprehensive Automated Survey of Pavement conditions, in Proceedings of the Applications of Advanced Technologies in Transportation Engineering (2004), pp. 531–536, Beijing, China, May 2004.

[5] P. Harikrishnan and V. P. Gopi, Vehicle vibration signal processing for road surface monitoring, IEEE Sensors Journal, vol. 17, no. 16, pp. 5192–5197, 2017.

[6] A. Glowacz, (2021). Ventilation diagnosis of angle grinder using thermal imaging, Sensors, vol. 21, no. 8, p. 2853, 2021. Thermographic fault diagnosis of ventilation in bldc motors, Sensors, vol. 21, no. 21, p. 7245, 2021.

[7] L. Huidrom, L. K. Das, and S. Sud, (2013). Method for automated assessment of potholes, cracks and patches from road surface video clips, Procedia-Social and Behavioral Sciences, vol. 104, pp. 312–321, 2013.