iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.59712

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

Enhancing Object Detection Accuracy Through Custom Dataset Using Yolo

Sonali Ashok Khude[1], Nishita Vitthal Patil[2], Snehal Santosh Darade[3], Sneha Santosh Galande[4], Aishwarya Maruti Sawant[5] , Dr. Swati Pawar[6]
[1, 2, 3, 4, 5]*Student,* [6] *Mentor, Computer Science and Engineering, SVERI's College Of Engineering Pandharpur*

Abstract: ***Potholes pose significant risks to road safety and vehicle maintenance, leading to accidents and costly repairs. Traditional methods of pothole detection are often labour-intensive and time-consuming. In this study, we propose an innovative approach to pothole detection using YOLOv8, a state-of-the-art object detection algorithm. By harnessing the power of deep learning, our system can accurately identify and locate potholes in real-time video streams from traffic cameras and vehicles. We employ YOLOv8, an advanced variant of the You Only Look Once (YOLO) algorithm, known for its speed and accuracy in real-time object detection tasks. Leveraging a large annotated dataset of road images, we fine-tune the YOLOv8 model to specifically detect potholes. Our trained model is capable of identifying various pothole sizes and shapes, even in challenging lighting and weather conditions.***
The goal of this study is to apply different YOLO models for pothole detection. Three state-of-the-art object detection frameworks (i.e., YOLOv4,YOLOv4-tiny, and YOLOv5s) are experimented to measure their performance involved in real-time responsiveness and detection accuracy using the image set. The image set is identified by running the deep convolutional neural network (CNN) on several deep learning pothole detectors. After collecting a set of 600 images in 720×720 pixels resolution that captures various types of potholes on different road surface conditions, the set is divided into training, testing, and validation subset [1].

I. PROBLEM STATEMENT

To implement real – time object detection and recognition in an images captured by webcam and videos in dynamic environment using deep learning model and YOLO ." The primary goal is to detect and recognition Objects in Real-time. We require rich data, all things considered. We need to observe the different type of objects which are moving in respect to the camera. It will help us with perceiving and in recognizing different objects collaboration and interaction .

II. INTRODUCTION

The state of roads and infrastructure plays a pivotal role in ensuring safe and smooth transportation for citizens. One of the most common and hazardous road defects is the presence of potholes. Potholes not only cause discomfort to commuters but also pose serious risks to vehicles and public safety. Detecting potholes in a timely manner is crucial for efficient road maintenance and accident prevention.

Traditional methods of pothole detection involve manual inspections, which are often time-consuming, expensive, and can only cover limited stretches of roads. With the rapid advancements in computer vision and deep learning technologies, there has been a paradigm shift in the way we approach road defect detection. Object detection algorithms, particularly YOLO (You Only Look Once), have shown remarkable capabilities in real-time detection of various objects within images and video frames.

In this context, this study explores the application of YOLOv8, an advanced version of the YOLO algorithm, for the accurate and efficient detection of potholes. By harnessing the power of deep learning, we aim to develop an automated system capable of identifying potholes in real-time from images and video streams captured by surveillance cameras and vehicles.

The algorithm has been upgraded into its versions of YOLOv3 [3], YOLOv4 [4], and YOLOv5 [5] models is set as a measure of performance.

The integration of YOLOv8 into pothole detection not only promises higher accuracy but also the potential for real-time monitoring, enabling swift responses to identified road defects. This research delves into the technical aspects of YOLOv8, its customization for pothole detection, and the implications of such technology in revolutionizing road maintenance strategies. By leveraging artificial intelligence, we can pave the way for safer roads, reduced maintenance costs, and improved overall transportation experiences for communities.

First, the overview of existing methods for object detection was investigated as elaborated in Section 2. Second, the dataset was described in Section 3. Third, the approach for evaluating the performance of pothole detectors using data attributes of computer vision and those of CNN is given in Section 4. In addition, the validation experiment outputs are presented in Section 5. Final, the discussion on the experiments, research contributions and limitations, and future research recommendation are discussed in Section 6. The material in this paper is organized in the same order [1].

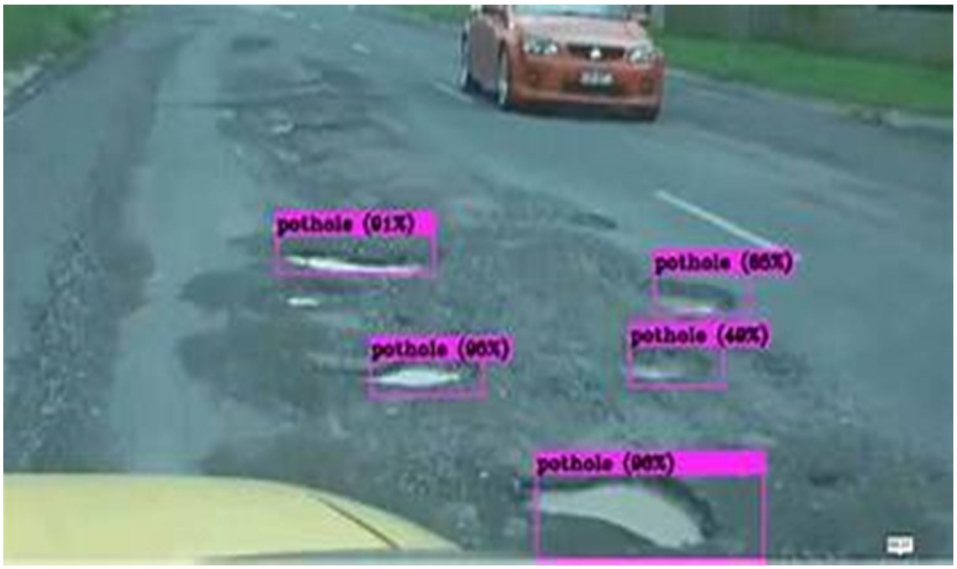


Fig : Pothole Detection

III. CURRENT STATE OF OBJECT DETECTION AND CLASSIfiCATION

A. Object Detection

Conventional Neural Networks (CNNs) play a fundamental role as the backbone architecture. CNNs are a class of deep neural networks designed to automatically and adaptively learn spatial hierarchies of features from input images. They are well-suited for image-related tasks, including object detection.

Object detection methods that use deep conventional neural networks (CNNs), which has a multi-stage or multi-layer architecture proposed by LeCun [6], do not require specific schemas to extract objects from images. The CNN, which includes sequential conventional, rectification, and pooling layers, may generate automatically features from input images. The parameters of each layer are automatically trained to detect an object (i.e., potholes, etc.). CNNs are typically utilized in object detection:

Feature Extraction, Shared Convolutional Layers, Localization and Classification, Backbone Networks, Transfer Learning.

In R-CNN working procedure of the selective search algorithm to select the most important regional proposals is to ensure that you generate multiple sub-segmentations on a particular image and select the candidate entries for your task. The greedy algorithm can then be made use of to combine the effective entries accordingly for a recurring process to combine the smaller segments into suitable larger segments.

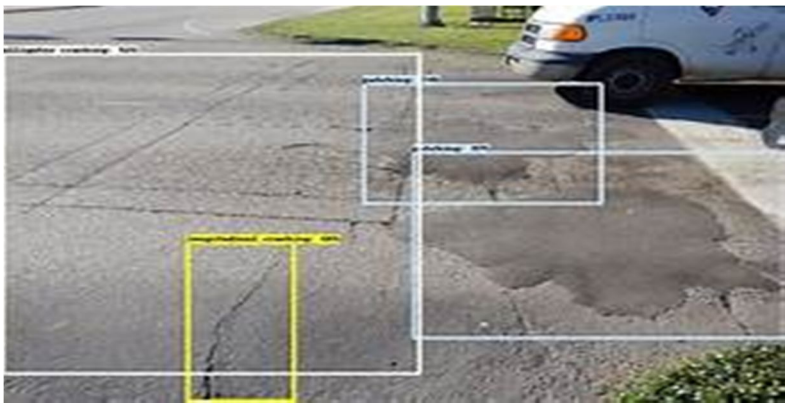


Fig 2 : Object Detection and Classification

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

In the fast R-CNN method, the entire image is passed through the pre-trained Convolutional Neural Network instead of considering all the sub-segments. The region of interest (RoI) pooling is a special method that takes two inputs of the pre-trained model and selective search algorithm to provide a fully connected layer with an output.

