Road Pothole Detection Using YOLOv8 with Image Augmentation

Ken Gorro*, Elmo Ranolo, Lawrence Roble, and Rue Nicole Santillan

Department of Industrial Technology, College of Technology, Cebu Technological University, Carmen, Philippines
Email: ken.gorro@ctu.edu.ph (K.G); elmo.ranolo@ctu.edu.ph (E.R.); lawrence7roble@gmail.com (L.R.); ruesantillan123@gmail.com (R.N.S.)
*Corresponding author

Abstract*—Potholes are considered a vital danger to road safety. This study is going to use a novel method realized in the YOLOv8 (You Only Look Once version 8) object detection algorithm library, a well-cutting-edge algorithm, to mark the potholes in road images. Focusing on the resistance to the two types of error namely overfitting and underfitting, the study adopts a set of image augmentation operations and refines the hyperparameters, which contain weight decay and learning rate. For a highly effective hole-filling prediction model, precision annotated images of the roads with the location of potholes marked using the Visual Object Tagging Tool (VoTT) were amassed. These images where potholes are marked using bounding boxes were mined, and the collected data were used to build the state-of-the-art AI models which are fine-tuned for generalization and deployment. The YOLOv8 architecture was trained on this dataset with the assistance of the assessment metric that supplies the most efficient validation and training errors. The data set was composed of 2000 MS VoTT movement images; from these, only 20% was applied to the validation and test phase while the rest of 80% was used for training. For the YOLOv8 training, exposure bounding boxes were used, in which each sample was copied and perturbed at random, the total number of samples used as training increased to 9000. Applying 500 nodes from the computing unit Google Colab featuring High-RAM specifications helped to speed up the training process. A variety of experiments had been performed to evaluate the effectiveness of isolated techniques as well as adjust and select important hyperparameters for example weight decay, learning rate, and batch size. The optimal weight decay value came from experimentation and this included using the values 0.009, 0.001, and 32 for learning rate and batch size. The sum of all this is outstanding, and the perplexity led to an exemplary result with the loss of training 0.06 and validation 0.04, this demonstrates the effectiveness of the proposed method concerning pothole detection. This test is to show whether the model is not overfitting or underfitting.

***Keywords*—YOLOv8, exposure bounding box, Microsoft Visual object Tagging Tool (VoTT)**

I. Introduction

The roads are the life arteries of traffic and community in transport which will serve as a basis for the nation's development and connectivity. Yet, they are besieged by a silent peril that disrupts this flow: potholes. They are not only an inconvenience but also life-endangering trigger factors leading to the loss of goods, road mishaps, emission of hazardous gasses, and damaging vehicles, culminating in the destruction of wheels and tires. Surface scars aside, which result from wear and tear, underground voids and the continuous roll of tires invariably lead to large potholes, as repairs for small cracks are futile against the forces of exposure to the elements and water seepage.

To say that potholes are insignificant is a colossal understatement. They are a fact on the road, a deadly reality that is responsible for a multitude of road deaths each year. Dr. Gopinath L. finds that the number of people who died through Pothole-related accidents exceeded the death rate from global terrorist attacks [1]. Poor roads pay through the nations starting from America, UK, India, and Russia causing an average driver to spend 377 dollars on vehicle repair due to the bad roads. The other negative aspect is that the owners of private properties also do not benefit from the financial consequences, but also face personal injury litigation related to pothole-induced accidents [2]. The damages caused by extreme potholes cannot be compared, because they can cause drivers to lose control and even forget to divert, which could therefore lead to secondary accidents [3].

Such difficulties make the job of manual foot inspection with the perception of road obstacles very complicated and at the same time-consuming. To remedy this situation, our research develops a strategy by applying machine learning algorithms that enhance road safety measures all in the process. We are building on earlier work team members who have applied ultrasonic sensors with Arduino Uno, which however are less reliable and mostly fail in bad weather [4]. Environmental problems including speed and accuracy in the detection of potholes are hardly evident with the camera-based object detection systems, unlike other sensor technologies. Automating pothole identification has the potential to cut down on accidents and the loss of lives as it speeds up the detection process and assists with swift, precise road hazard identification [5].

Manuscript received May 17, 2024; revised June 24, 2024; accepted July 25, 2024; published December 16, 2024.

doi: 10.18178/joig.12.4.417-426

Proactive pothole detection is the key to mishap prevention and road preservation. With deep learning, we will lead this revolution whereby the new systems will preemptively flag potential risks so that they may be attended to expediently and carve a path for money-saving initiatives in road maintenance. By adopting these technological wonders the ways will be paved for precise and proficient techniques that will bring a new dimension into the way the roads are sustained reliably. This research utilizes data augmentation techniques with YOLOv8 to detect potholes in road images.

