

Real-Time Road Damage Detection Based on Improved YOLOv5

Abdullah As Sami¹, Saadman Sakib², Kaushik Deb^{3*}

¹Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chattogram Bangladesh.

²Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chattogram, Bangladesh.

³*Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chattogram, Bangladesh.

*Corresponding author(s). E-mail(s): debkaushik99@cuet.ac.bd;
Contributing authors: abdullahassami@gmail.com; saadman@cuet.ac.bd;

Abstract

Deep learning has enabled a straightforward, convenient method of road infrastructure administration that facilitates a secure, cost-effective, and efficient transportation network. Manual road inspection is time-consuming and dangerous, making timely road repair difficult. This research showcases YOLOv5, the most commonly employed object detection model trained on the latest benchmark Road Damage Dataset, Road Damage Detection 2022 (RDD 2022). Limited research has been conducted on the effectiveness of the advanced RDD 2022, which encompasses the road damage of six nations: Japan, Norway, the United States, the Czech Republic, and China. The RDD 2022 includes four common types of road damage, namely vertical cracks, horizontal cracks, alligator cracks, and potholes. This paper presents a deep neural network model for road damage detection in photographic representations of road surfaces. The model is based on the YOLOv5 object detection framework, and it has been modified to incorporate several techniques that improve its accuracy and generalization performance. These techniques include the efficient Channel Attention module (ECA-Net), label smoothing, the K-means++ algorithm, Focal loss, and an additional prediction layer. In addition, our model achieved a 1.9% improvement in terms of mAP and 1.29% in terms of F1-Score from Yolov5s, while increasing the 1.1 million parameters. Moreover, our model achieved a 0.11% improvement in mAP and 0.05% in F1 score from YOLOv8s while containing 3 million fewer parameters and 12 Gigabytes fewer GFlops.

Keywords: road damage detection, deep neural network, RDD 2022, object detection, YOLOv5, ECA-Net, label smoothing, K-means++, Focal loss

1 Introduction

Modern civilizations rely notably on transporting individuals and commodities through road networks, leading to a persistent deterioration of road infrastructure and a resilient road maintenance system. Global spending on road repair and maintenance exceeds \$400 billion annually [34]. As a result of the comparatively elevated expenses associated with the construction of road networks over the last ten years, the maintenance of roads has become a crucial focus for the department responsible for traffic management. Several studies and investigations have been carried out regarding the impact of road deterioration on both safety and the economy [19], [27]. The principal aim of researchers in this field is to reduce expenditures associated with inspections and maintenance. Furthermore, creating efficient and economical technologies for detecting road surface cracks is crucial in establishing a sustainable road management framework [7].

The present techniques utilized for evaluating road damage can be classified into three distinct categories: manual assessment, automatic assessment, and image processing. Road assessment in developing nations often involves the utilization of manual inspection techniques. The conventional evaluation method demands a significant allocation of human resources, material assets, and time frame investment. Members of the maintenance department must assess the severity of road damage in real-time. However, in practical circumstances, the manual procedure proves to be tedious due to the extensive nature of the testing routes and the significant resource requirements of the task. The employment of automated detection systems for road assessment is on the rise. Using sensor-equipped vehicles exemplifies technological progress [33]. However, the complexity of road conditions poses a challenge for automated assessment tools. Consequently, the attainment of precision and swiftness becomes increasingly challenging. Additionally, the equipment requires costly hardware, which incurs high maintenance expenses.

Image processing techniques can integrate high efficiency while maintaining low cost. As technological advancements are made, the precision of detection is increasing. As a result, many researchers have employed image-processing methodologies to identify road deterioration [23, 24]. The complexity of road surroundings makes manual feature extraction impractical using conventional image-processing techniques. In contrast to traditional image processing methods, deep learning-based image processing techniques have gained widespread adoption to detect road damage due to their superior accuracy, speed, and embeddability [39].