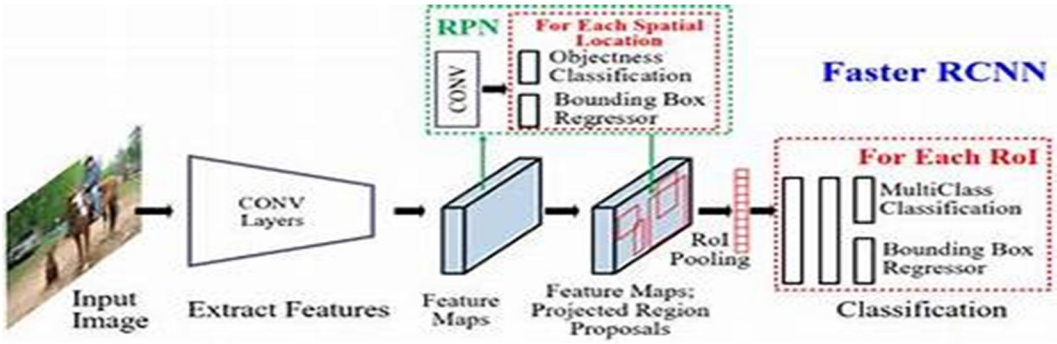


Fig 3 : Fast R-CNN

In this section, we will learn more about the Faster R-CNN network, which is an improvement on the fast R-CNN model[7-8].
Single Shot Detector (SSD)The single-shot detector for multi-box predictions is one of the fastest ways to achieve the real-time computation of object detection tasks. While the Faster R-CNN methodologies can achieve high accuracies of prediction, the overall process is quite time-consuming and it requires the real-time task to run at about 7 frames per second, which is far from desirable[9].

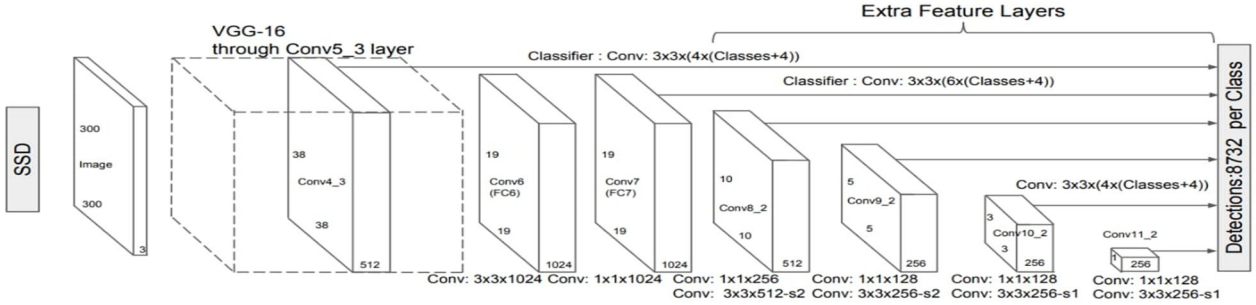


Fig 4 : Single Shot Detector

The YOLO architecture utilizes three primary terminologies to achieve its goal of object detection. Understanding these three techniques is quite significant to know why exactly this model performs so quickly and accurately in comparison to other object detection algorithms. The first concept in the YOLO model is residual blocks. In the first architectural design, they have used 7×7 residual blocks to create grids in the particular image. Each of these grids acts as central points and a particular prediction for each of these grids is made accordingly. In the second technique, each of the central points for a particular prediction is considered for the creation of the bounding boxes. While the classification tasks work well for each grid, it's more complex to segregate the bounding boxes for each of the predictions that are made. The third and final technique is the use of the intersection of union (IOU) to calculate the best bounding boxes for the particular object detection task[10].

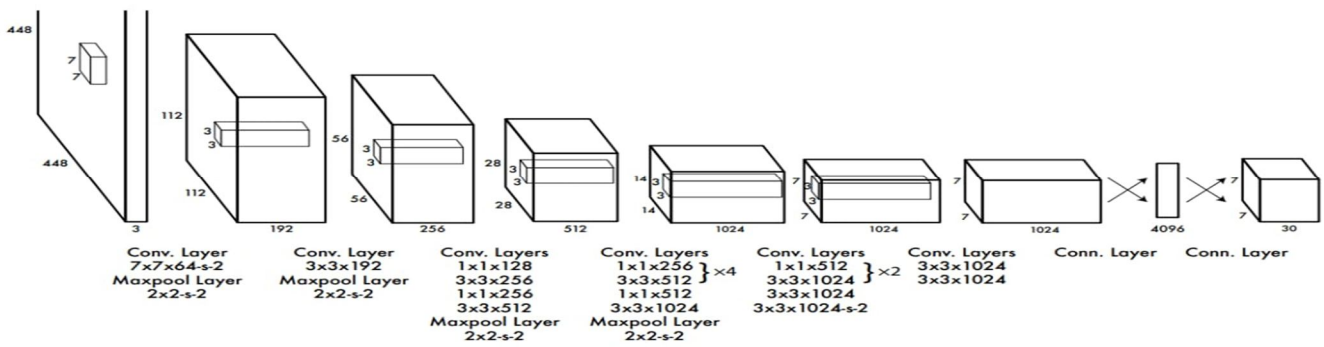


Fig 5 : YOLO layers

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

B. YOLO Architectures

YOLO, which stands for You Only Look Once, is a real-time object detection system. Its architecture is based on a deep convolutional neural network. Here's a simplified overview of the YOLO architecture:

YOLO takes an input image divided into a grid. CNN Backboneis a input image goes through a convolutional neural network (CNN) to extract features. YOLO often uses popular architectures like Darknet-53, which is a 53-layer version of the Darknet architecture.

In detection at multiple scales YOLO divides the image into a grid, typically, for example, a 19x19 grid. Each grid cell predicts bounding boxes and class probabilities. Importantly, YOLO can also operate at different scales. It constructs feature pyramids and applies detection at multiple scales to detect objects of various sizes.

In bounding box prediction for each grid cell, YOLO predicts multiple bounding boxes. Each bounding box is represented by 5 values: (x,y,w,h,confidence),where (x,y) is the center of the box, w and *h*

are the width and height of the box, and confidence confidence represents how confident the algorithm is about the presence of an object.

In class prediction YOLO also predicts class probabilities for each bounding box. It uses soft max activation to assign each bounding box to a specific class.

The output of YOLO is a set of bounding boxes, each associated with a class label and a confidence score. Post-processing techniques like non-maximum suppression (NMS) are often used to filter out redundant or weak predictions.

This architecture allows YOLO to detect objects in real-time with a single pass through the network, making it popular in applications like autonomous driving, video surveillance, and more. Different versions of YOLO, such as YOLOv1[11], YOLOv2 (or YOLOv2 Tiny)[12], YOLOv3[13], and YOLOv4[14],YOLOV8[13] have been developed with improvements in accuracy and speed.

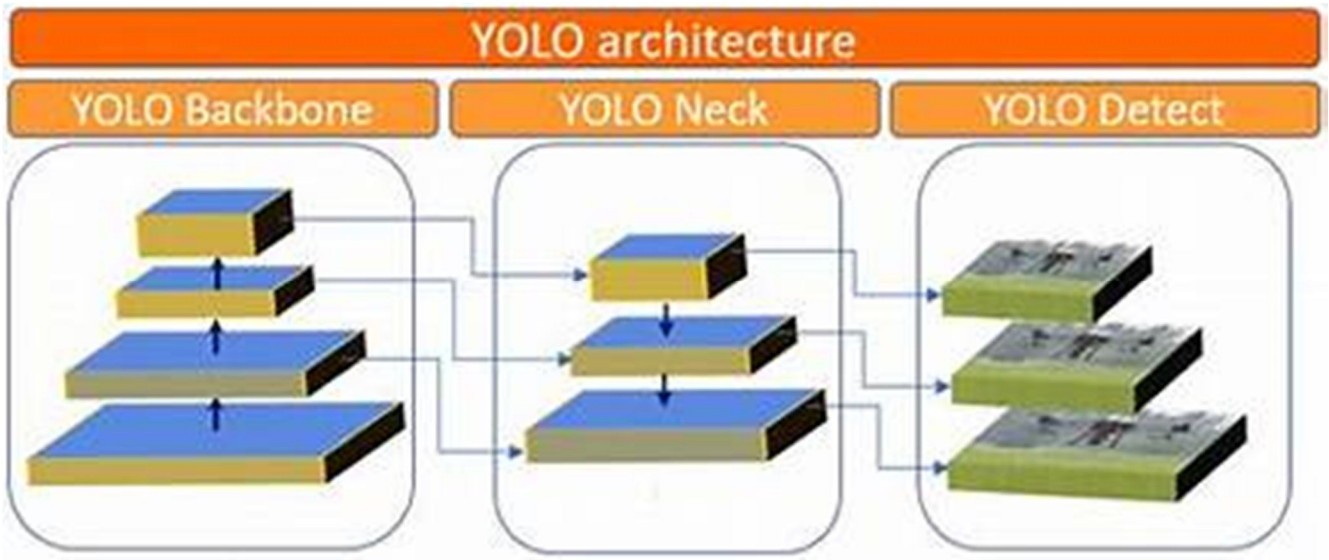


Fig 6 : YOLO Architecture

IV. DATASET

Roboflow, being a platform for managing and annotating datasets, would provide tools for organizing this data, annotating images, and potentially augmenting the dataset for training machine learning models.

For the most accurate and detailed information, I recommend visiting Roboflow's official website or contacting their support directly, as they might have updated datasets and specific details about the pothole detection dataset you're interested in.

A. A Dataset for Pothole Detection are Include

1) *Images or Videos:* The dataset would contain images or video frames where potholes are visible.
2) *Annotations:* For each image or video frame, there would be annotations specifying the coordinates of the potholes. Bounding boxes around potholes are a common annotation format for object detection tasks.
3) *Metadata:* Additional information such as pothole depth, location, image source, weather conditions, or any other relevant data could be included.