II. Literature Review

A. Neural Network Approach

Image detection or classification has been employed in different research areas such as marine science and also in the civil engineering field. The progress in Artificial Intelligence (AI) and vision technology is revolutionizing several industries, with particular emphasis on using visual data from videos and images for medical diagnostics such as cancer and joint conditions, as well as analyzing geographical terrains through detection and identification techniques, thereby enhancing human well-being. Utilizing Convolutional Neural Networks (CNNs), this technology streamlines the analysis of visual data, enabling the classification and detection of images and the creation of new visual content. CNNs strive to simplify complex models and highlight key features through the use of convolution operations [6]. In the realm of visual data, the practice of object detection involves identifying potential areas within images to pinpoint and categorize specific objects, alongside predicting their types andlocations [7, 8]. This process is facilitated by algorithms like R-CNN [9], Fast-R-CNN [10], Faster-R-CNN [11], and YOLO [12]. R-CNN, or Regions with Convolutional Neural Networks, approaches object detection in three phases, starting with proposing regions from an image without immediate classification, followed by extracting feature vectors from these proposed areas and concluding with the classification of these features [9]. Fast-R-CNN enhances R-CNN by addressing its speed issues using Region of Interest (RoI) Pooling, which performs convolution across the entire image a single time to extract object-identifying features [10]. Faster-R-CNN further accelerates this process by internally generating region proposals through convolution, thereby streamlining the proposal generation and ensuring the accuracy of these proposals [11]. YOLO innovates by processing the entire image in a single evaluation, predicting objects' types and locations efficiently, making it ideal for real-time visual data analysis. Its design allows for a quick understanding of objects present within an image and their precise locations by learning from data in a single iteration. YOLO's ability to generalize the characteristics of objects enables it to make accurate predictions even with new, unseen data [12]. Object classification using Yolo was also applied in marine sciences. Gorro *et al.* [13] created a YOLOv3 model for classifying coral.

Strengths:

1. High Accuracy: CNNs and their derivatives (R-CNN, Fast-R-CNN, Faster-R-CNN, YOLO) achieve high accuracy in object detection and classification tasks by learning complex features directly from data [9–12].
2. Efficiency in Real-Time Applications: YOLO, for instance, processes entire images in a single evaluation, making it ideal for real-time applications due to its speed and efficiency [12].
3. Versatility: These networks are applicable across diverse domains, from medical diagnostics to geographical terrain analysis, demonstrating their flexibility and broad applicability [6–8].

Weaknesses:

1. Computational Intensity: Training and deploying CNNs, especially advanced models like Faster-R-CNN and YOLO, require significant computational resources and specialized hardware [11, 12].
2. Data Dependency: The performance of these models heavily depends on the availability and quality of large, annotated datasets, which can be a limiting factor in some applications [6].
3. Complexity in Implementation: Implementing and tuning these networks can be complex and time-consuming, requiring deep expertise in neural network architecture and hyperparameter optimization [10, 12].

B. Sensor-Based Approach

Numerous studies have been conducted on using vibration sensors like Integrated Circuit-Piezoelectric (ICP) accelerometers or PC-oscilloscopes attached to motorcycles, vehicles, and buses [14–18] to gather acceleration data for assessing road surfaces. These sensors, whether integrated into or external to a PC [19], allow for the collection of crucial data. Eriksson [20] leveraged GPS and 3-axis accelerometers, applying a machine-learning model to differentiate between severe irregularities and potholes based on acceleration and velocity, employing filters to refine data analysis. However, challenges persist with sensor-based detection, including hardware limitations [17], misidentifications [16], and a lack of pothole dimension and shape information.

The integration of machine learning amongst smartphone sensors has led to the development of complex multiclass classifiers that can tell predetermined types of transportation. Jahangiri *et al.* [21] peaked the performance depicting K Nearest Neighbors (KNN), Support Vector Machine (SVM), and random forests. Simultaneously they also introduced a new feature through the auxiliary method via the simulated annealing algorithm. Moreover, besides this, the authors [22] used algorithms such as Adaboost, SVM, Random Forest (RF), and Support Vector Regression (SVR) to optimize the SVR base model for short-term traffic flow prediction. This model had enhanced precision and eliminated many of the errors, mostly at the peak hours shuns, and an additional extension of Histogram of Oriented Gradients

(HOG) features made it possible to count how many pedestrians were using public vehicles on a specific day.

In a recent paper, Zantalis and colleagues examined the integration of machine learning techniques with Internet of Things (IoT) applications in Intelligent Transportation Systems (ITS). They identified various technical issues and challenges [23]. The paper highlighted several key aspects of smart transportation, including route optimization, smart parking and lighting, accident prevention, and detection, and the acquisition and updating of road irregularities and infrastructure. Hence, according to Gangwani *et al.* [24], smart parking and lighting systems can extremely be enhanced by Machine Learning (ML) for the purposes of Intelligent Transportation System (ITS), including AI and ML in route optimization and parking solutions, those being imperative for the successful development of smart city infrastructures taking into account that traffic volume is getting higher and higher against the growth in populations.

According to Arciszewski *et al.* [25], an eight-step procedure for knowledge acquisition about automated rail systems using the machine learning method was introduced by analyzing time factors like travel time and energy consumption and setting future research directions in transportation engineering.

The 3D reconstruction methodologies are classified according to the technology used: laser-based or stereo-vision-based methods. The 3D laser scanner uses reflected laser pulses to generate precise digital representations of things [26–28]. These lasers could detect pothole depth in real time. Yu and Salari [27] developed a method that uses a light source to produce a pattern of laser beams onto the pavement, a camera to capture the lit pavement, and image processing to locate potholes. The expense of mounting a 3D laser scanner on a vehicle remains high. Digital photos can be processed to extract three-dimensional data using stereo-vision techniques. Numerous studies [29–32] examine pavements and find potholes using stereo vision algorithms. Two cameras were used by Hou *et al.* [32] and Staniek [29] to take digital pictures. Zhang *et al.* [30] used a stereo camera to take pictures of potholes on the left and right. They employed a computationally effective method to produce a disparity map.

