2022 4th International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)
29-31 December 2022, Rajshahi-6204, Bangladesh

2022 4th International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE) | 979-8-3503-2054-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICECTE57896.2022.10114508

Road Damage Detection and Classification using Deep Neural Network

Md. Mahadi Hasan
Dept. of CSE,
CUET
Chattogram, Bangladesh
ORCID:0000-0003-4982-1689

Saadman Sakib
Dept. of CSE,
CUET
Chattogram, Bangladesh
ORCID:0000-0002-4213-8169

Kaushik Deb
Dept.of CSE,
CUET
Chattogram , Bangladesh
ORCID:0000-0002-7345-0999

***Abstract*—Road damage detection and classification is an important sector in computer vision and image processing. Road damage detection is rapidly getting a high application in various sectors of computer vision, especially in self-driving cars; it is getting great attention. We used the RDD-2020 dataset with four classes for training our models. Most of the existing methods using the RDD-2020 dataset failed to detect multiple overlapping damage regions of different classes in the same image. To overcome these limitations and enhance the previous model's performance, we experimented with five different pre-trained models. We fine-tuned the best-performing model (Faster RCNN with ResNet-101) by using three different optimizers (Adam, RMS Prop., and SGD with momentum) and kept batch sizes as 4, 8, 16, and 32. Then, we proposed a fine-tuned Faster RCNN with the ResNet-50 model. The experimented pre-trained models are EfficientDet, SSD with MobileNet-v1, SSD with MobileNetv2, SSD with ResNet-50, and Faster RCNN with ResNet-101. These models have a backbone network and a detecting head. The backbone network extracts the feature maps, and the detection head generates bounding box coordinates for detecting the damaged region and classifies these bounding boxes. We have found a maximum f1 score of 0.47 for the longitudinal class, 0.41 for transverse class, 0.41 for alligator class, and 0.35 for pothole using Fine-tuned Faster RCNN with ResNet-101 by keeping the batch size as 4 and optimizer as SGD with momentum which are better than the previous model of literature using RDD-2020 dataset.**

***Index Terms*—Road damage detection, Faster RCNN, SSD, EfficientDet, ResNet-101, MobileNet, EfficientNet, Region Proposal Network, ROI Pooling.**

I. Introduction

The primary form of public transportation is the city road system. Every day several accidents occur on the roads. According to the World Health Organization's 2021 report, vehicle accidents in Bangladesh account for 16.5% of all fatalities [1]. The most crucial task in resolving such a bad scenario is to repair roads to protect human life. The condition of roads needs to be frequently inspected. The truth is that maintaining a road's surface takes a lot of time and money. A low-cost automated method is required to identify road damages. The number of automated vehicles is rising steadily in the modern day. Therefore, autonomous vehicles require automated road surface damage detection. Therefore, identifying road damage in the present era is a challenging task. In urban areas, checking road conditions is typically done using human resources, a labor-intensive operation that takes much time. In our approach, we propose to explore the object detection approach for detecting road damages and then classifying them from road surface images. Various object detection algorithms, such as in the RCNN family Faster R-CNN is one of the best states of the art two stages of the detection algorithm. On the other hand, SSD (Single Shot Detection) is accessible now, one of the best state-of-the-art single-stage algorithms. They operate as the model's detection head. Here, We use RDD(Road Damage Detection)-2020 dataset. As the feature extractor, we employed four transfer learning models (MobileNet-v1, MobileNet-v2, EfficientNet, ResNet-101) and three detection algorithms (Faster RCNN, SSD, EfficientDet) as the detection head. We found the best F1 score using Faster RCNN with ResNet-101 for all four classes. These four classes are longitudinal crack or D00, transverse crack or D10, alligator crack or D20, and pothole or D40.