The dataset consists of the images and their corresponding labels. The dataset was divided into training, validation, and testing subsets each of which ratio is 70%, 20%, and 10%, respectively. Typical pothole images in the testing subset are shown in Figure 7.

Figure 7 : Typical pothole images on road surfaces of the testing subset.

V. METHODOLOGY

A. *Problem Definition*

We want a pothole object detection. For each image or video frame, there would be annotations specifying the coordinates of the potholes. Bounding boxes around potholes are a common annotation format for object detection tasks. Additional information such as pothole depth, location, image source, weather conditions, or any other relevant data could be included.

B. *Data Collection and Preparation:*

For the most accurate and detailed information, I recommend visiting Roboflow's official website or contacting their support directly, as they might have updated datasets and specific details about the pothole detection dataset you're interested in.
Roboflow, being a platform for managing and annotating datasets, would provide tools for organizing this data, annotating images, and potentially augmenting the dataset for training machine learning models.

C. *Choose a Model*

Select a pre-trained CNN architecture suitable for object detection tasks. Popular choices include Faster R-CNN, YOLO (You Only Look Once), and SSD (Single Shot MultiBox Detector). Alternatively, you can design a custom architecture based on your specific requirements.

D. *Transfer Learning*

Fine-tune the chosen pre-trained model on your dataset. Transfer learning from a model pre-trained on a large dataset helps your model converge faster and perform better.

E. *Loss Function*

The sum squared error is used throughout the loss function which is presented using Eq. 6. As we can notice, there are total five terms in the underlying equation wherein notations can be defined as follows:
(xi, yi) is the ground truth center of the bounding box,
(xi^,yi^) is the predicted center of the bounding box,
(wi, hi) is the wid is the width and height respectively of the predicted bounding box.
th and height respectively of the ground truth bounding box,

(wi^,hi^)The first two terms are related to errors correspond to the differences in positioning of predicted bounding boxes and the ground truth bounding boxes. The deviations have a greater impact on IoU in case of smaller bounding boxes as compared to larger bounding boxes. To address this problem, square root of the width and height of the bounding box is considered instead of width and height directly in the loss function. Third term refers to the prediction difference in the confidence score when the object is present in the corresponding bounding box. In first three terms, 1 objij1is an indicator function, which represents *ith* grid responsible for predicting the *jth* bounding box. It will be 1 if the cell contains the object, 0 otherwise. It may happen that ground truth may contain the object in a particular grid cell but the model wrongly predicts there's no object. Apart from considering the loss when the grid cell contains the object, they also aimed to reduce the loss when there's no object in the grid cell. It may happen that ground truth may not contain the object in a particular grid cell but the model wrongly predicts there's an object. Similarly, 1noobjij1is an indicator function, which represents *ith* grid responsible for predicting the *jth* bounding box. It will be 1 if the cell does not contain the object, 0 otherwise. The last term is classification loss, aimed to minimize the mis classification error. Herein, 1obji1 is an indicator function, which refers to a grid cell containing an object, it will be 1 if the grid cell contains an object and 0 otherwise. The *λcoord* and *λnoobj* are the hyper parameters that basically used to avoid divergence of gradients. All the grid cells may not contain the object, the confidence score and thereafter gradients will tend to zero in this case. So, in order to overcome this problem, they tried to maximize the loss of bounding box coordinates when the grid cell contains the object by multiplying the hyper parameter *λcoord* to the first and second terms and minimize the loss when there is no object in the grid cell by multiplying the hyper parameter *λnoobj* to the fourth term. In general, high value is assigned to *λcoord* and low value to *λnoobj* [15].

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \qquad (3)$$

F. Training

Train the model on the training dataset. Monitor its performance on the validation set to prevent overfitting. Adjust hyperparameters like learning rate, batch size, and number of epochs as needed.

G. Evaluation

Evaluate the trained model on the test dataset using metrics like mean Average Precision (mAP) to measure its accuracy. Adjust the model or the training process if the performance is not satisfactory.

H. Post-Processing

Apply non-maximum suppression (NMS) to eliminate duplicate or low-confidence detections. This step ensures that only the most confident predictions are kept.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

VI. RESEARCH GAP

As of our last knowledge update in September 2021, YOLOv8 was not a recognized version of the YOLO (You Only Look Once) series of object detection models. YOLO versions up to YOLOv4 were well-documented and widely discussed in the research community.

If YOLOv8 has emerged since then, I recommend looking for recent literature and official sources to gather information specific to that version.

However, if you are referring to training a YOLO-based object detection model (such as YOLOv4) on a custom dataset, there could be research gaps and challenges that still exist. Some potential research gaps and challenges for training YOLO-based object detection models on custom datasets include:

A. Architecture and Hyper-parameter Tuning

While YOLO provides a solid foundation, finding the optimal architecture and hyper-parameters for your specific dataset and application is a research area. Exploring different backbone architectures, anchor box configurations, and other hyper-parameters could lead to improved performance[16].

B. Domain Adaptation and Generalization

Models trained on one dataset might struggle to generalize well to new and unseen datasets due to domain shifts. Developing techniques for domain adaptation and improving the generalization capabilities of YOLO-based models is a research gap[17].

C. Handling Imbalanced Data

In real-world datasets, object class distribution might be imbalanced, leading to biased models. Research into effective techniques for handling imbalanced data during training could improve the model's ability to detect rare objects.

D. Data Augmentation Strategies

Data augmentation is crucial for training robust models. Research could focus on identifying novel data augmentation techniques that enhance the model's performance while addressing challenges specific to object detection.

E. Small Object Detection and Occlusion Handling

YOLO-based models, like any object detectors, might struggle with detecting small objects accurately or handling occlusions. Developing strategies to improve detection in these challenging scenarios remains a research area.

F. Transfer Learning Efficiency

Transferring knowledge from a pre-trained model to a custom dataset is essential for efficient training. Research could focus on techniques that accelerate this transfer process while maintaining or improving performance.

G. Multi-Object Tracking Integration

If working with video data, integrating multi-object tracking capabilities with YOLO-based object detection can be valuable. Research could explore ways to seamlessly combine these two tasks for enhanced scene understanding.

H. Resource-Efficient Inference

Developing methods to optimize YOLO-based models for inference on resource-constrained devices while maintaining accuracy is a research area of interest, especially for applications like edge computing.

VII. OBJECT DETECTION LIBRARIES

A. ImageAI

The ImageAI library aims to provide developers with a multitude of computer vision algorithms and deep learning methodologies to complete tasks related to object detection and image processing. The primary objective of the ImageAI library is to provide an effective approach to coding object detection projects with a few lines of code[11].

B. GluonCV

The GluonCV is one of the best library frameworks with most of the state-of-the-art implementations for deep learning algorithms for various computer vision applications. The primary objective of this library is to help the enthusiasts of this field to achieve productive results in a shorter time period. It has some of the best features with a large set of training datasets, implementation techniques, and carefully designed APIs[17].

C. Detectron2

The Detectron2 framework developed by Facebook's AI research (FAIR) team is considered to be a next-generation library that supports most of the state-of-the-art detection techniques, object detection methods, and segmentation algorithms. The Detectron2 library is a PyTorch-based object detection framework[17].