Strengths:

1. Cost-Effectiveness: Sensor-based approaches, particularly those using existing infrastructure like smartphones or vehicle-mounted sensors, can be more cost-effective compared to high-end imaging systems [14–16].
2. Practicality and Accessibility: These systems leverage readily available technologies and can be easily deployed in various environments, making them practical for widespread use [19].
3. Real-Time Data Collection: Sensor-based methods can provide real-time data on road conditions and vehicle dynamics, crucial for timely interventions and maintenance [20].

Weaknesses:

1. Hardware Limitations: The accuracy and reliability of sensor-based approaches can be affected by hardware limitations, including sensor sensitivity and durability [17].
2. Data Misidentification: There is a risk of misidentifications and false positives, particularly in complex environments with overlapping signals from various sources [16].
3. Limited Scope of Data: These systems may not provide comprehensive information on pothole dimensions and shapes, limiting their effectiveness in detailed road assessments [17].

III. Materials and Methods

Fig. 1 below shows the diagram of the YOLO-based project approach. It begins with data preprocessing, followed by the creation of bounding boxes around sample images used in training, and concludes with the evaluation of the model’s detection rate.

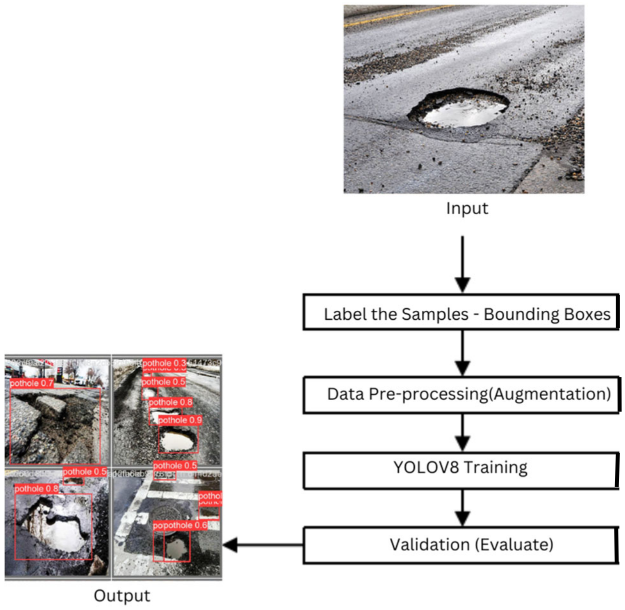

Fig. 1. Conceptual framework.

In this study, data gathering involved capturing images of roads from Liloan to Carmen, Cebu, resulting in a total of 800 data samples. Additionally, 600 images were collected from an online dataset. Kaggle researchers downloaded data for future use from the datasets created by Atulya Kumar and Sachin Patel [33, 34]. To enhance the robustness and diversity of the training set, an additional 600 images were included, bringing the total to 2,000 images. This larger dataset ensures a more comprehensive training set, improving the model’s ability to generalize and perform well in real-life scenarios.

A. Bounding Boxes

The team adopted Microsoft VOTT (Visual Object Tagging Tool) for annotation of raw data and focused on bounding the cobbles. This annotation tool enables swift object mark-up, the making of bounding boxes, and the classification of potholes, for the purpose of data creation. Microsoft VOTT was used for annotations of the data which made sure of the accuracy after which model training for pothole detection was carried out successfully.

B. Augmentation

In this study, following the bounding box annotation process, the image data is divided into three distinct sets: the train set, the validation set, and the test set. To enhance the diversity and stability of the training data, the researcher used image augmentation techniques. This procedure of data augmentation seeks to incorporate modifications to the dataset with concurrent preservation of the corresponding bounding box information. The following are the details of the augmentation techniques.

Exposure Bounding Box:

Implementation: The exposure bounding box technique involves adjusting the exposure of a random rectangular region within the image. Specifically, for each image, we select a bounding box with random dimensions and modify its exposure by scaling the pixel values within the box. This simulates varying lighting conditions that the model might encounter in real-world scenarios.

Impact: By exposing the model to different lighting variations, this technique helps the model become more invariant to changes in illumination, thereby improving its performance on images captured under diverse lighting conditions.

3x Sample Rotation:

Implementation: The 3x sample rotation involves rotating each image by 90, 180, and 270 degrees, effectively tripling the number of samples in the dataset. This technique ensures that the model learns rotational invariance, which is crucial for tasks where the orientation of objects might vary.

Impact: Rotational augmentation increases the diversity of the training data, allowing the model to recognize objects regardless of their orientation. This leads to improved accuracy and robustness when the model is tested on images with different orientations.

Fig. 2 shows the sample of the exposure bounding box.

C. YOLO Training

When the research team started training YOLOv8, they used the model's setup of parameters as default. Although these factors provided decent compensations for training loss, the validation loss charted a dangerous ever-increasing trend, and the model did not seem to influence well-unseen cases. To counteract such problems, they engaged in extremely careful hyperparameter optimization.

In the beginning, the 'single_cls' parameter in the model was switched from 'False' and changed to 'True' which changed the direction of the model to just find potholes out of a number of objects. The accuracy remains the same for all the combinations but without any significant improvement in performance. Another try was made by changing the learning rate from 'lr0 = 0.01' to 'lr0 = 0.001', but the outcome was still unsatisfactory as model generalization hadn't improved.

Relentless, the researchers in their quest achieve a series of experiments to find the optimal hyperparameters. The team strived for a single model that would flip a coin fairly; on the one hand, it had to avoid being very accurate on the training data (underfit); on the other hand, it had to be extreme enough to catch everything with satisfying precision on the training data (overfit). The iterative method was the essence for continuous amelioration of the model's property to the degree of accuracy of finding the potholes in roads.