II. Literature Review

Over the past few years, researchers have focused on road damage detection and classification. In [2], the authors proposed two models for road damage detection using SSD with MobileNet v2 and SSD with inception v2. They have used RDD 2020 datasets which contain 12,195 images with 25,046 road surface image labels of Japan, India, and the Czech Republic. The limitation of the paper was that, in some cases, the model gave unsatisfactory results in detecting overlapping multiple damaged regions. They considered four classes. In [3], the authors suggested a Faster R-CNN-based model for categorizing and detecting road damage. Their

feature extractors were ResNet-50 and ResNet-101. They used RDD 2018 dataset, which contains 9,053 images and 15,435 bounding boxes collected from Japan using a smartphone. At a 50 % IoU, they have attained a mean F1 score of 0.528. They used the Region proposal network to generate region proposals in damaged regions by bounding boxes. They trained their model with a 0.01 learning rate. They used adam optimizer and kept batch size at 4. In [4], the authors proposed a framework for road damage detection using SSD as the detection head and using two transfer learning models, MobileNet-v2 and Inception-v2. They used RDD 2018 dataset. They were able to attain precision and recall greater than 62%. They created the smartphone application also. The limitation of this paper and paper [3] was that their model gave an unsatisfactory result in detecting multiple overlapping damage regions in the same image. They found better fps results; however, in terms of accuracy, they did not get results as Faster R-CNN produced. In [5], the authors proposed a road damage detection model using RCNN and Faster R-CNN. They collected 1100 images from Dhaka city. They have considered three classes pothole, rivaling, and clack. They employed an Adam optimizer and trained with 50 epochs, validating their model on 200 images. They found a precision of 99%, recall 97%, and a maximum F1 score of 98% using faster R-CNN in their 200 test images. The main limitation of their paper was that they used their model only for classification. No detection techniques were applied there. They did not generate any bounding boxes. In [6], the authors used an improved Mask R-CNN algorithm for the localization of damages. They used ResNet-50 with FPN (Feature Pyramid Network) for feature extraction and Mask R-CNN for detection purposes. Additionally, they modified the skip connection module of ResNet-50 and got a better result. A total of 2,000 damaged images were gathered for the experiment through web downloads and regular photos. Initially, they kept the learning rate at 0.001, learning momentum at 0.9, IoU at 0.8, and epochs at 100, and they considered 2 classes. Their paper's limitation was that they used a small dataset, and their model gave unsatisfactory results in detecting damages from low light. In [7], the authors used different learning rates as 0.001, 0.002, and 0.003 while training their model and got better results at 0.003. They used Mask R-CNN to generate masks for damaged regions. They used 800 images for their model. They considered two classes, compared their model with other different transfer learning models, and found better results in their model. In [8], the authors used ProposeNet, the modified version of DenseNet-121, for feature extraction and used Mask R-CNN as semantic segmentation for generating masks in the damaged region. In [9], the authors used an earthquake-based dataset to detect and classify damage. They used ResNet-50 as a feature extractor and faster RCNN to generate bounding boxes on the damaged region. In [10], the authors used an earthquake-based dataset to detect and classify damage. They have used MobileNet v2 as a feature extractor and Faster RCNN as a detection head to generate bounding boxes. In [11], the authors made a survey paper about Global Road Damage Detection Challenge 2020, in which 121 teams participated in this competition. They summarized the top 12 team's F1 score and their working principle. In [12], the author of the previous paper [2] modified SSD with the ResNet-50 model by fine-tuning. They found slightly better f1-score than the previous one. They trained their model using the RDD-2020 dataset.

III. Methodology

Our proposed road damage detection and classification method combine pre-trained transfer learning models and a detection head. We used four different transfer learning models and three detection heads. We found a maximum F1 score using Faster RCNN with ResNet-101. For classification, the softmax activation function produces the probability distribution of the damage classes.