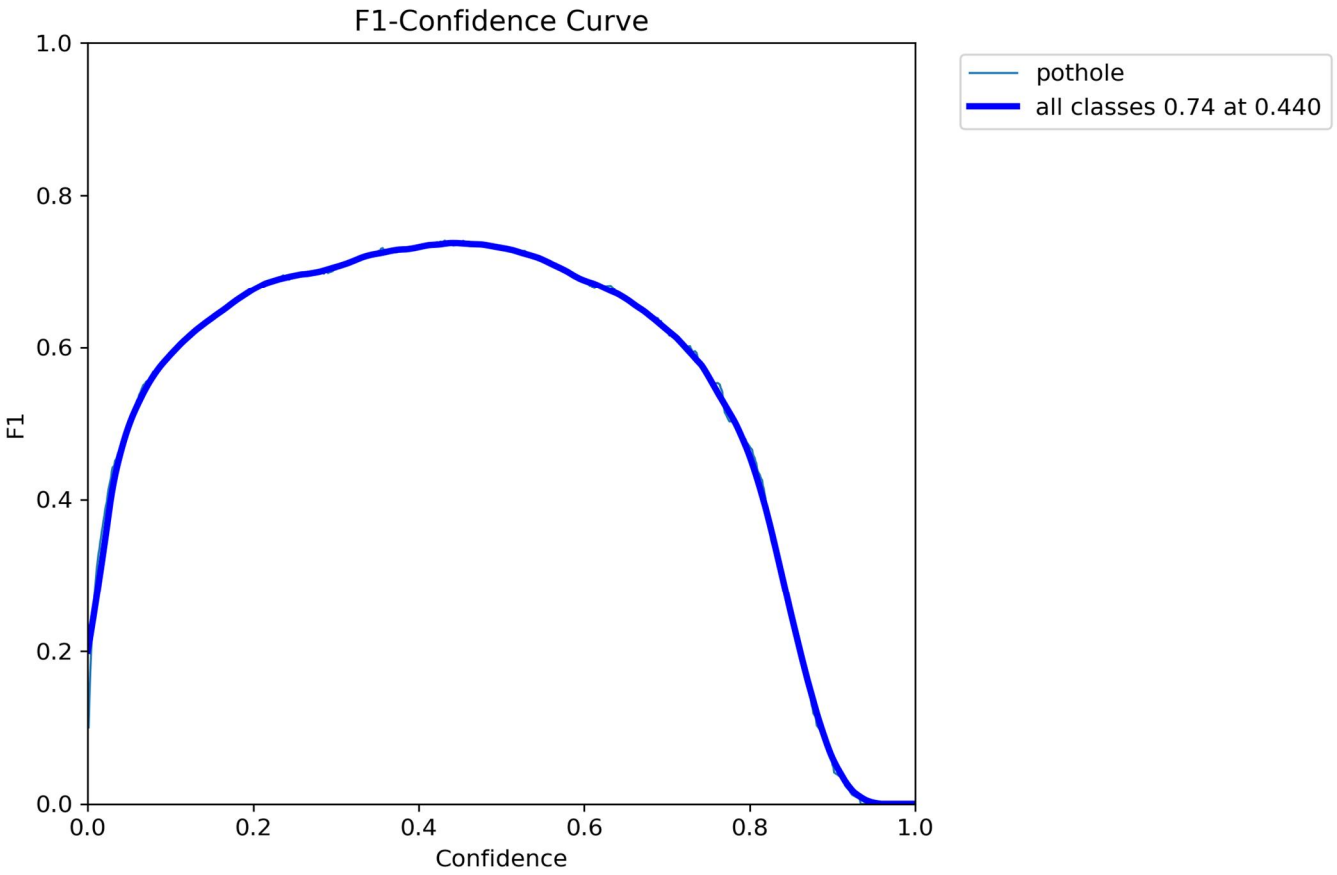
D. YOLOv3_TensorFlow

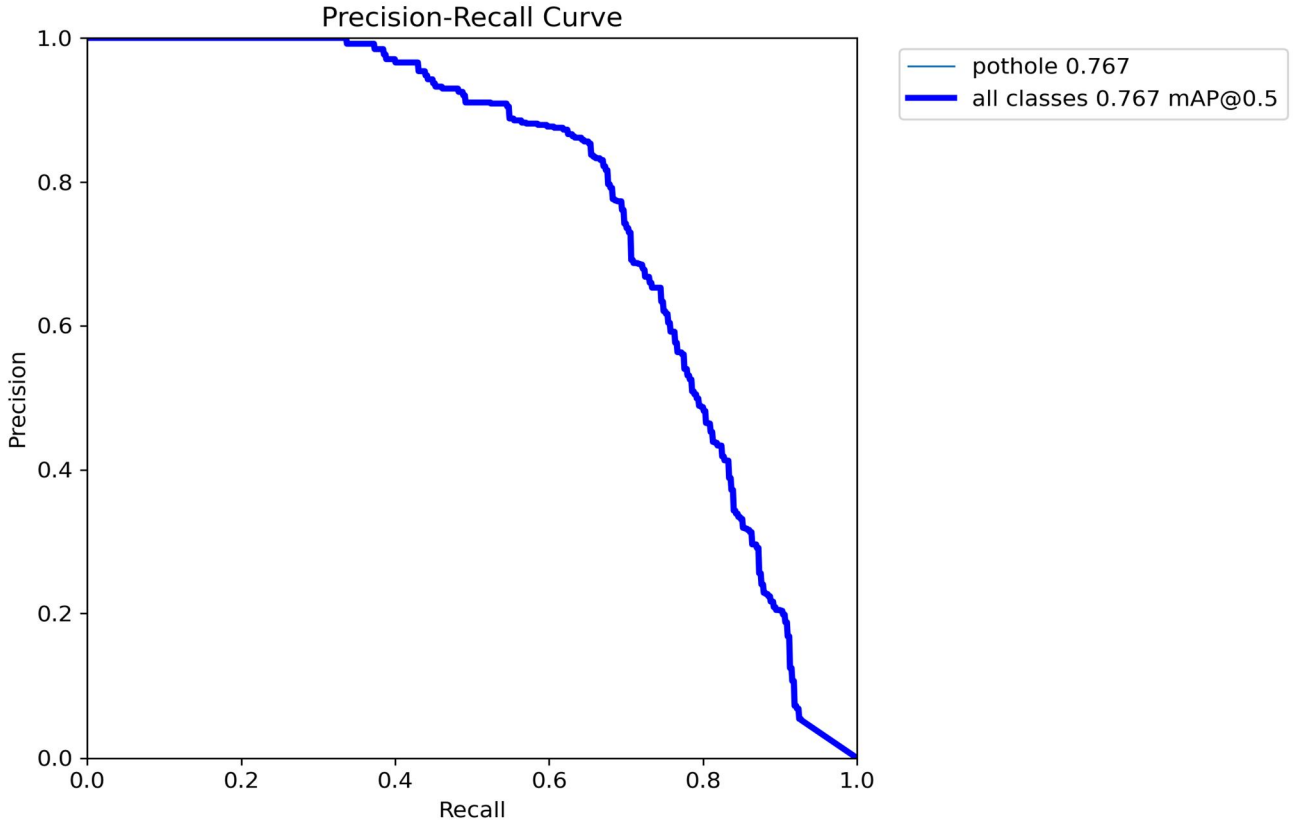
The YOLO v3 model is one of the successful implementations of the YOLO series, which was released in 2018. The third version of YOLO improves on the previous models. The performance of this model is better than its predecessors in terms of both speed and accuracy[17].

