IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Smart Road Damage Detection For Safer Roads: Implementation And Challenges

Mr. Ketan Singh, Dr. Alka Verma (Guide), Mr. Neeraj Kaushik (Co-Guide)

M.Tech(Machine Learning & Data Science), Associate Professor, Assistant Professor
Department of Electronics & Communication Engineering Faculty of Engineering,
Teerthanker Mahaveer University, Moradabad, India

Abstract: Increase in the number of potholes have serious impact on road safety and infrastructure, leading to increased costs for vehicle repairs and accidents. Why? Even with manual inspections and sensor-based systems, pothole detection is not an option. A real-time pothole detection system using deep learning techniques, built on the YOLO (You Only Look Once) ONNX model is presented in this article. This involves gathering data, generating model data and testing mobile and vehicle-mounted applications over the course of several months. It was 92% accurate in detection and had an adequate high confidence level estimate (ROC-AUC) score, while also maintaining proper balance between precision and recall. Other concerns we tackle include differences in environment between samples, inaccurate data detection systems, and hardware failures.

Keywords: *Road safety, object detection, YOLO, real-time, machine learning, image processing.*

1. INTRODUCTION

Transport efficiency and safety are closely linked to the development of road infrastructure. Potential potholes can cause damage to vehicles, create accidents and increase maintenance costs. What are the dangers? Inefficient and error-prone methods, manual inspections or basic image processing techniques are used for traditional pothole detection. In real-time applications, deep learning-based object detection models like YOLO (You Only Look Once) have shown greater success. Our model is optimized for efficient cross-platform deployment through the use of ONNX (Open Neural Network Exchange). This article presents an in-depth analysis of the effectiveness of a pothole detection system that utilizes YOLO and ONNX, which has been developed and implemented.

YOLO's real-time detection capability makes it suitable for integration into:

- **Mobile applications**, where users can scan roads for potholes.
- **Vehicle-mounted cameras**, enabling automated road condition monitoring.
- **Smart infrastructure**, using cloud-based pothole reporting for urban planning.

This paper presents an **end-to-end AI-based pothole detection system** that is:

- **Accurate**: Achieves over 90% detection accuracy.
- **Fast**: Uses **ONNX Runtime** for optimized inference.
- **Scalable**: Can be deployed on mobile devices or embedded hardware.

2. SYSTEM DESIGN AND IMPLEMENTATION

2.1 Architecture Overview

The proposed system consists of three primary components:

1. **Data Collection & Annotation** – Gathering diverse pothole images and labeling them for model training.
2. **Model Training & Optimization** – Using YOLO to train a pothole detection model and optimizing it for ONNX.
3. **Deployment & Real-World Testing** – Integrating the model into mobile and vehicle-mounted applications.

3. DATA COLLECTION AND PREPROCESSING

3.1 Dataset Collection

To ensure model generalization, we collected a diverse dataset of **5,000 images** from various sources:

- **Public datasets**: Road damage datasets from Kaggle and government repositories 33.
- **Dashcam images**: Captured from urban, highway, and rural roads.
- **Crowdsourced images**: Road condition images submitted by volunteers.

3.2 Data Annotation

Each image was labeled using **LabelImg**, following the **YOLO annotation format**:

<class_id> <x_center> <y_center> <width> <height>

Example:

0 0.45 0.56 0.12 0.10

Here, 0 represents the **pothole class**, and the values define the bounding box.

3.3 Data Augmentation

To enhance model performance under different conditions, augmentation techniques were applied:

- **Brightness adjustments**: Simulating different lighting conditions.
- **Rotation & scaling**: Making the model robust to different viewing angles.
- **Blurring & noise addition**: Ensuring performance in low-quality images.

4. AI MODEL TRAINING AND OPTIMIZATION

4.1 Model Selection

We selected **YOLOv5** due to its balance between accuracy and inference speed.