Deep learning technology has made significant advancements in detecting road damage. The authors Naddaf et al. (2019) [22] employed the Faster R-CNN network to detect road damage. They additionally evaluated the impact of different lighting and weather conditions on the detection process. A study by Mandal et al. [18] presented a novel approach for detecting road damage using the one-stage YOLO CSPDarknet53

network. The authors' work contributes to the growing research on the road damage dataset, further expanding the range of models explored in this domain. Although the research in road damage detection has made progress, there remains significant potential for improvement in accuracy and detection speed.

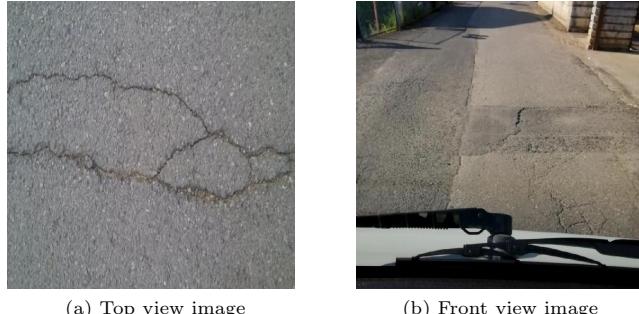
In recent years, the maintenance of roads has become a critical responsibility for traffic management departments due to the rapid expansion of the traffic network. Road damage detection and classification application can be used for various purposes. Some of its applications are Structural Health Monitoring, Autonomous Personal Mobility Vehicles, and Intelligent Transportation Systems (ITS). This study aims to design a deep neural network model that utilizes object detection techniques to identify instances of road damage in photographic representations of road surfaces. Our contributions can be summarized below:

- The Efficient Channel Attention module (ECA-Net) was initially incorporated into the model's backbone and neck network to enhance the model's accuracy.
- Label smoothing techniques are employed as a way of enhancing the generalization of the model.
- We utilized the K-means++ algorithm to identify an acceptable anchor box for our dataset instead of the default COCO dataset-based anchor box.
- Subsequently, this research utilizes Focal Loss, a cost-sensitive loss function, to address the class imbalance by assigning higher importance to challenging data points.
- Finally, an additional prediction layer has been incorporated to extend the model's learning capacity across various objects and scales.

2 Related Work

The employment of methods based on deep learning has gained substantial acceptance because of their remarkable efficacy across a wide range of domains, such as computer vision, soil science, and solar flare prediction. The transportation industry is subject to the effects of technological advancements. The rapid progress and extensive implementation of deep learning technologies can potentially enhance the ability to detect road damage. According to current studies, maintaining road infrastructure is essential for establishing a secure, economically feasible, sustainable transportation system. Therefore, researchers are now aiming to utilize techniques such as deep learning to evaluate road damage and optimize cost reduction.

Several methodologies based on deep learning have achieved noteworthy advancements. The efficacy of deep learning techniques has catalyzed the creation of additional algorithms that rely on feature extraction and pattern recognition to detect road damage. Oliveira and Correia [25] proposed using the mean and standard deviation from unsupervised learning to discriminate between road cracks and non-crack blocks. Cord and Chambon [5] suggested using the Adaboost algorithm to select structural descriptors that effectively depict fractured images and produce the crack saliency map. Despite the impressive outcomes produced by these techniques, their practical utility is significantly limited by their dependence on extracted features. The detection of objects can be achieved through various techniques, with Faster R-CNN and YOLO



(a) Top view image

(b) Front view image

Fig. 1: Different types of the image used in road damage detection research, (a): Top view image (b): Front view image.

being two of the approaches most commonly used. The Faster R-CNN approach was utilized by Li et al. to propose a method for the recognition of road surface damage [12]. The system can discriminate among six distinct categories of road surface defects and accurately pinpoint the precise location of the affected region within a given image. The research employed 5,966 photographs, resulting in an average precision rate of 96.3%.

Researchers use two distinct image types for road damage detection study. One is a snapped photo of the top view, while the other is of the front perspective. It is described in figure 1. Top-view photographs are acquired from the road's surface, while a dashboard-mounted camera captures front-view images.