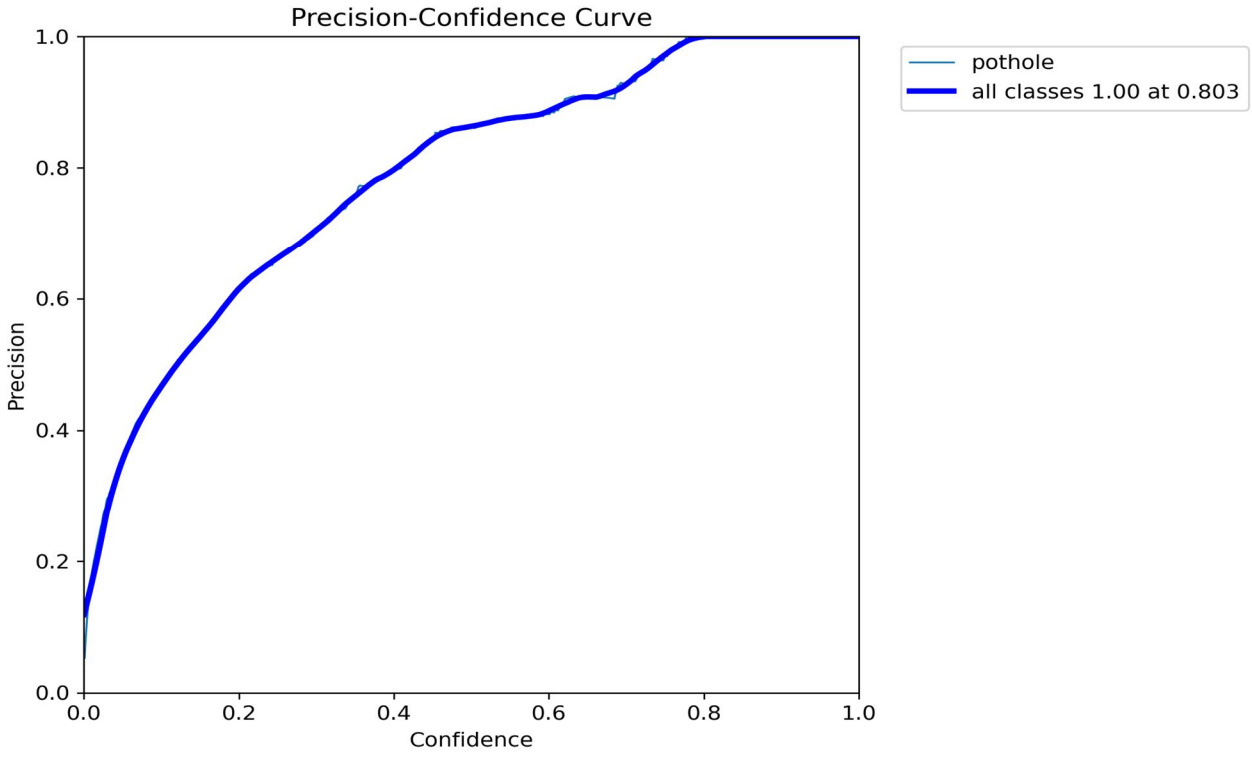
E. Darkflow

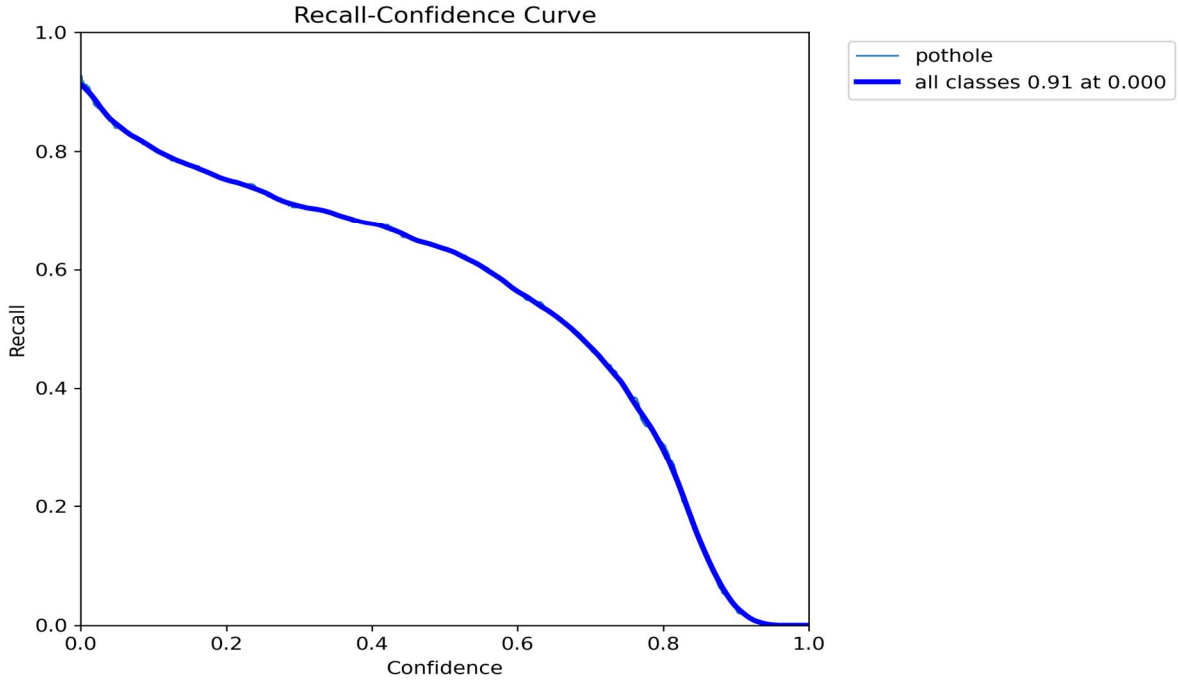
Darkflow is inspired by the darknet framework and is basically a translation to suit the Python programming language and TensorFlow for making it accessible to a wider range of audiences. Darknet is an early implementation of an object detection library with C and CUDA [17].

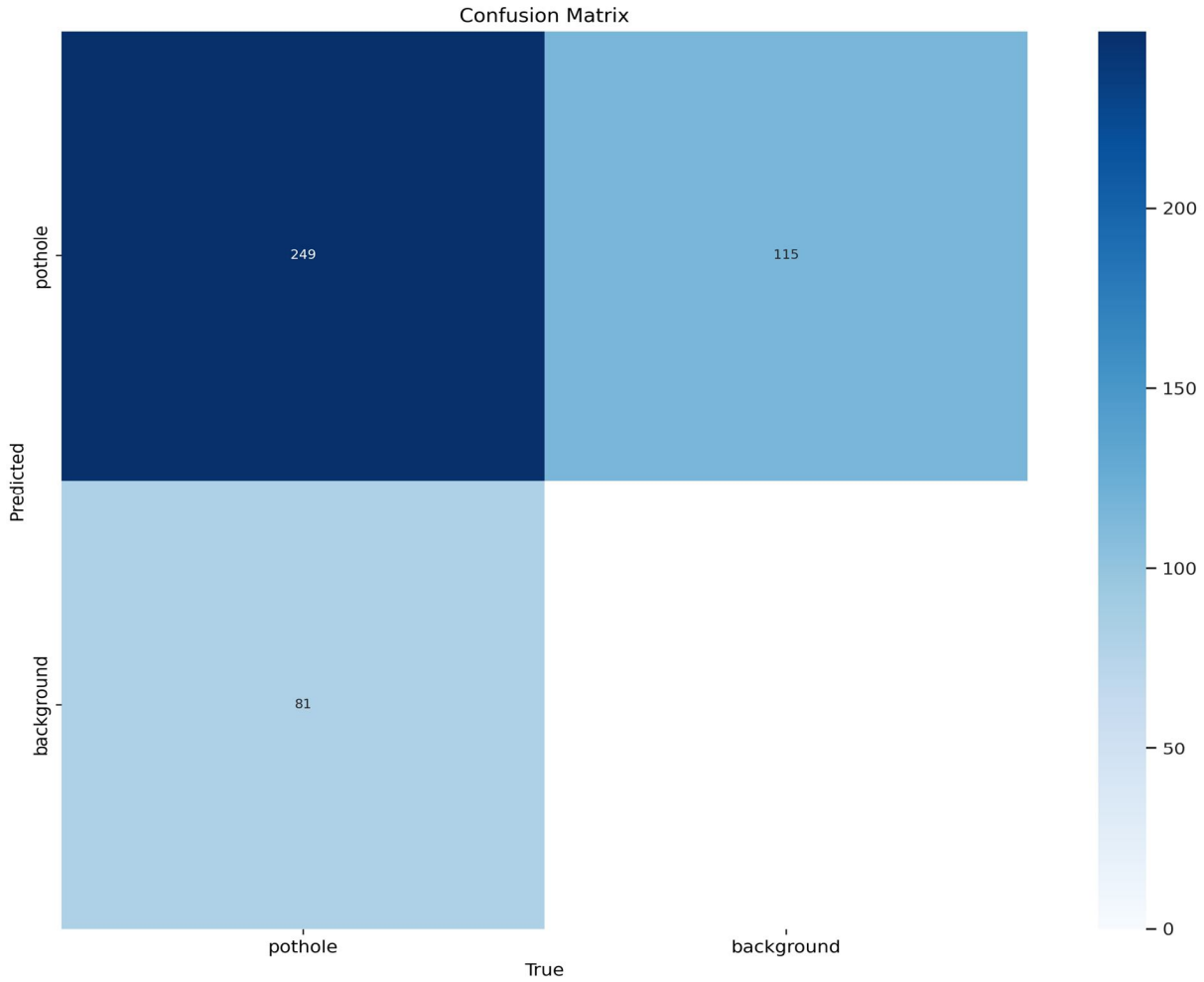
VIII. RESULT



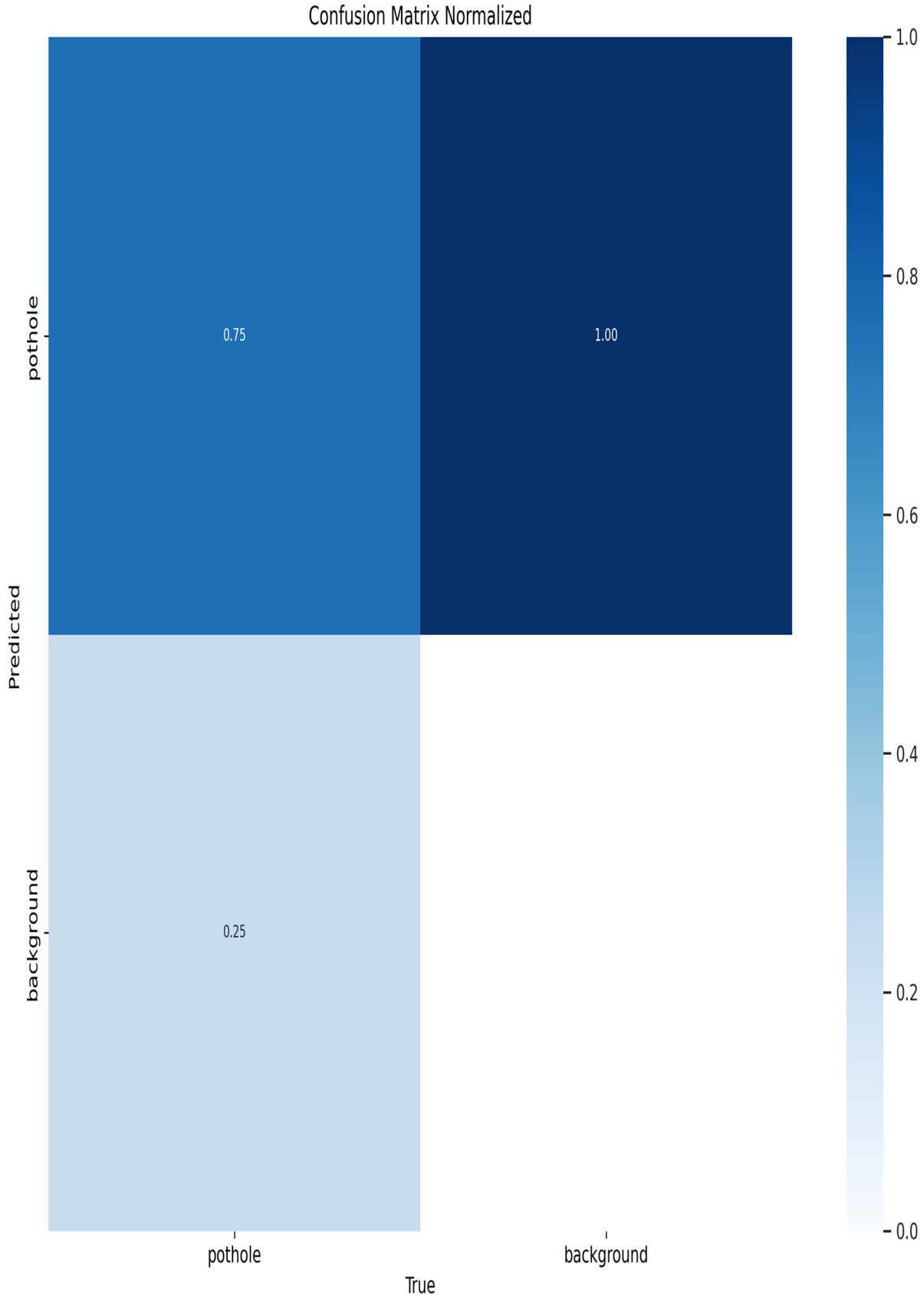

Precision-Recall Curve
pothole 0.767
all classes 0.767 mAP@0.5
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