A. Overview of Our Proposed Model

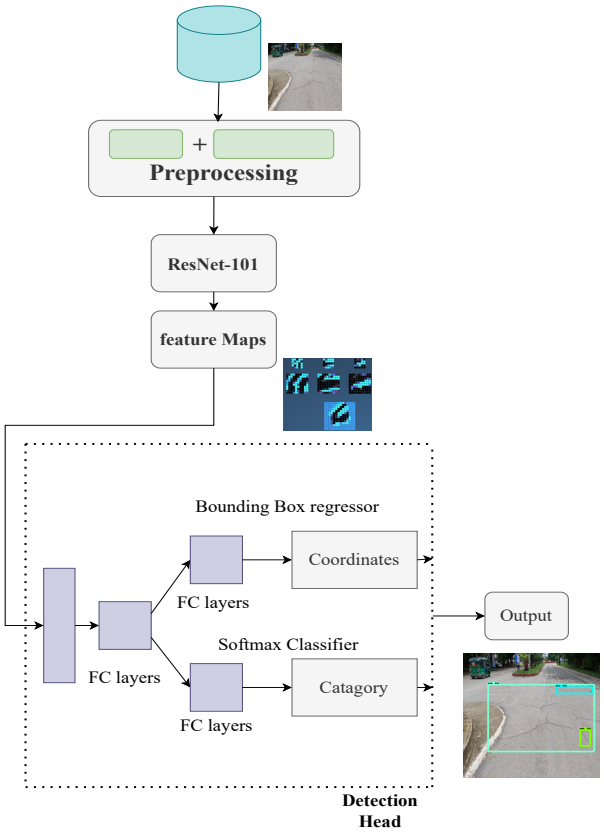


Fig. 1: Methodology of our proposed model.

Figure 1 displays the proposed road damage detection and classification method. We resized the images from the RDD-2020 dataset to reduce the computational cost, and we used four different transfer learning models. Different transfer learning model takes input image of different sizes. We normalized the data to converge faster training speed. Moreover, we used min-max normalization for image preprocessing. Then we convert our annotated XML files into CSV files using the element tree. Then the CSV files are converted into TfRecords using UTF-8 (Unicode Transformation Format) as

we are using TensorFlow Object detection API for detecting the damaged region. After preprocessing, these images are taken by ResNet-101. The task of ResNet-101 is to generate feature maps. We used four different transfer learning models and found the best f1-score using ResNet-101. We used three different detections dead and found the best f1-score using Faster RCNN. The standard part of a detection head is that there is a fully connected network where our model learns using these feature maps, which are necessary for road damage detection. Our models automatically select these feature maps as we work with label data. The output layer of the Fully Connected layer contains two layers. One is the bounding box regressor layer, and another is the softmax classification layer. The task of the bounding box regressor layer is to generate four coordinate values (X-min, Y-min, X-max, and Y-max). The task of the softmax classification layer is to generate the probability value of each class. In the bounding box regressor layer, there are four output neurons. Here, we used the sigmoid activation function in the output layer and the Relu activation function in the hidden layer. We used MSE (Mean Square Error) as a loss function here. In the softmax classification layer, we used the softmax activation function and used Sparse Categorical Cross Entropy as a loss function. Finally, we got the output result with bounding boxes for detecting the damaged region and the corresponding class and confidence score.

B. Detailed Explanation of Our Proposed Methodology

The explanation of each step conducted in the entire process of road damage detection and classification is represented in the following subsection.

1) Data Preprocessing: In RDD 2020 dataset, there are 25,046 resized images, and they are annotated into XML format. The resolution of these images was high. So, we resized these images. As different transfer learning backbone networks take the different sizes of the input image and to reduce the computational cost, we need to resize these images. We normalized our data to reduce training time. We have used min-max normalization, which ranges the pixel values from 0 to 1. The equation of min max normalization is shown in equation 1.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

We also converted annotation XML files into CSV files and then converted these CSV files into Tfrecords.

2) Convolutional neural network (CNN)- extracts spatial features: We evaluated different deep-learning architectures to extract spatial features. Different backbone transfer learning models like MobileNet-v1, MobileNet-v2, ResNet-101, and EfficientNet was used in this study. These architectures have been selected because they have been used widely to extract features from images. These architectures contain fully-connected, pooling, and convolutional layers that generate feature vectors. The depth of ResNet-101 layers (101 convolutional layers & 3 fully connected layers) whereas MobileNet-v1 & MobileNet-v2 with 28 & 53 deep layers respectively. The key feature of MobileNet is that this transfer learning model uses depthwise separable convolution followed by pointwise convolution. The input of the images for our case was set to 224x224. We found a maximum f1-score using ResNet-101. The key feature of ResNet-101 is that there is a skip connection to reduce the vanishing gradient problem. Figure 2 shows the architecture of the ResNet-101 model. ResNet-101 solves the vanishing gradient problem by using a skip connection shown in figure 3.