Top-view damage photos are less complicated, and most studies on top-view damage have excellent detection accuracy. However, most of these models solely exhibit damage caused by cracks. Most studies based on top-view images did not attempt to classify different types of road damage. Naddaf et al. [21] used a Bayesian technique to optimize CNN design to maximize the abstraction of the final model with a 97% percent accuracy rate. In their study, Yusof et al. [44] conducted fine-tuning of the filter sizes in a Convolutional Neural Network (CNN) to attain a satisfactory accuracy rate of 94.5% in determining various types of cracks in road photographs. The authors' findings suggest that this approach may be effective in road maintenance and inspection. The majority of research about top-view images is centered around semantic segmentation. In their study, Zhang et al. [45] proposed a CrackNet model that does not incorporate a pooling layer to identify cracks and non-cracks in asphalt pavement. The model achieved a high accuracy rate of 90.1%. Liu et al. [13] utilized the U-Net architecture to identify cracks in concrete. The model that has undergone training exhibits a high degree of accuracy in identifying cracks present in images. FCNs can be outperformed by this approach, as it requires a smaller number of training sets to achieve superior results. Wu et al. achieved a recall rate of 97% through the utilization of diverse image processing techniques on the UNET model, as reported in their study [41]. Zhang et al. [46] developed a CrackNet architecture that does not incorporate a pooling layer to detect cracks on roads. The model achieved a high level of accuracy, specifically 90.1%. The model exhibited impressive results by attaining a

precision of 98.00% and an accuracy of 97.92% in the classification of various types of crack damages. While the models mentioned above exhibit remarkable detection capabilities, their practical implementation is restricted. The acquisition of such road images through specialized vehicles regularly incurs substantial costs.

The road damage detection model using front-view images focuses on developing detection and classification models employing images captured by dashboard-mounted cameras. Identifying road damage through automated means is a viable and pragmatic approach in real-world scenarios, with extensive ongoing research being conducted in this domain. However, the procedure is considered challenging due to the complex and obscure nature of the damage in this particular type of image. Furthermore, apart from the road surface, the contextual elements captured in these photographs include environmental factors such as atmospheric conditions, geographical features, and other noise, which pose challenges to identifying road damage. Nevertheless, effectively utilizing these images for damage detection would yield significant benefits in road monitoring activities. Currently, the repositories of front-view images are extensive. Furthermore, it has the potential to be swiftly and effortlessly augmented. Again, utilizing comprehensive road coverage images during inspections and camera installation procedures is simple and cost-effective. Despite the numerous benefits of using images as a resource, studies in this area remain limited due to images' diverse and complex textures. Deep learning-based models have been proposed as a viable solution to tackle these challenging requirements effectively [42]. However, only a few models have their performance evaluated using images taken by dashboard-mounted cameras.

Shim et al. [29] proposed a lightweight semantic segmentation method. This is noteworthy as most research on front-view images typically focuses on object detection. Their proposed approach enhances precision, decreases parameters by 85.7%, and improves computational speed by 6.1%. Furthermore, they enhanced the model's parameters without considering any potential impact on its detection speed. They proposed additional research utilizing a lightweight auto-encoder network with a hierarchical layout to detect road damage [30], [31]. Most of these models did not attempt to categorize damage into distinct classifications. Instead, researchers employ object detection models, such as Faster R-CNN, YOLO, and SSD, to identify and categorize instances of road damage. Maeda et al. [17] proposed combining a Single shot detector (SSD) with an Inception backbone and an SSD with a MobileNet backbone to detect and classify eight categories of road damage with a 71% Recall score. Jeong et al. [10] proposed a model based on YOLOv5x architecture, which yielded an F1 score of 67%. However, the detection speed could be more satisfactory. Wang et al. [38] identified and classified road damage with a 62.5% F1 score using a Faster R-CNN model with data augmentation techniques. Guo et al. [6] employed the Mobilenetv3 architecture as the backbone and trained YOLOv5s using the RDD 2020 dataset. The algorithm utilized by the authors integrates coordinate attention, K-means, and label smoothing techniques. Incorporating diverse techniques in their algorithm resulted in a 2.5% enhancement of the F1 score compared to the YOLOv5 baseline model. Wan et al. [36] trained RDD 2020 using ShuffleNetv2 as the backbone and ECA-Net as the attention mechanism in YOLOv5s. They use focal loss and BiFPN neck instead

of YOLOv5's original neck. Their model increases the efficiency of image recognition by 22.3%. Recently, Pham et al. [26] employed the YOLOv7 model with coordinate attention and label smoothing techniques on the RDD 2022 dataset. In addition, they customize the default image enhancement settings of YOLOv7. Exhibits enhanced capacity for generalization and resilience.