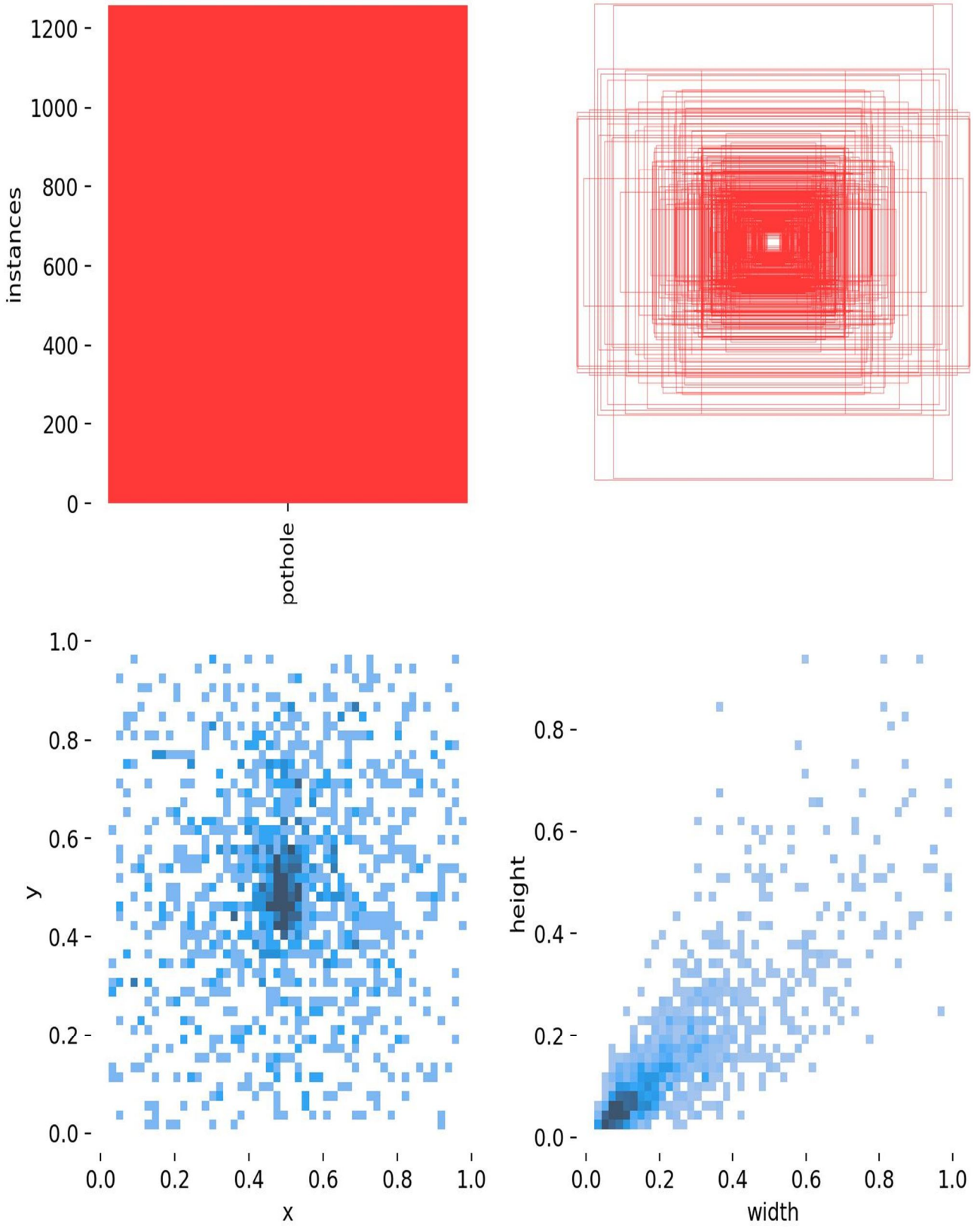

Precision-Confidence Curve
pothole
all classes 1.00 at 0.803
Precision
Confidence
0.0
0.2
0.4
0.6
0.8
1.0


Recall-Confidence Curve
pothole
all classes 0.91 at 0.000
Recall
Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Confusion Matrix
pothole
background
Predicted
249
115
81
pothole
background
True
200
150
100
50
0

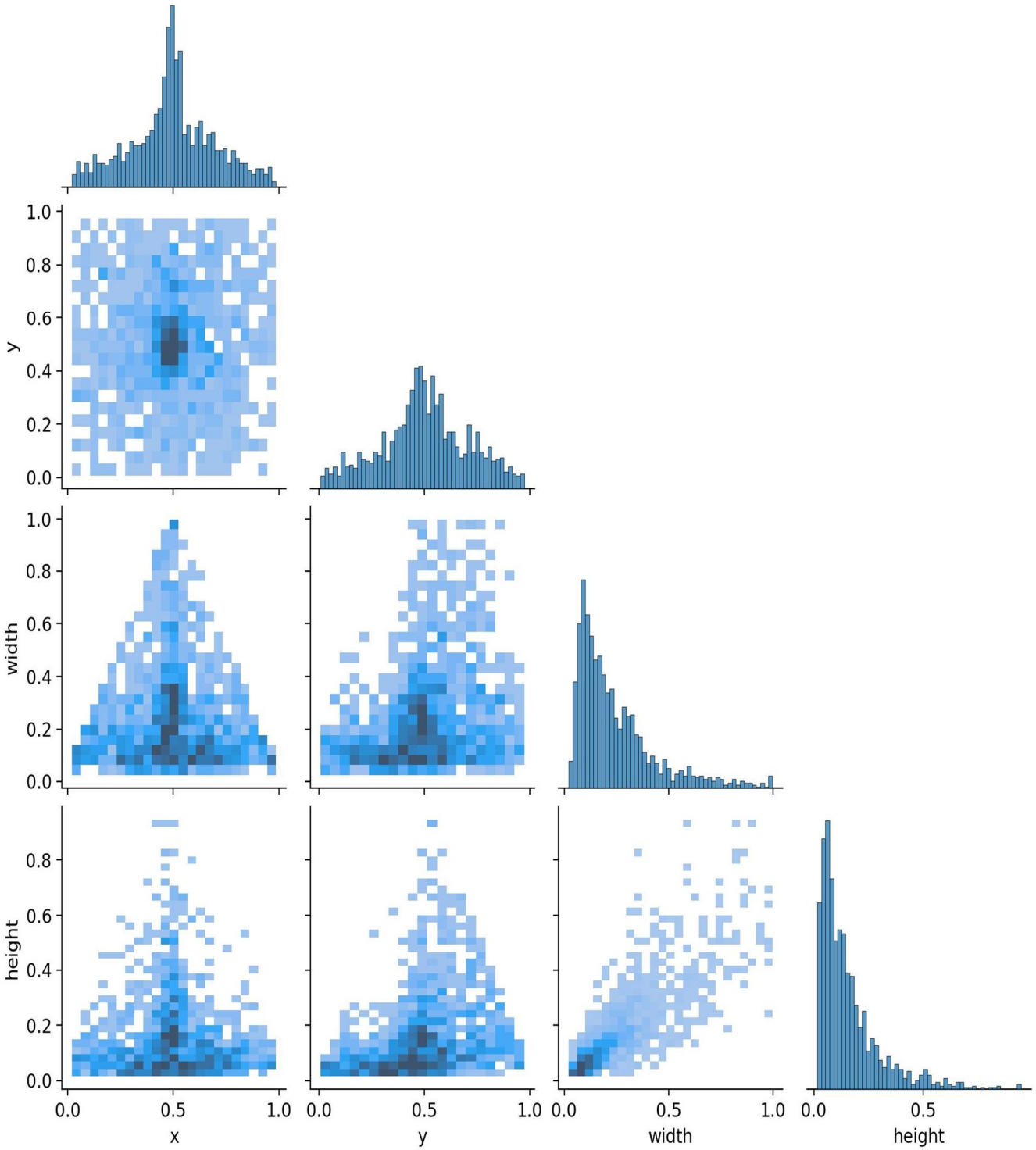
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com