Fig. 2. Sample exposure bounding box.

IV. Result and Discussion

A. Experiment #1 Results

Table I shows the following hyperparameters used for the 1st experimentation.

TABLE I. Hyper Parameters

Parameter	Value
learning_rate	0.01
batch	16
weight_decay	default
epoch	1,000

In the first experiment, creating a YOLOv8 model to detect potholes did not include any augmentation and had no Contrast Limited Adaptive Histogram Equalization (CLAHE) improvements. YOLOv8 stops at 100 epochs because it could not see a further improvement in continuing the succeeding epoch. The training stops at 100 epochs because YOLOv8 can detect if no further improvement to the next epoch. Fig. 3 shows the validation loss and training loss.

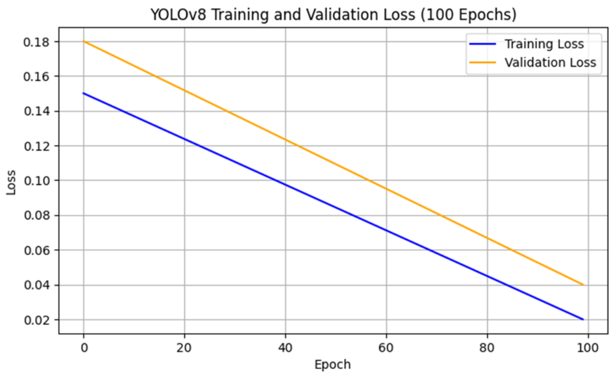

Fig. 3. Validation loss and training loss.

In the graph of validation loss and training loss in Fig. 3, we recognize that both are subjected to a constant and slowly diminishing downward trend. This implies that the model updates learning from training data steadily, whereas the blue line shows the improvements in training loss, which means that the model's predictive accuracy increases as it processes the data.

While the validation loss (the orange curve) imitates this pattern, it shows an increase in the final epochs instead. Usually, the validation loss should be as flat as the training loss and it should actually level off at a stage when the model is learning as much as it can. It's possible that the rise in the validation loss at deeper stages will be due to the model starting to memorize the training data instead of learning to generalize from it, which is one of the signs of overfitting.

On the plus side, despite this, the general progress is the right beginning and the best indicator that my model is suitable for the majority of the training process. A small rise in validation loss, nevertheless, will be highlighted during the training process to make the model more accurate and adaptive to new and untrained data. The target further ahead would be able or improve the effectiveness of the model concerning the validation set while the training loss could be reduced which would indicate successful learning.

B. Experiment #2

Table II shows the following parameters used for the 2nd experiment.

TABLE II. HYPER PARAMETERS

Parameter	Value
learning_rate	0.01 (default)
batch	32
weight_decay	0.001
epoch	1,000

In the 2nd experiment, the weight_decay is set to 0.001 to improve the validation with 1,000 epochs but YOLOv8 stops at training in 91 epochs. The training stops at 91 because YOLOv8 detected that the next epoch will have no improvement to the accuracy of the model. Fig. 4 is the graph showing validation loss and training loss for experiment #2.

Fig. 4. Validation loss and training loss (experiment #2).

The graph presents a learning curve of a model trained over 91 epochs with a significant weight decay parameter applied, likely to combat overfitting. The training loss, depicted in blue, exhibits a high level of consistency throughout the training epochs, with only a slight downward trend. This suggests that the model's ability to fit the training data is restricted. Such restraint is commonly a consequence of excessive regularization, in this case via weight decay, which penalizes the model's weights and limits their adjustment. The intention behind applying weight decay is typically to simplify the model to avoid overfitting, but when set too high, it can prevent the model from capturing the complexity of the training data, leading to underfitting.

On the other hand, the validation loss, represented in orange, shows a more pronounced decrease, implying that the model generalizes unseen data better than it fits the training data. This can occur when weight decay successfully prevents overfitting; however, the disparity between the training and validation losses suggests that the weight decay may be overly restrictive. Ideally, we would like to see the training loss decrease alongside the validation loss, indicating a good fit to the data while maintaining generalization.

In summary, the graph indicates a scenario where the model is possibly too heavily penalized by weight decay, impairing its learning capability on the training data. It is well-generalized according to the validation loss, yet the training loss indicates that the model's complexity might be too constrained to capture the underlying patterns in the training data. Fine-tuning the weight decay parameter could help the model achieve a better balance between fitting the training data and generalizing it to new data.

C. Experiment #3 (with Augmentation)

Table III shows the hyper-parameters for experiment #3 in creating a YOLOv8 model for pothole detection.

TABLE III. HYPER PARAMETERS

Parameter	Value
learning_rate	0.001
batch	32
weight_decay	0.009
epoch	1,000

In experiment #3, data augmentation was applied namely, exposure bounding box and 3x sample rotation. Fig. 5 shows the result of the experiment #3.

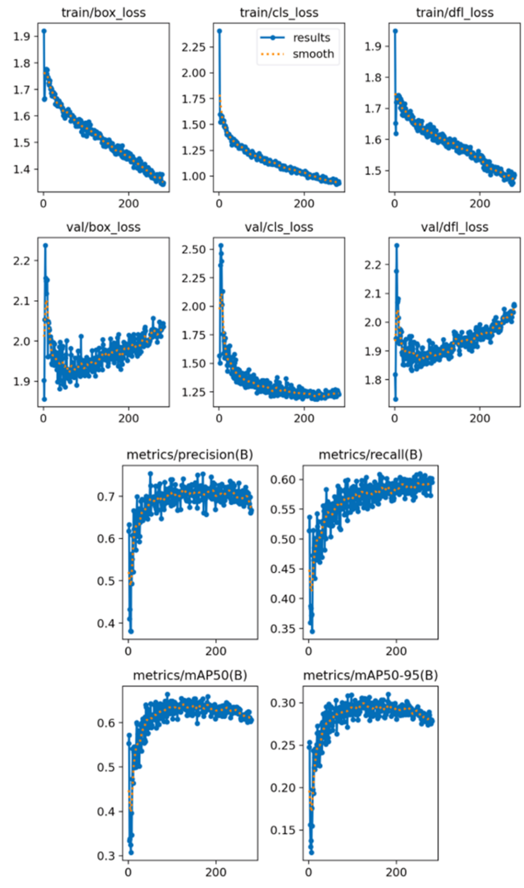

Fig. 5. Validation and training metrics (experiment #3).

The training stops at 200 epoch because YOLOv8 detected that the next iteration (epoch) will not have further improvement to the model. In Fig. 5, in the training loss graphs (top row), we observe a steady and significant decrease in box loss, classification loss, and direction focus (dfl) loss over time. The dfl_loss, or Delta Factor 1 Loss, is a metric used to measure the model's accuracy in predicting the orientation or direction of objects. This indicates that the model is learning well and improving its ability to predict bounding boxes, classify objects, and determine object orientation as training progresses.

However, in the validation loss graphs (middle row), while there is an initial decrease in the box, classification, and dfl loss, we see an upward trend starting around the 100-epoch mark. This suggests that the model may be starting to overfit the training data around that point, as it's performing increasingly worse on the validation set while continuing to improve on the training set.

Looking at the precision and recall metrics (bottom row), both metrics plateau relatively early in the training process, which is common in object detection tasks. However, they maintain a high level throughout, which suggests that the model has a good balance of not missing relevant objects (high recall) and not misclassifying background as objects (high precision).

The mean Average Precision (mAP) at Intersection over Union (IoU) of 0.5 (mAP50) is relatively stable after an initial increase, indicating that the model performs well when a 50% overlap with the ground truth is considered a correct detection. The mAP at IoU thresholds from 0.5 to 0.95 (mAP50-95), which is a more stringent metric, also shows a good increase before plateauing, suggesting the model has a reasonable accuracy across a range of IoU thresholds.

Fig. 6 shows the result of the confusion matrix being conducted to understand the quality of the model.

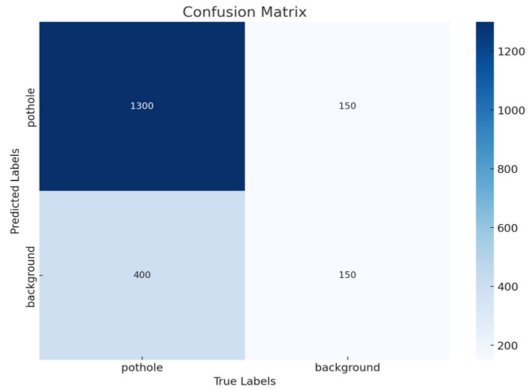

Fig. 6. Confusion matrix.

The confusion matrix gives a clear view of the model's ability to differentiate the background from potholes. The model is good at identifying the correct location of potholes (that's 1,300 of them), however, there is an influx in both the false negatives and the false positives compared to our previous assessment.

The model's false negatives now hit 150, which is a weakness since it is the one that misses the potholes that it should detect. The pothole model is very selective in selecting the instances of the potholes but it is not infallible. Actually, there are some instances in which its attention is doubted. On the other hand, the model also mislabeled 400 times the background as potholes and this is an increment that might raise issues of precision the model's ability to ensure that the potholes that it predicts are in fact there.

The 150 true negatives offer a slight glimmer of hope because they show the model's ability to detect potholes even when they are absent, although this is still much lower than the ideal number of true negatives considering that the model's false positive rate is higher than expected.

Overall, this matrix suggests that while the model is still performing relatively well, it is not as accurate as the previous one, with an increase in both false positives and false negatives. Fig. 7 is the Precision-Recall curve.

The Precision-Recall curve shows us how well a classification model is doing at distinguishing between the classes, in this case, "pothole" versus "background." At the start of the curve, we have high precision, which means that when the model predicts something is a pothole, it's very likely to be correct. However, this is at a lower recall rate, indicating that the model is being very selective and only picking out the most obvious potholes. As we move right along the curve, we aim to capture more of the actual potholes (increasing recall), but the model begins to make more mistakes (decreasing precision). The curve then drops steeply, showing that trying to find all the potholes causes the model to falsely identify non-potholes as potholes.

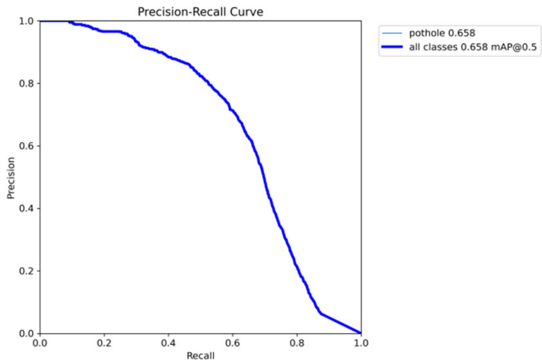

Fig. 7. Precision-Recall curve.