Rapid and precise road damage identification can greatly benefit the road maintenance industry and contribute significantly to the economy. The maintenance of road pavement technology is of utmost importance in ensuring the preservation of pavement quality, injury prevention, and prompt detection and resolution of road surface issues.

3 Methodology

To detect and classify road damage, our team has opted to utilise YOLOv5 as our object detection model. Initially, a thorough exposition of the YOLOv5 framework will be presented. Subsequently, we shall expound upon our proposed adapted framework tailored to suit our designated objective. Finally, a comprehensive elucidation of every altered constituent will be presented, delineating the modifications implemented to augment the precision and efficacy of the model.

3.1 YOLOv5s Architecture

The YOLOv5 model is a one-stage object detection framework that comprises four distinct versions, namely YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. These versions are regulated by two parameters, namely multiple-depth and multiple-width. The smallest and quickest model is YOLOv5s, with a parameter of 7.0 M, while the most prominent model is Yolov5l, with 76 M. The architectural structure is partitioned into four distinct components, the input, backbone, neck, and head, as illustrated in Figure 2.

3.1.1 Input

The YOLOv5 network initiated image pre-processing, then resizing the image to $640 \times 640 \times 3$ before feeding it into the backbone network. The greater efficiency of YOLO with a 640×640 image can be linked to the algorithm's built-in design [28]. The YOLO algorithm partitions an image with dimensions of 640×640 into a grid consisting of 20×20 cells, where each cell measures 32×32 pixels. This particular grid dimension facilitates an optimal equilibrium in detecting objects of varying sizes.

The YOLOv5 input undergoes adaptive image scaling and Mosaic data enhancement for processing. The process of adaptive image scaling involves resizing the source image to a predetermined size, followed by its input into the neural network for detection. The utilization of mosaic data enhancement is observed during the training phase of the model. The Mosaic data enhancement technique involves the integration of four images employing random arrangement, random cropping, and random scaling to augment the training dataset. Subsequently, the image integrated with Mosaic undergoes rotation, panning, flipping, and color adjustment.