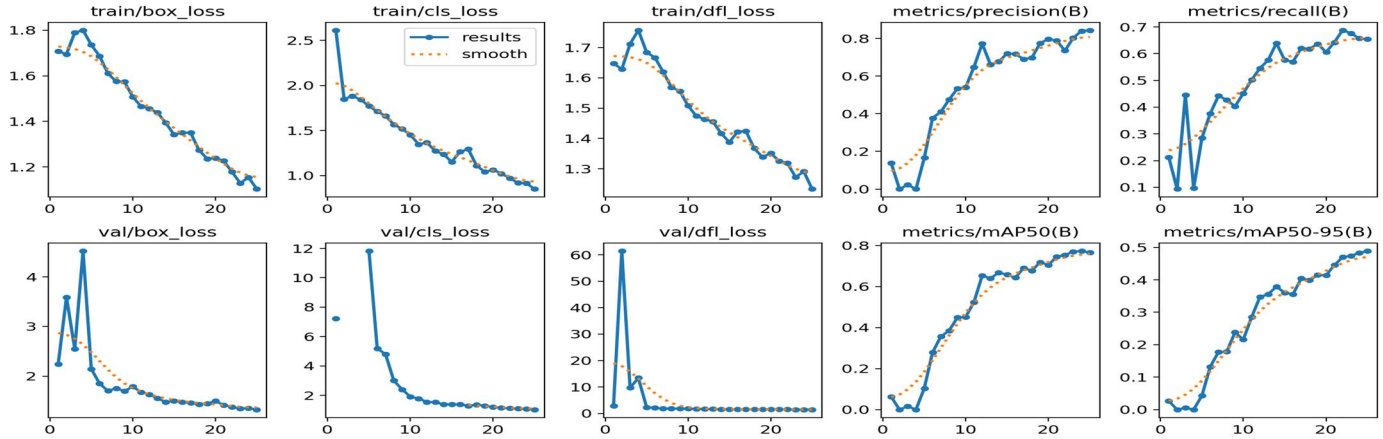

instances
1200
1000
800
600
400
200
0
pothole
y
1.0
0.8
0.6
0.4
0.2
0.0
x
0.0
0.2
0.4
0.6
0.8
1.0
height
width

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

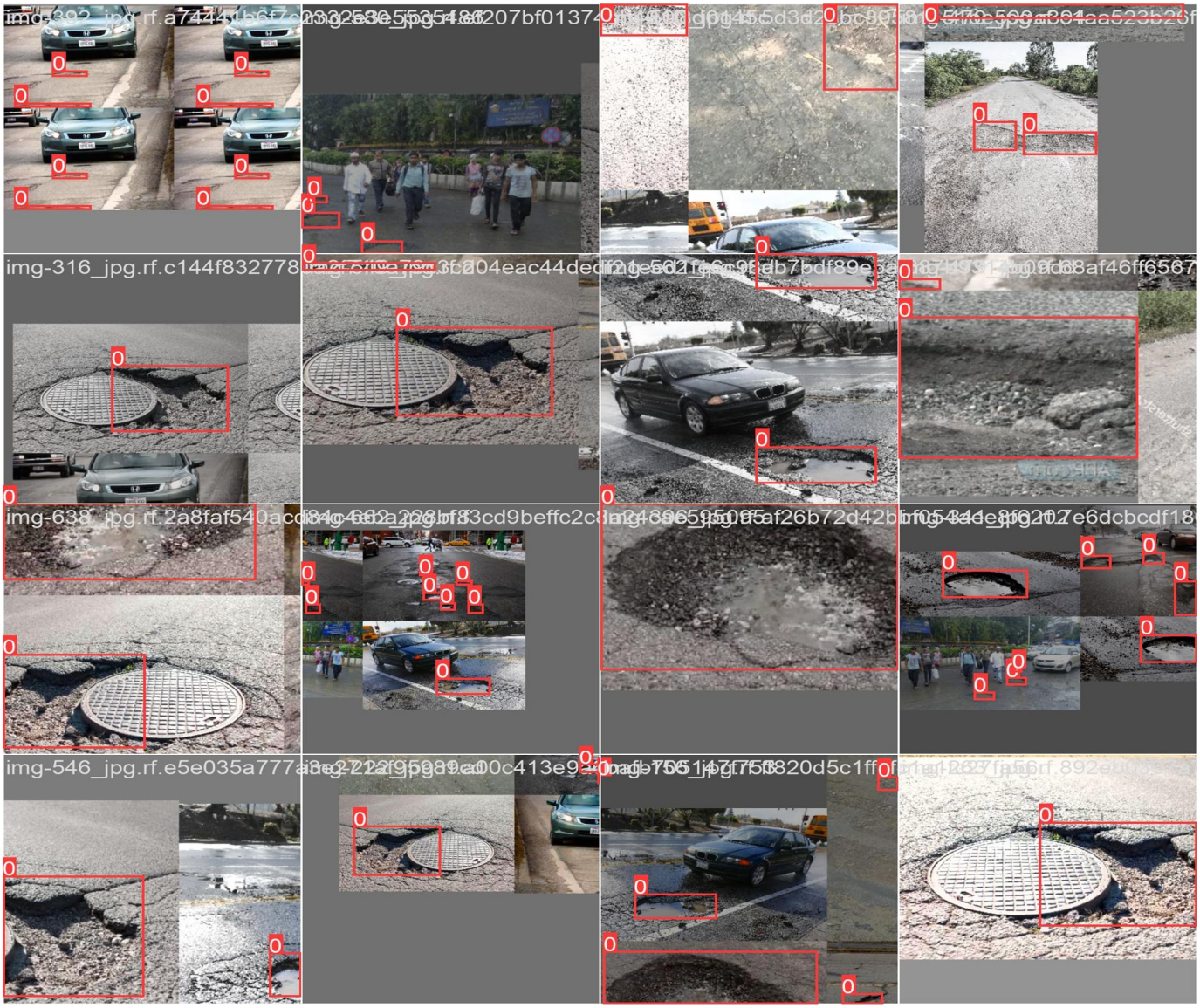
A. Labels



B. Labels Correlogram



C. Results

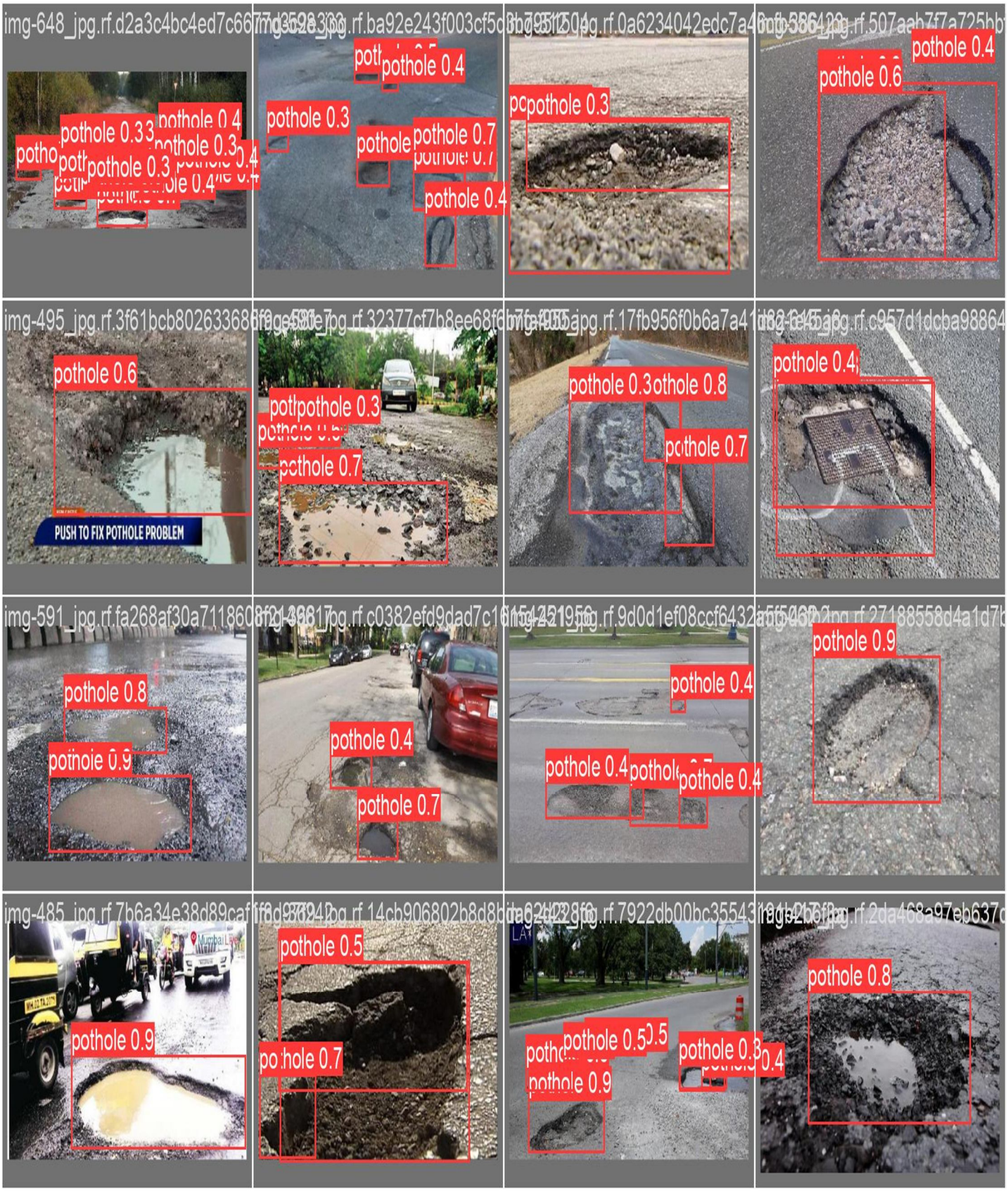


Train_Batch0

Train_Batch1

Val_batch1_labels

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com



Val_batch1_pred

Results.csv(1st 10 columns)