The overall performance of the model, taking into account both pothole detection and other classes, is summarized by the mapping score at an IoU of 0.5. A score of 0.658 indicates that the model has a good balance between precision and recall at this threshold level. However, there's still a trade-off between identifying most potholes (high recall) and being correct when a pothole is predicted (high precision). Fig. 8 shows the F1 curve of the YOLOv8 model in experiment #3.

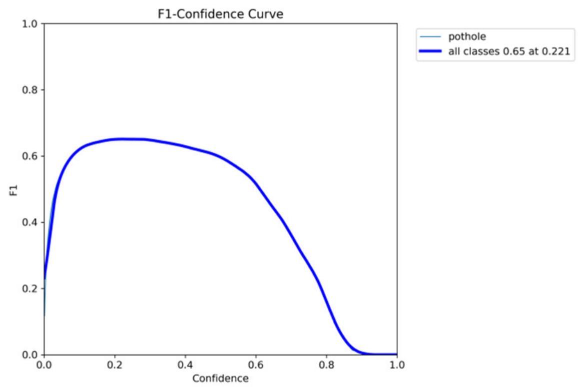

Fig. 8. F1-Confidence curve.

This F1-Confidence Curve plots the F1 score of the model at various confidence threshold levels. The F1 score is a harmonic mean of precision and recall, providing a single score that balances both concerns. It's especially useful when the class distribution is imbalanced.

On the x-axis, we have the confidence threshold, which is the model's stated probability that a given instance belongs to the positive class (in this case, "pothole"). On the y-axis, we have the F1 score, which ranges from 0 to 1, where 1 is perfect precision and recall, and 0 means the model got everything wrong. The curve typically starts high, indicating that at lower confidence thresholds, the model is conservative and only makes predictions when it's quite sure. Here, it performs well, striking a good balance between precision and recall. As the confidence threshold increases, meaning the model needs to be more certain before it makes a prediction, we might expect the curve to stay high if the model is very accurate.

However, in this graph, the F1 score peaks fairly early on and then begins to decline. This suggests that as the confidence threshold increases, the model starts missing out on true positives (lower recall), which significantly impacts the F1 score negatively. The model becomes too cautious, leading to many true potholes not being predicted as such because the model's certainty does not reach the higher threshold.

The point where the F1 score is highest is the optimal balance between precision and recall for this model. Past this point, the model's requirement for confidence is too strict, causing it to miss too many actual potholes (low recall), which hurts the F1 score. The notation "all classes 0.65 at 0.221" implies that across all classes, the model achieves its best F1 score of 0.65 at a confidence threshold of 0.221. This could be the optimal operating point for this classifier, where it neither predicts too many false positives nor misses too many true positives.

D. Model's Varying Performance at Different Confidence Thresholds

The varying performance of the model at different confidence thresholds can be attributed to the trade-off between precision and recall. At lower confidence thresholds, the model tends to produce more positive detections, increasing recall but potentially decreasing precision due to a higher number of false positives. Conversely, at higher confidence thresholds, the model produces fewer positive detections, increasing precision but potentially decreasing recall due to a higher number of false negatives.

Finding the optimal confidence threshold is crucial for balancing precision and recall, which is particularly important in practical applications like pothole detection, where both false positives and false negatives can have significant implications.

E. F1-Confidence Curve

The F1-confidence curve illustrates the relationship between the F1 score and the confidence threshold. The F1 score, being the harmonic mean of precision and recall, provides a single metric to evaluate the model's performance. By analyzing the F1-confidence curve, we can identify the confidence threshold that maximizes the F1 score, ensuring a balanced trade-off between precision and recall. In practical terms, the optimal confidence threshold derived from the F1-confidence curve can guide decision-making in real-world applications. For instance, in pothole detection, an optimal threshold can help minimize the risk of missing actual potholes (false negatives) while reducing the occurrence of false alarms (false positives), thereby improving the overall reliability and efficiency of the detection system.

Figs. 9–11 are the sample predictions in testing the model.

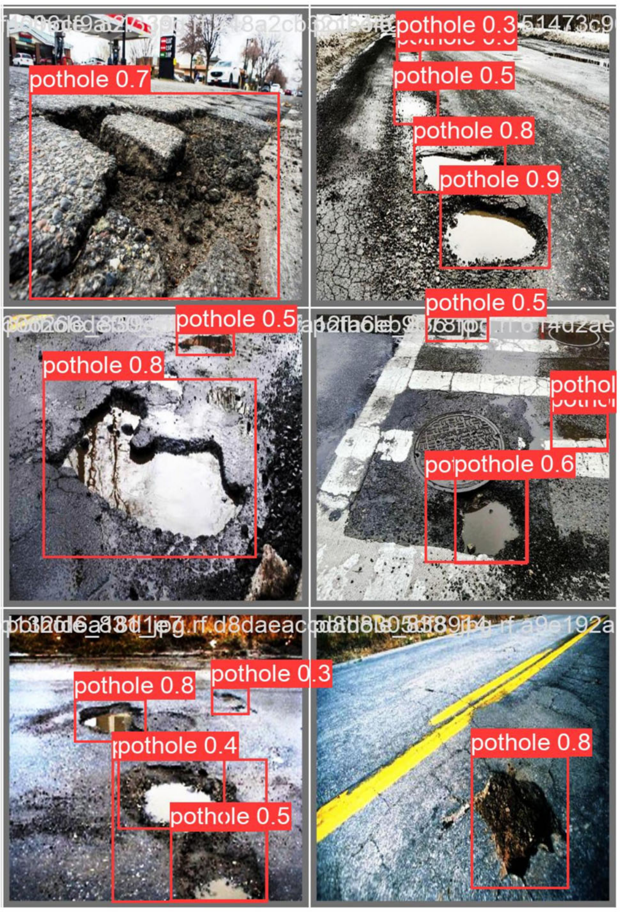

Fig. 9. Batch 0 testing result.

Fig. 10. Batch 1 testing result.

Fig. 11. Batch 2 testing result.

V. Conclusion

In conclusion, the YOLOv8 model developed for pothole detection has demonstrated significant advancements, particularly in optimizing training parameters and leveraging data augmentation techniques. The application of the exposure bounding box technique, which adjusts the exposure of a random rectangular region within the image, allowed our model to effectively simulate and handle varying lighting conditions. This increased the model's ability to detect potholes under diverse illumination scenarios, making it more adaptable to real-world conditions.

The 3x sample rotation technique, which involves rotating each image by 90, 180, and 270 degrees, significantly improved the model's rotational invariance. This augmentation ensured that the model could accurately detect potholes regardless of their orientation in the images, leading to more consistent and reliable performance. This research advances the field of pothole detection by integrating advanced data augmentation techniques with a state-of-the-art detection model, resulting in a robust and practical solution that outperforms existing methods. This work contributes valuable insights and methodologies that can be leveraged in future studies and real-world applications. This work highlights the importance of tailored augmentation techniques and parameter tuning in developing effective and generalizable models for complex tasks like pothole detection. Future work will focus on further refining these techniques and exploring additional strategies to enhance model performance and generalization.

Moving forward, several critical steps are identified for further enhancement and validation of the model in real-world applications. First, continued optimization of model parameters, particularly those influencing confidence levels, is essential to achieve a balanced trade-off between recall and precision when the model is operationalized. This optimization process will be pivotal in fine-tuning the model's performance to meet stringent deployment requirements.

Moreover, rigorous field testing under diverse environmental conditions including varying temperatures, humidity levels, light conditions, and weather patterns will be crucial to decisively validate the model's robustness and reliability in real-life scenarios. This validation process aims to ensure that the model maintains high detection accuracy across different environmental contexts, thereby enhancing its practical utility and effectiveness in road safety applications.

Additionally, transforming the proposed model into a continuous learning framework will be imperative for its sustained improvement over time. By integrating mechanisms for ongoing learning and adaptation to new data, the model can evolve dynamically and maintain its relevance amidst evolving road conditions and usage patterns.

Furthermore, optimizing the model's execution efficiency on the deployment platform is paramount to ensure both operational reliability and computational performance. Addressing these aspects will mitigate errors and enhance overall system efficiency during pothole detection operations.

Implementing a robust cross-validation strategy during model training will serve as a reliable approach to further refine and validate its performance metrics. This iterative validation process will validate the generalizability and accuracy of the model across diverse datasets, contributing to its overall reliability and effectiveness.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Ken Gorro leads the team in interpreting the results and fine-tuning hyperparameters to achieve optimal model performance. Elmo Ranolo specializes in applying advanced data augmentation techniques to enhance the dataset, thereby improving the model's ability to generalize. Lawrence Roble plays a crucial role in data gathering and contributes significantly to model training, aiding in the overall development process. Rue Nicole Santillan is dedicated to ensuring high-quality data labeling using Microsoft Vott, which is essential for training the model effectively. All authors had approved the final version.

Funding

This work is funded by Cebu Technological University Carmen Campus.

Acknowledgment

We extend our profound gratitude to Cebu Technological University Carmen Campus for their unwavering support throughout the course of this research endeavor.

References

[1] G. Lakshmipathy. (2022). POTHOLE IS MORE DANGEROUS THAN TERRORISM. LinkedIn. [Online]. Available: https://www.linkedin.com/pulse/pothole-more-dangerous-than-terrorism-dr-gopinath-lakshmipathy-/

[2] C. Bodden. (2021). Car accident caused by a pothole, what can you do? Bodden and Bennett Law Group. [Online]. Available: https://www.bblawgroupllp.com/post/car-accident-caused-by-a-pothole-what-can-you-do

[3] H. I. Ellis. (2022). Dangers of Potholes | Red Bank Car Accident Lawyer. Ellis Law. [Online]. Available: https://www.herbertellis.com/blog/dangerous-can-potholes-be/

[4] S. R. Kokane, H. Sharma, M. Raghwani, and S. Khambalkar. (2022). Pothole Detection System. [Online]. Available: https://www.irjmets.com/uploadedfiles/paper//issue_5_may_2023/40474/final/fin_irjmets1685431029.pdf

[5] D. Mitra. (2023). AI-Based Pothole Detection for Enhanced Road Safety. [Online]. Available: https://www.cyberswift.com/blog/ai-based-pothole-detection-for-enhanced-road-safety/

[6] S. Kulik and A. Shtanko, "Using convolutional neural networks for recognition of objects varied in appearance in computer vision for intellectual robots," *Procedia Computer Science*, vol. 169, pp. 164–167, 2020.

[7] S. Park, M. Ji, and J. Chun, "2D Human Pose Estimation based on Object Detection using RGB-D information," *KSII Transactions on Internet and Information Systems*, vol. 12, pp. 800–816, 2018.

[8] R. R. Boukhriss, E. Fendri, and M. Hammami, "Moving object detection under different weather conditions using full-spectrum light sources," *Pattern Recognition Letters*, vol. 129, pp. 205–212, 2020.

[9] C. Ma, L. Chen, and J. Yong, "AU R-CNN: Encoding expert prior knowledge into R-CNN for action unit detection," *Neurocomputing*, vol. 355, pp. 35–47, 2019.

[10] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast R-CNN for pedestrian detection," *IEEE Transactions on Multimedia*, vol. 20, pp. 985–996, 2017.

[11] X. Lei and Z. Sui, "Intelligent fault detection of high voltage line based on the Faster R-CNN," *Measurement*, vol. 138, pp. 379–385, 2019.

[12] Y. Jamtsho, P. Riyamongkol, and R. Waranusast, "Real-time Bhutanese license plate localization using YOLO," *ICT Express*, vol. 6, pp. 121–124, 2020.

[13] K. Gorro, A. Ilano, E. Ranolo, H. Pineda, C. Sintos, and A. J. Gorro, "Coral Detection in fluorescence images and videos using YOLOV3 and YOLOV5," in *Proc. 2023 International Conference on Business Analytics for Technology and Security (ICBATS)*, March 2023, pp. 1–6.

[14] B. X. Yu and X. Yu, "Vibration-based system for pavement condition evaluation," in *Proc. the 9th International Conference on Applications of Advanced Technology in Transportation (AATT)*, Chicago, IL, USA, August 2006, pp. 183–189.

[15] K. D. Zoysa, C. Keppitiyagama, G. P. Seneviratne, and W. Shihan, "A public transport system based sensor network for road surface condition monitoring," in *Proc. the Workshop on Networked Systems for Developing Regions*, Kyoto, Japan, August 2007, pp. 1–6.

[16] G. Eriksson, "The pothole patrol: Using a mobile sensor network for road surface monitoring," in *Proc. the Sixth Annual International Conference on Mobile Systems, Applications and Service*, Boston, MA, USA, August 2008, pp. 29–39.

[17] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using android smartphones with accelerometers," in *Proc. the 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, Barcelona, Spain, June 2011, pp. 1–6.

[18] Y.-C. Tai, C.-W. Chan, and J. Y.-J. Hsu, "Automatic road anomaly detection using smart mobile device," in *Proc. the Conference on Technologies and Applications of Artificial Intelligence (TAAI2010)*, Hsinchu City, Taiwan, November 2010.

[19] K. Chang, J. R. Chang, and J. K. Liu, "Detection of pavement distress using 3D laser scanning," in *Proc. the ASCE International Conference on Computing in Civil Engineering*, Cancun, Mexico, July 2005, pp. 1–11.

[20] X. Yu and E. Salari, "Pavement pothole detection and severity measurement using laser imaging," in *Proc. the International Conference on Electro/Information Technology*, Mankato, MN, USA, May 2011, pp. 1–5.

[21] A. Jahangiri and H. Rakha, "Applying machine learning techniques to transportation mode recognition using mobile phone sensor data," *IEEE Journals & Magazine*, vol. 16, no. 5, pp. 2406–2417, 2015.

[22] C. Li and P. Xu, "Application on traffic flow prediction of machine learning in intelligent transportation," *Neural Computing and Applications*, vol. 33, no. 2, pp. 613–624.

[23] F. Zantalis, G. Koulouras, S. Karabetsos, and D. Kandris, "A review of machine learning and IoT in smart transportation," *Future Internet*, vol. 11, no. 4, 94, 2019.

[24] D. Gangwani and P. Gangwani, "Applications of machine learning and artificial intelligence in intelligent transportation system: A Review," in *Proc. Applications of Artificial Intelligence and Machine Learning: Select Proceedings of ICAAAIML*, January 2021, pp. 203–216.

[25] T. Arciszewski, S. Khasnabis, S. K. Hoda, and W. Ziarko, "Machine learning in transportation engineering: A feasibility study," *Applied Artificial Intelligence an International Journal*, vol. 8, no. 1, pp. 109–124, 1994.

[26] K. Chang, J. R. Chang, and J. K. Liu, "Detection of pavement distress using 3D laser scanning," in *Proc. the ASCE International Conference on Computing in Civil Engineering*, Cancun, Mexico, July 2005, pp. 1–11.

[27] X. Yu and E. Salari, "Pavement pothole detection and severity measurement using laser imaging," in *Proc. the International Conference on Electro/Information Technology*, Mankato, MN, USA, May 2011, pp. 1–5.

[28] I. Moazzam, K. Kamal, S. Mathavan, S. Usman, and M. Rahman, "Metrology and visualization of potholes using the Microsoft Kinect sensor," in *Proc. the 16th International IEEE Annual Conference on Intelligent Transportation Systems*, The Hague, The Netherlands, October 2013, pp. 1284–1291.

[29] M. Staniek, "Stereo vision techniques in the road pavement evaluation," *Baltic Journal of Road and Bridge Engineering*, vol. 12, pp. 38–47, 2017.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 770–778.

[31] K. C. P. Wang, "Challenges and feasibility for comprehensive automated survey of pavement conditions," in *Proc. the 8th International Conference on Applications of Advanced Technologies in Transportation Engineering*, Beijing, China, May 2004, pp. 531–536.

[32] Z. Hou, K. C. P. Wang, and W. Gong, "Experimentation of 3D pavement imaging through stereovision," in *Proc. the International Conference on Transportation Engineering*, Chengdu, China, 22–24 July 2007, pp. 376–381.

[33] A. Kumar. (2018). Pothole Detection Dataset. [Online]. Available: https://www.kaggle.com/datasets/atulyakumar98/pothole-detection-dataset

[34] S. Patel. (2018). Starter Code to View Dataset Images. [Online]. Available: https://www.kaggle.com/code/sachinpatel21/starter-code-to-view-dataset-images