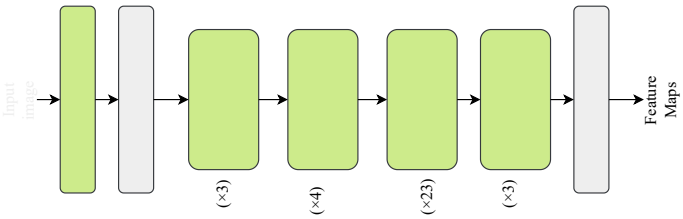


Fig. 2: Overview architecture of the CNN model (ResNet-101).

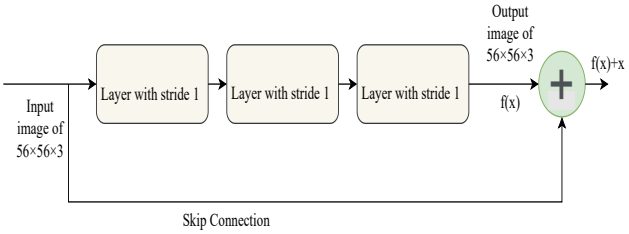


Fig. 3: Skip connection of the ResNet 101 CNN architecture.

3) Detection head: The detection head takes feature maps. The task of the detection head is to detect damaged regions by bounding boxes. These boxes are generated in different ways based on which detection algorithm is used. The detection head can be one stage or two-stage algorithm. We used SSD as a single-stage detection algorithm and used Faster RCNN as a two-stage detection algorithm. We found a maximum f1-score using Faster RCNN, which is a two-stage detection algorithm. Two-stage detection algorithms like Faster RCNN use Region Proposal Network, which task is to generate region proposals where the object might present. Single-stage detection algorithms like SSD generate 8732 anchor boxes of different sizes, scales, and aspect ratios and then select only 200 anchor boxes, calculating IoU value and using Non-Max suppression. In our proposed method, we have used three detection algorithms, i.e., Faster RCNN, SSD, and EfficientDet. We found a maximum f1-score using Faster RCNN. Figure 4 shows the detection and classification process using Faster RCNN. Pre-trained transfer learning model ResNet-101 generate feature maps of different size, and the region proposal network generates regions. First, the Region Proposal network generates 9 anchor boxes for each pixel of different accept ratios (1:1, 1:2, and 2:1). Then, these anchor boxes calculate the IoU value with the overlapped portion of ground truth

boxes. If the IoU value is greater than the IoU threshold, this anchor box is considered the foreground class. We considered IOU threshed value as 0.6, which means more than 60% overlapping anchor boxes will be considered foreground class. These foreground anchor boxes are the regions of RPN. The ROI pooling layer takes these different size regions and feature maps. The task of the ROI pooling layer is to convert these different size feature maps and regions into a fixed size. Finally, these fixed feature maps and regions are taken by a Fully Connected layer and generate bounding boxes and damage classes.

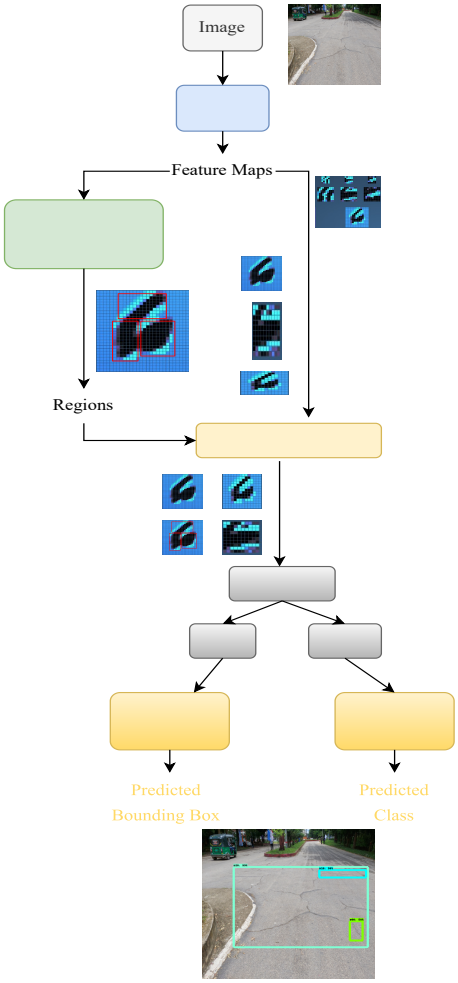


Fig. 4: Faster RCNN architecture.

IV. Result and Discussion

We conducted a series of experiments contrasting our approaches with others to assess our proposed method's efficacy.