epoch	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precision(B)	metrics/recall(B)	metrics/mAP50(B)	metrics/mAP50-95(B)	val/box_loss	val/cls_loss	val/dfl_loss	lr/pg0	lr/pg1	lr/pg2
1	1.707	2.6127	1.648	0.1383	0.21212	0.06354	0.02617	2.2433	7.221	2.8606	0.00058	0.00058	0.00058
2	1.695	1.8481	1.6285	0.00078	0.09394	0.00043	0.00017	3.5905	inf	61.478	0.0011333	0.0011333	0.0011333
3	1.7896	1.8824	1.7117	0.02268	0.44545	0.01633	0.0055	2.5495	inf	9.6732	0.001639	0.001639	0.001639
4	1.8001	1.8418	1.7573	0.0008	0.09697	0.00044	0.00018	4.5206	inf	13.403	0.0017624	0.0017624	0.0017624
5	1.7353	1.7709	1.6841	0.16564	0.28485	0.1044	0.04315	2.1454	11.82	2.2737	0.0017624	0.0017624	0.0017624
6	1.6866	1.7103	1.667	0.3775	0.37576	0.27904	0.13208	1.8501	5.1879	2.0942	0.0016832	0.0016832	0.0016832
7	1.6106	1.662	1.6195	0.41049	0.44242	0.3564	0.17761	1.7066	4.7963	1.7906	0.001604	0.001604	0.001604
8	1.5752	1.5663	1.5689	0.47501	0.42727	0.38541	0.17832	1.7527	3.0033	1.8504	0.0015248	0.0015248	0.0015248
9	1.5742	1.5183	1.5564	0.53384	0.40303	0.44916	0.23837	1.7015	2.3981	1.7719	0.0014456	0.0014456	0.0014456
10	1.5082	1.4515	1.5081	0.54057	0.45152	0.45215	0.21573	1.7891	1.9072	1.7804	0.0013664	0.0013664	0.0013664

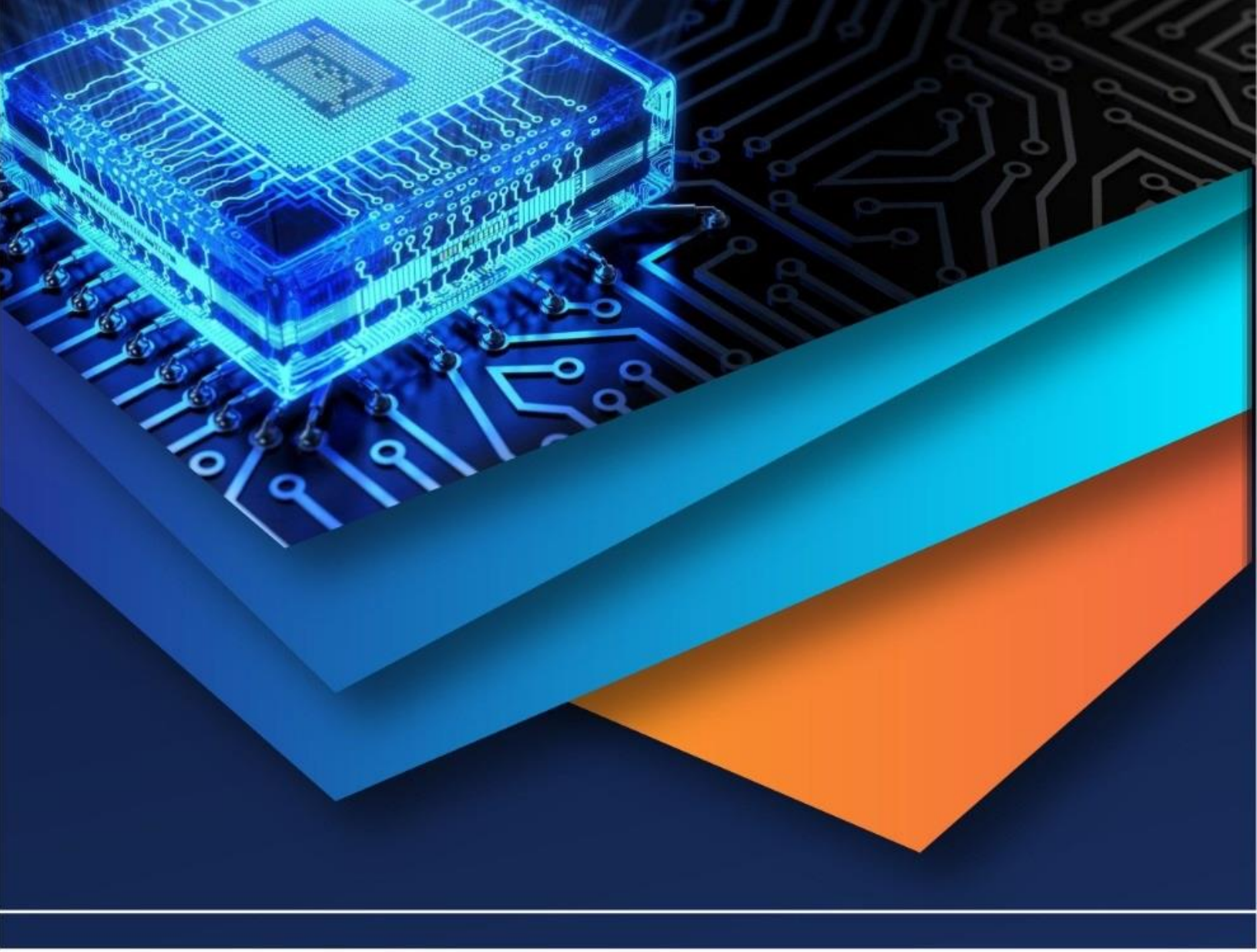
IX. CONCLUSION

In conclusion, integrating pothole detection and object detection technologies offers significant advancements in road safety and infrastructure maintenance. By identifying potholes accurately and detecting objects on the road, such as vehicles and pedestrians, this combined system enhances overall road safety measures. Timely identification of potholes helps authorities prioritize repairs, ensuring smoother and safer roads for drivers and pedestrians alike. Moreover, object detection adds an extra layer of security by alerting drivers to potential obstacles, reducing the risk of accidents. This innovative fusion of technologies not only enhances transportation safety but also contributes to the efficient management of urban road networks, making our roads safer and more reliable for everyone.

REFERANCES

[1] Application of Various YOLO Models for Computer Vision-Based Real-Time Pothole Detection Author : Sung-Sik Park 1 ,Van-Than Tran and Dong-Eun Lee 2
[2] Feasibility study of asphalt pavement pothole properties measurement using 3D line laser technology Author links open overlay panel Xin She a b, Zhang Hongwei c d, Zhou Wang e, Jiao Yan f
[3] Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. arXiv 2018, arXiv:1804.02767.
[4] Bochkovskiy, A.; Wang, C.Y.; Liao HY, M. Yolov4: Optimal speed and accuracy of object detection. arXiv 2020, arXiv:2004.10934.
[5] Malta, A.; Mendes, M.; Farinha, T. Augmented Reality Maintenance Assistant Using YOLOv5. Appl. Sci.2021, 11, 4758.
[6] LeCun, Y.; Kavukcuoglu, K.; Farabet, C. Convolutional networks and applications in vision. In Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, Paris, France, 30 May–2 June 2010; pp. 253–256.
[7] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587
[8] Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448
[9] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37
[10] Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271
[11] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Amsterdam, The Netherlands, 11–14 October 2016; pp. 779–788

[12] Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271
[13] Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. arXiv 2018, arXiv:1804.02767
[14] Bochkovskiy, A.; Wang, C.Y.; Liao HY, M. Yolov4: Optimal speed and accuracy of object detection. arXiv 2020, arXiv:2004.10934
[15] Z. Lu, J. Lu, Q. Ge and T. Zhan, "Multi-object Detection Method based on YOLO and ResNet Hybrid Networks," 2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM), Toyonaka, Japan, 2019, pp. 827-832.
[16] Effect of hyperparameter tuning on classical machine learning models in detecting potholes, Seena Joseph,Alveen Singh.
[17] Indian pothole detection based on CNN and anchor-based deep learning method Mallikarjun Anandhalli, A. Tanuja, Vishwanath P. Baligar & Pavana Baligar International Journal of Information Technology volume 14, pages3343–3353 (2022).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY