Pavement Distress Detection Using YOLO and Faster RCNN on Edge Devices

Chen-Kang Chiu[1], Jung-Chun Liu[1], Yu-Wei Chan[2], and Chao-Tung Yang[1,3]

[1] Department of Computer Science, Tunghai University, Taiwan
aganggggg@gmail.com, jcliu@thu.edu.tw, ctyang@thu.edu.tw,
[2] Department of Information Management, Providence University, Taiwan
ywchan@gm.pu.edu.tw
[3] Research Center for Smart Sustainable Circular Economy, Tunghai University, No. 1727, Sec. 4, Taiwan Boulevard, Xitun District, Taichung, 407224, Taiwan, ROC
ctyang@thu.edu.tw

Abstract. In this study, transfer learning techniques will be used for model training, using edge computing [1] and deep learning object detection technology, combined with image road pothole detection applications, and deploying devices and tools that accelerate neural network operations, including DeepStream [2] and Intel NCS2. The performance and accuracy of model recognition will be compared, and finally, real-time streaming video technology will be used to present the results on the web. According to the experimental results, the best model achieved an mAP of 70.% in YOLOv4-tiny-3l, and in terms of operating efficiency, deployment on Jetson Xavier NX using DeepStream for acceleration can achieve 30FPS. Finally, the deep learning model recognizes the screen presented on the web. This application can improve the accuracy of Pavement Distress identification and help road maintenance units improve the efficiency of repairing roads.

Keywords: Transfer Learning, Deep Learning, Edge Computing, Object Detection, DeepStream

1 Introduction

Traditional road survey work usually requires manual visual inspection and written records [3], but this method has some problems. Factors such as the pressure of driving on the road, rain, and high temperatures caused by sun exposure can cause various potholes in the road. Passing through potholes can reduce the service life of vehicle parts, increase vehicle maintenance rates, and seriously affect the safety of road users. We can develop more efficient and accurate image-processing methods for various application scenarios based on this approach [4].

On the other hand, You Only Look Once (YOLO) [5]belongs to the one-stage object detection algorithm, in which the entire architecture consists of only convolutional and fully connected layers. After inputting an image, the algorithm can quickly obtain the positions and categories of objects much faster than other methods like R-CNN [6]that require obtaining candidate regions before classification. Yang et al. [7] design a management system that uses drone-mounted cameras combined with object identification and live broadcast systems to assist disaster relief.

2 System Design and Implementation

2.1 System Architecture

In this study, we used Ubuntu 18.04 as the operating system for the research. We conducted deep learning and object detection experiments using Tensorflow and PyTorch, respectively, with Python as the primary programming language. we collected videos of pavement distress in Taiwan using a dashcam and used Python OpenCV to extract frames from the videos to augment the dataset. Various neural network architectures were constructed as different machine learning models for training, followed by model evaluation and performance comparison. The most suitable model was selected and deployed on different edge devices, utilizing acceleration tools to improve FPS, ensure stability, and reduce data transmission latency and network issues. The generated data was transmitted to the client-side through a web server to enable real-time viewing functionality. Figure 1 illustrates the complete system architecture.

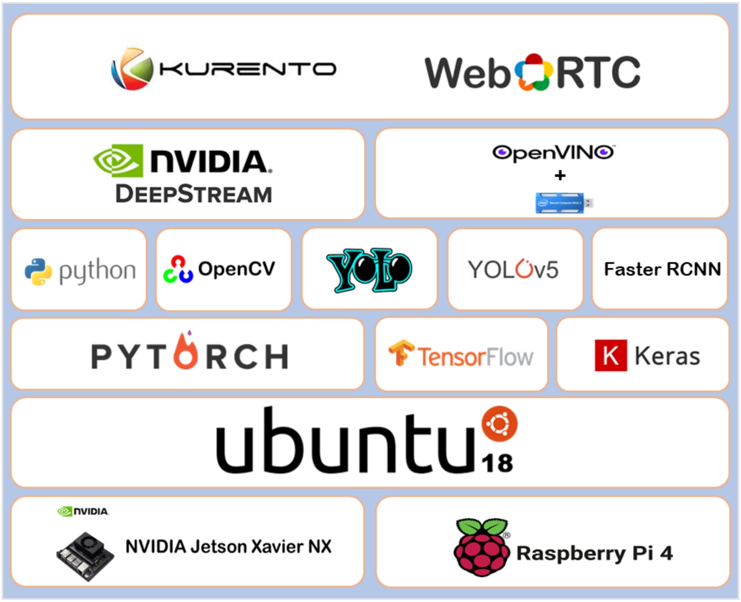


Fig. 1. System architecture diagram.

3 Type of Pavement Distress and Data Collection

3.1 Type of Road Damage

Considering that different countries and regions have varied types of road damage, this study takes into account the limited availability of self-labeled data and also refers to a pavement distress dataset from six countries in 2022: Japan, India, the Czech Republic, Norway, the United States, and China. [8] This dataset consists of 47,420 road images and categorizes pavement distress into four classes, as shown in Table 1.

Due to the limited availability of pavement distress data currently provided in Taiwan, it is not possible to segment the data and conduct training, we will utilize the pavement distress dataset from 2022 to conduct model training and performance testing.

Table 1. 2022 Pavement Distress Dataset

Damage type / Eng	Class Name
Longitudinal crack	D00
Transverse crack	D10
Alligator cracks	D20
Pothole	D40

3.2 Data collection

To train a deep learning model, a sufficient amount of dataset is required, and it takes time to train a good model. In this study, we used the pavement distress dataset provided by RDD2022 for our data. This dataset includes categories such as potholes, alligator cracks, transverse cracks, and longitudinal cracks. It was collected from roads in six different countries and exhibits excellent complexity. However, only the test set of this dataset is annotated. Since our experiments are conducted in Taiwan, we selected a dataset from Japan that is more representative of Taiwanese roads to ensure the accuracy of the model. The Japanese road dataset consists of a total of 16,470 images. Additionally, we collected 2,000 images of pavement distress in Taiwan through car recorders, resulting in a current total of 18,470 images. Figure 2 shows the pavement distress dataset.

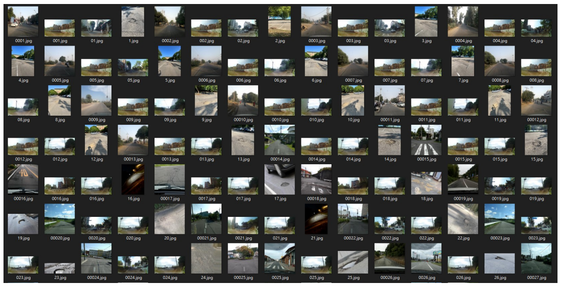


Fig. 2. Pavement Distress Dataset.

3.3 Data Augmentation

The training process of deep learning involves extracting meaningful image features from the training data, such as edges, colors, orientations, positions, and textures. However, since the collected data cannot cover all scenarios, this study used a total of 18,470 images as the training dataset. Data organization and augmentation were performed using the online data management platform, Roboflow. Each image was augmented every 30 degrees, and additional transformations like translation, flipping, and distortion were applied to account for rotational variations in the shooting angles. This expanded the dataset to a total of 6,867 images, resulting in a final dataset of approximately 25,337 images. Furthermore, all images were resized to a dimension of 416x416. In terms of data distribution, the datasets were divided into 60% for training, 20% for validation, and 20% for testing. Figure 3 shows the After data augmentation.

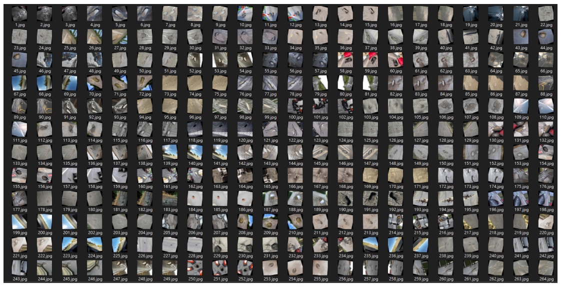


Fig. 3. After data augmentation.

3.4 Feature Label and CSV File Generation

In this study, LabelImg [9] will be used for feature labeling. This tool is open-source software that allows setting the locations of the original and storage files and annotating specific positions and classes of images. The generated XML files provide detailed information about the labeled images, including the image name, image type, and x, y position values of the potholes, among other relevant information. Figure 4 display the content of the CSV files, and Figure 5 illustrates the number of labels for each category.

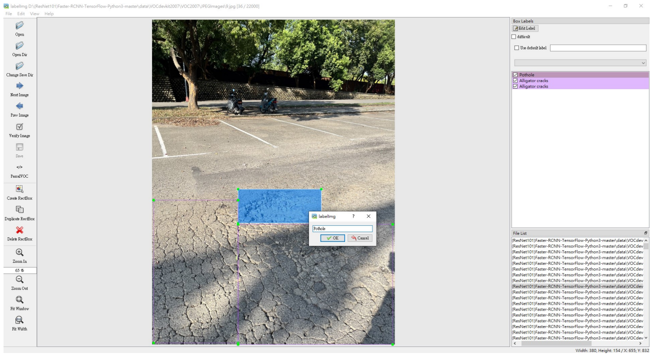


Fig. 4. Data annotation.

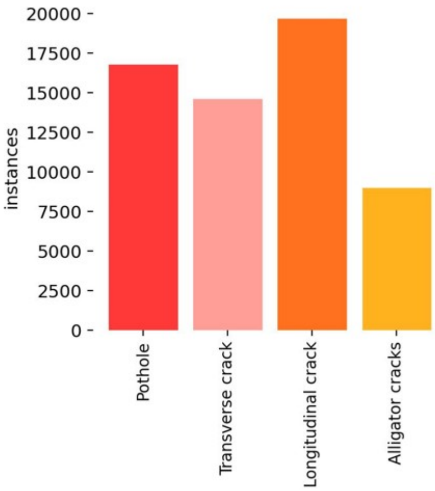


Fig. 5. amount of annotation

4 Pavement Distress Model Training Results

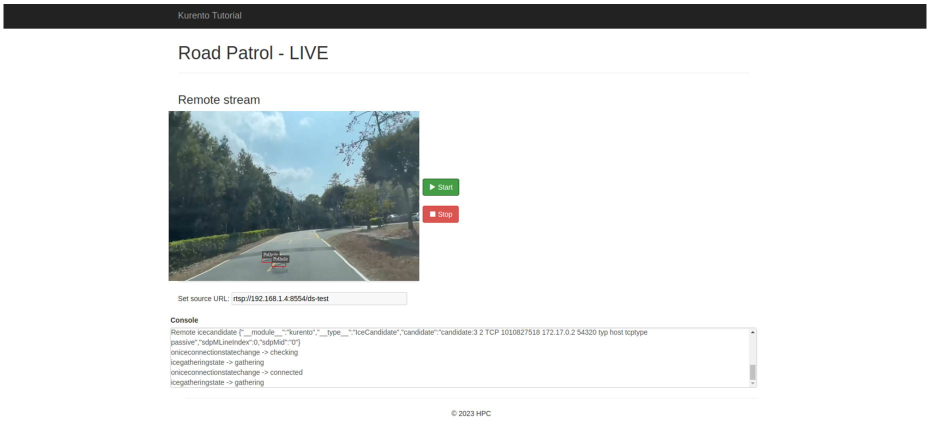
According to the comparison results in Table 2, YOLOv4-tiny-3l is more suitable for road pothole detection. This study trained models using six different neural network architectures. Since the images were resized to a size of 416×416 during the data augmentation process, all six models were trained with an input size of 416×416. It can be observed that YOLOv4-tiny-3l achieved the highest Precision at 75% while having a relatively lower Recall of 63%. This indicates that the models have conservative predictions, resulting in high Precision but lower Recall. On the other hand, YOLOv5m generated higher Recall but led to a decrease in Precision. This implies that if a model is too greedy and aims to predict more positive examples as Ground Truth, it may result in false positives. In such a trade-off situation, another metric, F1-score, which is the harmonic mean of Precision and Recall, is derived. It can be seen that YOLOv4-tiny-3l, YOLOv5s, and YOLOv5m all achieved the same F1-score of 68%. Considering the same value, further comparison of the models is needed. Therefore, this study adopted mAP0.5 as the final evaluation metric. The results showed that YOLOv4-tiny-3l achieved the highest mAP0.5 at 70.5% among the six models, while YOLOv5s and YOLOv5m achieved 68.4% and 69% respectively. On the contrary, due to the complex architectures of VGG16 and ResNet101, they required more training time and exhibited relatively poorer performance in terms of evaluation metrics compared to YOLOv4-tiny-3l. Meanwhile, YOLOv4-tiny has only two yolo layers, reducing the computational complexity and model size compared to YOLOv4-tiny-3l.