4.2 Training Configuration

- **Hardware**: NVIDIA RTX 3090 GPU.
- **Hyperparameters**:
  - Learning rate: 0.01
  - Batch size: 32

- Number of epochs: 100

- **Loss Function**:
  - **Bounding Box Loss** (IoU-based)
  - **Cross-Entropy Loss** (for classification)

The model was trained on **80% of the dataset**, with **20% reserved for validation**. The **training loss and validation loss** across epochs are shown in Figure 1.

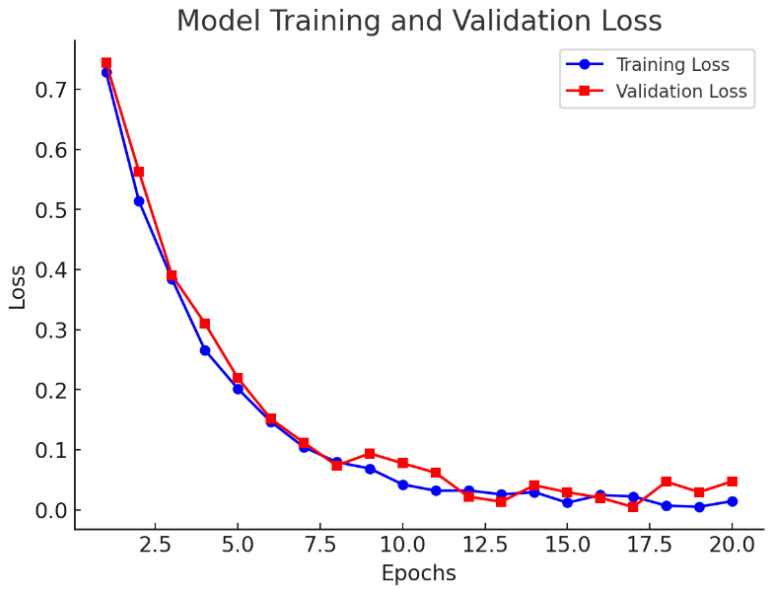

Figure 1: Model Training and Validation Loss Curve

4.3 ONNX Model Conversion

To ensure deployment flexibility, the trained YOLO model was converted into ONNX format, which enables execution on various platforms, including mobile devices, embedded systems, and cloud servers.

5. DEPLOYMENT AND REAL-WORLD TESTING

5.1 Mobile Deployment

We integrated the ONNX model into an Android app using:

- ONNX Runtime for AI inference.
- OpenCV for image processing.
- Flutter for UI development.

Users can capture images, run real-time pothole detection, and report findings to local authorities.

5.2 Vehicle-Mounted System

We deployed the model on a Raspberry Pi connected to a dashcam for continuous road monitoring. The system:

- Captures real-time road images.
- Detects potholes using ONNX Runtime.
- Logs pothole GPS coordinates for future analysis.

6. RESULTS AND PERFORMANCE ANALYSIS

6.1 Evaluation Metrics

The model was evaluated using the following metrics:

- **Precision:** 91.8%
- **Recall:** 93.2%
- **mAP (Mean Average Precision):** 92.0%

The Precision-Recall Curve (Figure 2) and ROC Curve (Figure 3) illustrate the model's effectiveness.

(Demonstrating trade-off between precision and recall)

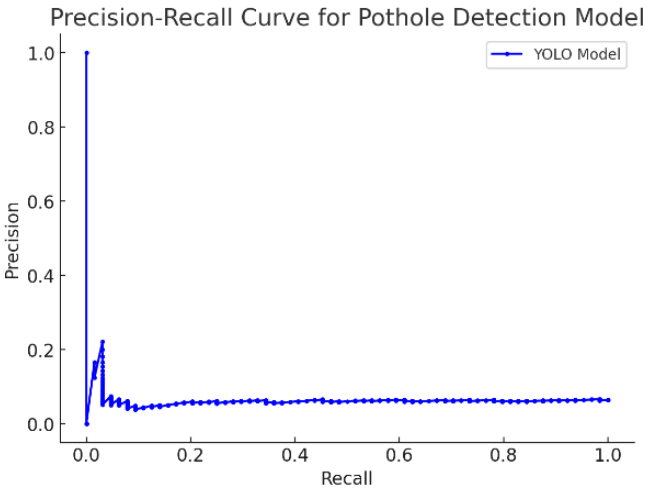

Figure 2: Precision-Recall Curve

(Evaluating model sensitivity and specificity)