A. Dataset Description

We used the RDD-2020 dataset to train and evaluate our proposed models. Road Damage Dataset-2020 (RDD-2020) is a sizable heterogeneous dataset of 12,195 images with 25,046 road surface image labels that were gathered using smartphones in various nations. The pictures were taken on Czech, Japanese, and Indian roadways. This dataset contains four classes. They are Longitudinal crack named D00, Transverse crack named D10, Alligator Crack named D20, and Pothole named D40. Figure 5 represents sample images of the RDD-2020 dataset. The 1st row represents sample images from Czech, the 2nd row represents sample images from Japan, and the 3rd row represents sample images from India.

Fig. 5: Sample images in RDD-2020 dataset (Row1 represents Czech images, row2 represents Japanese road images, and row3 represents Indian road images).

B. Experimental Result

We used precision, recall, and f1-score matrices to evaluate our models. The equation of precision, recall, and f1-score is shown in equation 2, 3, and 4 respectively:

$$\texttt{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\texttt{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\texttt{F1-score} = \frac{2 * precision * Recall}{Precision + Recall} \tag{4}$$

True Positive (TP) means the actual value was positive, and the model predicted a positive value. True negative (TN) means the actual value was negative, and the model predicted a negative value. False positive (FP) means the actual value is negative, but the model predicted a positive value known as the Type 1 error. False Negative means the actual value was positive, but the model predicted a negative value which is known as the Type 2 error. Precision tells us how many correctly predicted cases turned out to be positive. Recall tells

us how many of the actual positive cases we could predict correctly with our model. F1-score is a harmonic mean of Precision and Recall, giving a combined idea about these two metrics. It is maximum when Precision is equal to Recall. Different pre-trained models like ResNet101, MobileNet-v1, MobileNet-v2, and EfficientNet have been used to exploit the learning of spatial information. We used three detection heads also. They are faster RCNN, SSD, and EfficientDet. The network was trained with a batch size of 4, 8, 16, and 32. we trained with three different optimizers (Adam, SGD with momentum, and RMS Prop) with a learning rate set to 0.001 for adam and 0.01 for SGD with momentum and RMS Prop. We have found the best F1 score keeping the batch size as 4 and the optimizer as SGD with momentum. We found the best f1-score using Faster RCNN with ResNet-101.The confusion matrix of Faster RCNN with ResNet-101 using the RDD-2020 dataset is shown in figure 6. We have four classes. So, a 4×4 confusion matrix is formed. The diagonal axis represents the true positive. Here total true positive labels are (152+47+142+75)=416.

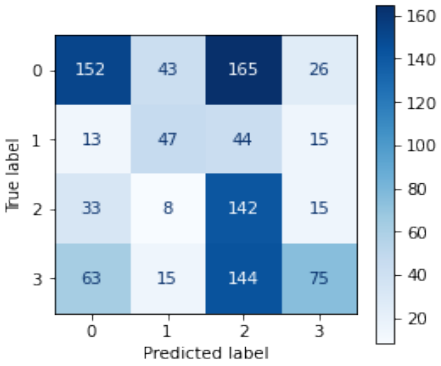


Fig. 6: Confusion matrix of our proposed method

TABLE I: Performance score of the RDD-2020 dataset on our five different models

Model Name	Class	Precision	Recall	F1 Score
Faster RCNN with ResNet-101	D00	0.58	0.39	0.47
	D10	0.42	0.39	0.41
	D20	0.29	0.72	0.41
	D40	057	0.25	0.35
SSD with MobileNet-v1	D00	0.5	0.19	0.22
	D10	0.49	0.17	0.26
	D20	0.2	0.64	0.36
	D40	0.1	0.05	0.07
SSD with MobileNet-v2	D00	0.44	0.44	0.44
	D10	0.04	0.01	0.01
	D20	0.32	0.74	0.45
	D40	0.15	0.07	0.1
SSD with ResNet-50	D00	0.23	0.12	0.15
	D10	0.00	0.00	0.00
	D20	0.22	0.79	0.35
	D40	0.12	0.04	0.08
EfficientDet	D00	0.63	0.09	0.15
	D10	0.26	0.18	0.21
	D20	0.2	0.86	0.33
	D40	0.5	0.03	0.06

The performance score of the RDD-2020 dataset for five different models are shown in table I. From the comparison, it can be viewed that Faster RCNN with ResNet-101 performs better than the other four different models.

The f1-score of the RDD-2020 dataset for different batch sizes for Faster RCNN with ResNet-101 is shown in table II. We found the best f1-score by keeping the batch size at 4.

TABLE II: F1-Score of the RDD-2020 dataset on different batch sizes

Model Name	Batch Size	D00	D10	D20	D40
Faster RCNN with ResNet-101	4	0.47	0.41	0.41	0.35
	8	0.42	0.41	0.39	0.34
	16	0.45	0.40	0.33	0.32
	32	0.43	0.39	0.31	0.31

The f1-score of the RDD-2020 dataset for model Faster RNN with ResNet-101 for different optimizers is shown in table III. We used three different optimizers for evaluating the RDD-2020 dataset. We found the best f1-score using SGD with momentum as an optimizer in Faster RCNN with ResNet-101.

TABLE III: F1-score of the RDD-2020 dataset on different optimizers

Optimizer	D00	D10	D20	D40
SGD with Momentum	0.47	0.41	0.41	0.35
RMS Prop.	0.22	0.26	0.07	0.31
Adam	0.47	0.40	0.33	0.32

We found a low f1-score for all four classes as in our dataset no of images per damage class is not same. As we are using imbalanced dataset our model performance is not high enough but we found better F1 score than the previous model using RDD-2020 dataset. No of images of test data shown in figure 7.

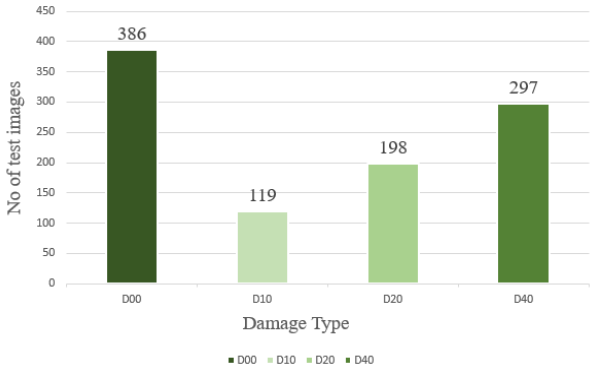
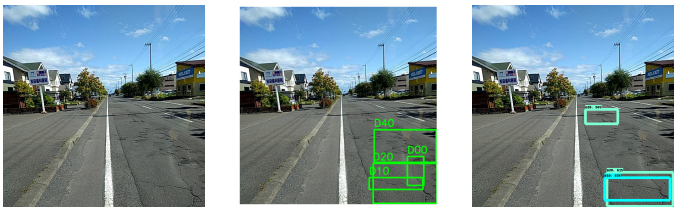


Fig. 7: No of images per class in the test data

Figure 8 shows the actual ground truth boxes and our predicted bounding boxes. The first image is collected from Japan, the second image is collected from India, and the third image is collected from the Czech Republic. Here four different colors represent four different damages. White color represents pothole class, lime color represents longitudinal

class, aqua color represents alligator crack, and dark aqua color represents the transverse class. The left column shows the actual ground truth boxes, and the right column shows the predicted bounding boxes. Actual ground truth boxes are labeled with the same green color, and class is given on the top portion of the bounding box. The first row represents the actual and predicted bounding boxes of images collected from Japan; the second row represents the Indian image's ground truth and the bounding boxes. Similarly, the third row shows the actual and predicted bounding boxes, an image collected from the Czech Republic.



(a) Input image (b) Ground truth (c) Predicted boxes

(d) Input image (e) Ground truth (f) Predicted boxes

Fig. 8: Input image, actual bounding boxes and predicted bounding boxes of RDD-2020 dataset.

A comparison of our proposed fine-tuned Faster RCNN with ResNet-50 and existing literature using RDD-2020 dataset is shown in table IV. In [2], they used SSD with MobileNet-v2. and in [12], same author of previous paper fine-tuned their model. We have found a maximum f1 score better than their models in the RDD-2020 dataset.

TABLE IV: Comparison of the F1 score of four classes with existing works of literature using the RDD-2020 dataset

Model Name	Class Label			
	D00	D10	D20	D40
Single Shot Detector model [2]	0.37	0.22	0.40	0.17
Fine-tuned Single Shot Detector model [12]	0.361	0.228	0.4094	0.334
Fine-tuned Faster RCNN with ResNet-101 (Our Proposed)	0.47	0.41	0.41	0.3

First, we experimented with five different pre-trained transfer learning models and then fine-tuned our best-performing pre-trained model by using three different optimizers and keeping the batch size as 4, 8, 16, and 32. We found a slightly better F1 score using SGD with momentum optimizer and by keeping the batch size as 4. Finally, we proposed a fine-tuned Faster RCNN with the ResNet model which performs better than the previous models using the RDD-2020 dataset.

V. Conclusion

Road damage detection systems based on deep learning can help with low-cost road maintenance. With the goal that it may aid future research in this area, we compared five deep neural algorithms in this thesis to see which one performed better at detecting road damage. Most existing methods using the RDD-2020 dataset failed to detect multiple overlapping damage regions of different classes in the same image. Our proposed method overcomes this limitation. We used four backbone transfer learning models and three different detection heads in these models. The best f1-score is obtained using Faster RCNN with ResNet-101.We trained our model using three optimizers and batch sizes 4, 8, 16, and 32 in Faster RCNN with ResNet-101. While training with different optimizers and batch sizes, we have found the best f1-score by keeping the batch size as 4 and the optimizer as SGD with momentum. Collecting images from Bangladesh road surfaces and training with this dataset can improve accuracy. Additionally, the work can be applied to the transportation system. Extend parameters to forecast the cost of fixing road damage so that the location of the most urgent repair needs can be identified.

References

[1] "The road traffic death rate." [Online]. Available: https://www.who.int/data/gho/data/themes/road-safety

[2] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, and Y. Sekimoto, "Transfer learning-based road damage detection for multiple countries," *arXiv preprint arXiv:2008.13101*, 2020.

[3] J. Singh and S. Shekhar, "Road damage detection and classification in smartphone captured images using faster R-CNN," *CoRR*, vol. abs/1811.04535, 2018. [Online]. Available: http://arxiv.org/abs/1811.04535

[4] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection using deep neural networks with images captured through a smartphone," *CoRR*, vol. abs/1801.09454, 2018. [Online]. Available: http://arxiv.org/abs/1801.09454

[5] M. S. Arman, M. M. Hasan, F. Sadia, A. K. Shakir, K. Sarker, and F. A. Himu, "Detection and classification of road damage using r-cnn and faster r-cnn: A deep learning approach," in *Cyber Security and Computer Science*, T. Bhuiyan, M. M. Rahman, and M. A. Ali, Eds. Cham: Springer International Publishing, 2020, pp. 730–741.

[6] Q. Zhang, X. Chang, and S. B. Bian, "Road-damage-detection segmentation algorithm based on improved mask rcnn," *IEEE Access*, vol. 8, pp. 6997–7004, 2020.

[7] P. Kumar, A. Sharma, and S. R. Kota, "Automatic multiclass instance segmentation of road damage using deep learning model," *IEEE Access*, vol. 9, pp. 90 330–90 345, 2021.

[8] S. Shim and G.-C. Cho, "Lightweight semantic segmentation for road-surface damage detection based on multiscale learning," *IEEE Access*, vol. 8, pp. 102 680–102 690, 2020.

[9] S. Karimzadeh, M. Ghasemi, M. Matsuoka, K. Yagi, and A. C. Zulfikar, "A deep learning model for road damage detection after an earthquake based on synthetic aperture radar (sar) and field datasets," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 5753–5765, 2022.

[10] K. Zhao, J. Liu, Q. Wang, X. Wu, and J. Tu, "Road damage detection from post-disaster high-resolution remote sensing images based on tld framework," *IEEE Access*, vol. 10, pp. 43 552–43 561, 2022.

[11] D. Arya, H. Maeda, S. Kumar Ghosh, D. Toshniwal, H. Omata, T. Kashiyama, and Y. Sekimoto, "Global road damage detection: State-of-the-art solutions," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 5533–5539.

[12] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, and Y. Sekimoto, "Deep learning-based road damage detection and classification for multiple countries," *Automation in Construction*, vol. 132, p. 103935, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580521003861