Table 2. Model prediction result table

	F1-score	**mAP0.5**	**Precision**	**Recall**	**Training Time**
VGG16	66%	66.5%	65%	66%	168H
ResNet101	66%	67.5%	65%	64%	192H
YOLOv4-tiny	65%	66.0%	66%	65%	36H
YOLOv4-tiny-3l	68%	70.5%	75%	63%	48H
YOLOv5s	68%	68.4%	67%	66%	26H
YOLOv5m	68%	69.0%	65%	68%	46H

5 Web Page Presentation

In terms of real-time streaming, this study utilized Kurento and WebRTC [10] technologies to establish a web application for live video streaming. To enable communication between WebRTC and IP cameras, their media formats must be compatible. This encoding format conversion task is handled by a WebRTC gateway such as Kurento. The web application allows direct viewing in the browser, while the server side is capable of performing video streaming recognition. In addition to providing the edge device page, the web application also offers a page for real-time video processing on the server side, supporting synchronized real-time streaming. Figure 6 showcases the interface of the real-time pavement distress recognition web application.



Set source URL: rtsp://192.168.1.4:8554/ds-test

Console

```
Remote icecandidate {"__module__":"kurento","__type__":"IceCandidate","candidate":"candidate:3 2 TCP 1010827518 172.17.0.2 54320 typ host tcptype
passive","sdpMLineIndex":0,"sdpMid":"0"}
oniceconnectionstatechange -> checking
icegatheringstate -> gathering
oniceconnectionstatechange -> connected
icegatheringstate -> gathering
```

© 2023 HPC

Fig. 6. Web results screen

References

1. Loc N. Huynh, Youngki Lee, and Rajesh Krishna Balan. Deepmon: Mobile gpu-based deep learning framework for continuous vision applications. MobiSys '17, page 82–95, New York, NY, USA, 2017. Association for Computing Machinery.
2. Nuha H. Abdulghafoor and Hadeel N. Abdullah. A novel real-time multiple objects detection and tracking framework for different challenges. Alexandria Engineering Journal, 61(12):9637–9647, 2022.
3. Ying-Ju Lai. Analysis of forward-looking plans improving the road quality (highway system) - first maintenance office, directorate general of highways, motc. http://ir.lib.ncu.edu.tw:88/ thesis/ viewetd.asp?URN = 107352005, 2020.
4. Yih–Chen Wang, Chao–Wei Yu, Xiu–Ying Lu, and Yen–Lin Chen. Road semantic segmentation and traffic object detection model based on encoderdecoder cnn architecture. In 2022 IEEE International Conference on Consumer Electronics - Taiwan, pages 421–422, 2022
5. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.
6. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014
7. Yang, CT., Lin, WY., Chen, YC., Wang, ZY., Lee, CH. (2021). Flame Recognition System Using YoLo. In: Chang, JW., Yen, N., Hung, J.C. (eds) Frontier Computing. FC 2020. Lecture Notes in Electrical Engineering, vol 747. Springer, Singapore.
8. Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, Hiroshi Omata, Takehiro Kashiyama, and Yoshihide Sekimoto. Crowdsensingbased road damage detection challenge (crddc-2022), 2022.
9. labelimg. https://github.com/heartexlabs/labelImg.
10. Webrtc architecture. https://webrtc.org/.