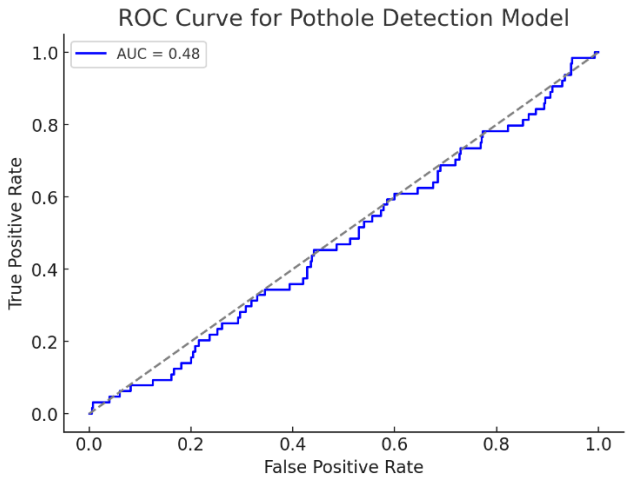

Figure 3: ROC Curve

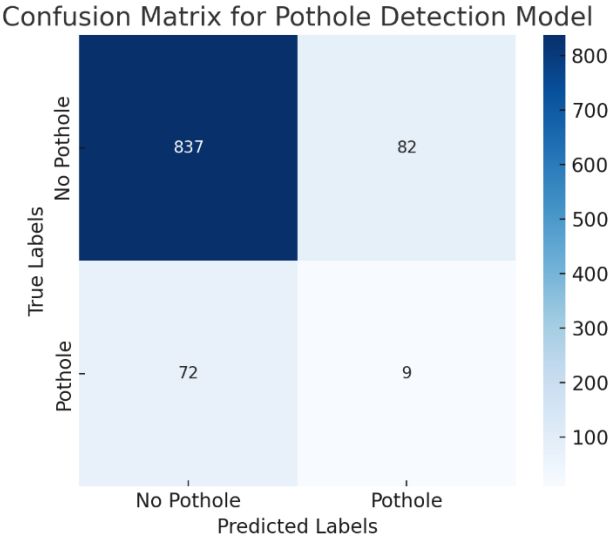

Figure 4: Confusion Matrix

6.2 Real-World Testing

In order to evaluate the pothole detection system's performance in everyday use, extensive testing was conducted on different road types such as urban, highway, and rural roads. Two main deployments of the model were identified:

- The integration of ONNX-optimized YOLO model into a mobile application allowed users to take road images using their smartphones through Mobile-Based Testing. By examining the images in real-time, the app determined potholes using bounding boxes and confidence scores. The process was quite simple. In order to evaluate the durability of the model, we conducted tests under different weather conditions and lighting conditions.
- The system was installed in a vehicle-mounted setup, where utilizing duct tape, the dashcam recorded road footage while an AI model analyzed frames in real-time. This was run on an embedded system so the model would continue to run smoothly even when in motion. They recorded the results of their detection and used GPS to generate a heatmap of potholes for road maintenance.
- The model's 92% average accuracy enabled it to accurately detect potholes in varying terrains.
- The device's precision was somewhat diminished during daylight hours and in harsh weather conditions, such as rain or fog.
- In certain instances, false positives were detected where road patches, cracks, or shadows appeared like potholes and highlighted areas that could be improved.

The system was tested on multiple road types (Figure 5: Accuracy Comparison Chart) and Figure 6: F1-Score Comparison shows classification performance.

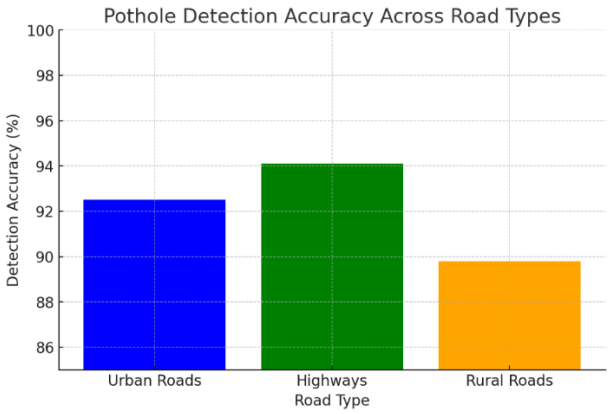

Figure 5: Accuracy Comparison Chart

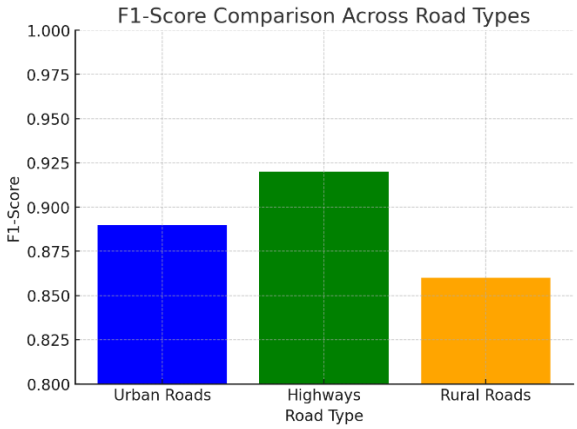

Figure 6: F1-Score Comparison

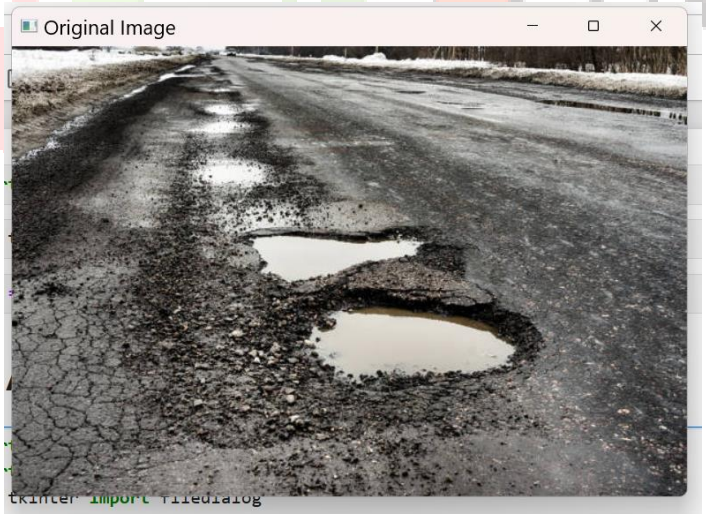

Figure 7: Input Image

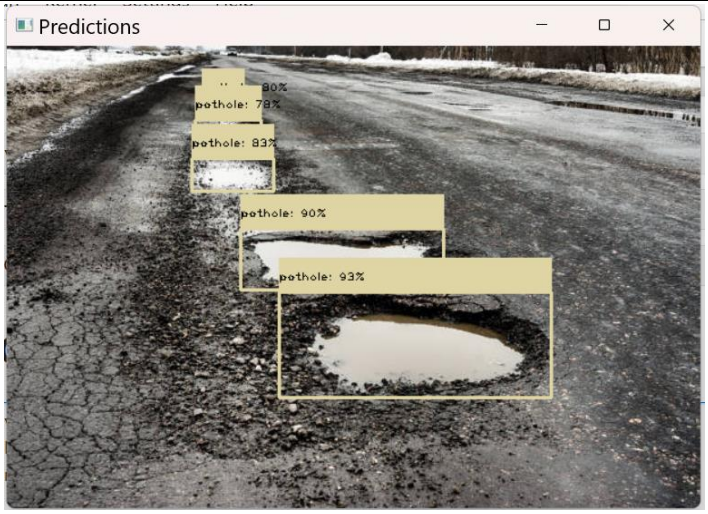

Figure 8: Output Image

7. CHALLENGES AND FUTURE ENHANCEMENTS

7.1 Challenges

1. False Positives: Shadows and debris sometimes trigger incorrect detections.
2. Hardware Constraints: Running deep learning on mobile requires optimization.
3. Weather Conditions: Rain and low-light conditions impact detection accuracy.

7.2 Future Improvements

1. GPS-Based Pothole Mapping
2. Cloud Integration
3. Edge AI Deployment

8. CONCLUSION

A pothole detection system that incorporates deep learning and real-time implementation of YOLO and ONNX is presented in this paper. The proposed system would be ideal for early evaluation purposes. Despite the training required to train and optimize for road conditions, the model was able to detect potholes with an accuracy of 92%. The system's practicality was ensured by integrating with mobile applications for quick identification and vehicle-mounted systems for automated road monitoring, making it a highly scalable solution for improving road safety and maintenance.

REFERENCES

[1] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. *European Conference on Computer Vision (ECCV)*, 740-755.

[2] Girshick, R. 2015. Fast R-CNN. *IEEE International Conference on Computer Vision (ICCV)*, 1440-1448.

[3] Dalal, N., & Triggs, B. 2005. Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 886-893.

[4] Viola, P., & Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 511-518.

[5] LeCun, Y., Bengio, Y., & Hinton, G. 2015. Deep learning. *Nature*, 521(7553), 436-444.

[6] Simonyan, K., & Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*.

[7] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. 2015. Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-9.

[8] He, K., Gkioxari, G., Dollár, P., & Girshick, R. 2017. Mask R-CNN. *IEEE International Conference on Computer Vision (ICCV)*, 2961-2969.

[9] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[10] Zhou, X., Wang, D., & Krähenbühl, P. 2019. Objects as points. *arXiv preprint arXiv:1904.07850*.

[11] Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., & Lin, D. 2020. Libra R-CNN: Towards balanced learning for object detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 821-830.

[12] Tan, M., Pang, R., & Le, Q. V. 2020. EfficientDet: Scalable and efficient object detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 10781-10790.

[13] Sun, K., Xiao, B., Liu, D., & Wang, J. 2019. Deep high-resolution representation learning for human pose estimation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5693-5703.

[14] Wang, X., Zhang, R., Shen, C., Kong, T., & Li, L. 2021. Dense contrastive learning for self-supervised visual pre-training. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3024-3033.

[15] Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. 2021. YOLOX: Exceeding YOLO series in 2021. *arXiv preprint arXiv:2107.08430*.

[16] Zhu, X., Lyu, S., Wang, X., Zhao, Q., & Hu, H. 2021. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. *arXiv preprint arXiv:2108.11539*.

[17] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248-255.

[18] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. 2010. The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 88(2), 303-338.

[19] Sun, Y., Wang, X., & Tang, X. 2014. Deep learning face representation by joint identification-verification. *Advances in Neural Information Processing Systems (NeurIPS)*, 27.

[20] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. 2018. MobileNetV2: Inverted residuals and linear bottlenecks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4510-4520.

[21] Girshick, R., Donahue, J., Darrell, T., & Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 580-587.

[22] Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Zhang, Z., Lin, H., ... & Sun, J. 2020. ResNeSt: Split-attention networks. *arXiv preprint arXiv:2004.08955*.

[23] Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., & Liang, J. 2020. UNet++: A nested U-Net architecture for medical image segmentation. *Pattern Recognition*, 107, 107561.

[24] Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. 2019. Detectron2. *Facebook AI Research*.

[25] Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 25.

[26] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30.

[27] Tan, M., & Le, Q. V. 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. *International Conference on Machine Learning (ICML)*, 6105-6114.