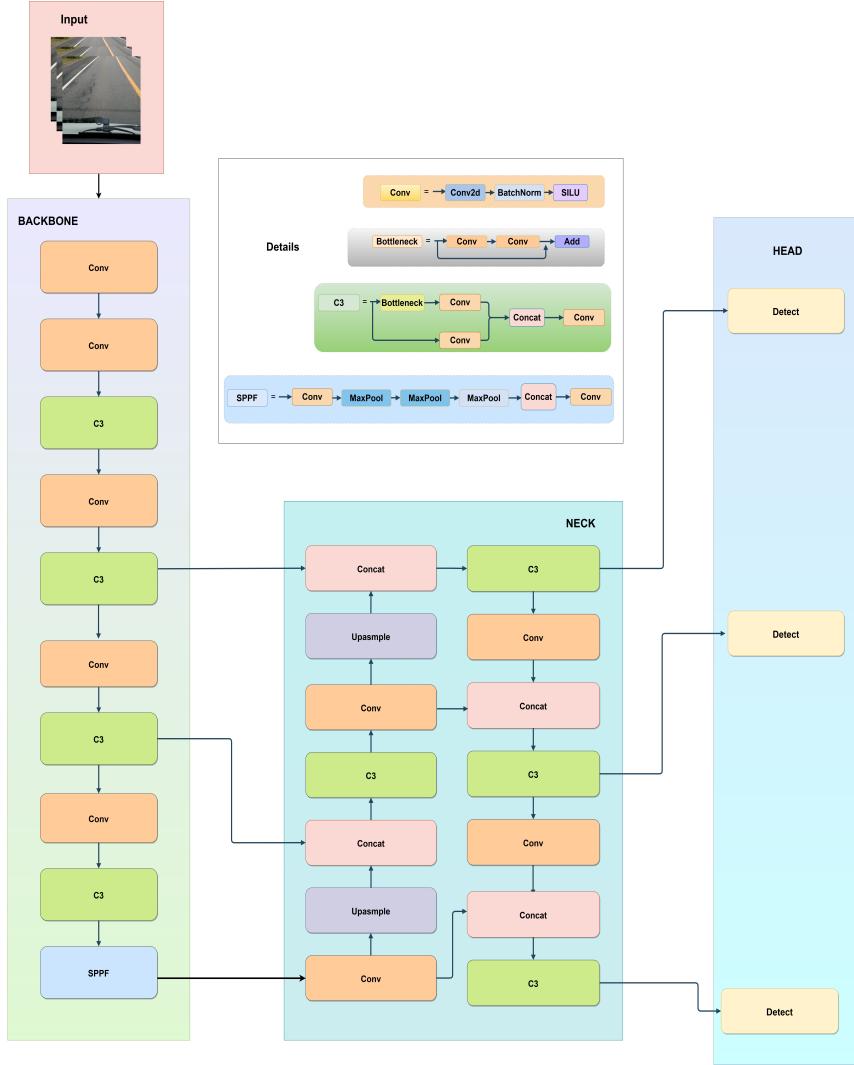


Fig. 2: A brief overview of YOLOv5s Architecture.

3.1.2 Backbone

The primary constituents of the backbone network are the Convolution Module (Conv), which encompasses Convolution, Batch Normalization, and Sigmoid Linear Unit operations; the C3 module; and the Spatial Pyramid Pooling Fusion(SPPF) module. The parameter N in C3 specifies using the BottleNeck module N times within a single module. Including a residual structure within the BottleNeck, module has been observed to effectively address the gradient disappearance arising from the deepening of neural networks. The SPPF structure evolved from Spatial Pyramid Pooling(SPP) to alleviate computational pressure and increase network speed. The network's feature

extraction is accomplished via maximum pooling with varying pooling kernel sizes to enhance the network's perceptual field.

3.1.3 Neck

The neck can be regarded as the feature fusion network. The neck includes the Conv module, C3 module, an upsampling module, and a Concat module. The neck structure employs a fusion approach that combines top-down and bottom-up features. The bottom-up approach facilitates the effective transmission of positional data from the lower to the upper layers. The fusion of high-level and low-level features through the top-down path enhances the detection efficacy of small objects. Integrating the two paths can consolidate the characteristics of distinct detection layers originating from diverse backbone layers. The utilization of this particular structure has the potential to enhance the detection efficacy of dense objects.

3.1.4 Head

The YOLOv5 algorithm's head mechanism employs three different YOLO Head classifiers to identify feature images of varying scales: large-scale, medium-scale, and small-scale. The prediction outcome encompasses the projected category, level of confidence, and positioning of each predicted box. Hence, it is essential to compute the losses of the three predicted results distinctly during the formulation of the loss function. The YOLOv5 model's loss function comprises three parts: namely, the confidence loss (L_{obj}), localization loss (L_{box}), and classification loss (L_{cls}). The aggregate loss is computed as the summation of three distinct losses, and each is assigned a specific weight. Equation 1 represents the full loss function.

$$\text{LOSS} = a \times L_{\text{obj}} + b \times L_{\text{cls}} + c \times L_{\text{box}} \quad (1)$$

The weighting factors denoted by a , b , and c correspond to the three partial losses. Equation 2, 3, and 4 represents all these partial loss.

$$L_{\text{box}} = 1 - \text{CIOU} \quad (2)$$

$$L_{\text{cls}} = \sum_{i=0}^{S^2} \prod_{i} \sum_{c \in \text{class}} (p_i(c) - \hat{p}_i(c))^2 \quad (3)$$

$$L_{\text{obj}} = \sum_{i=0}^{S^2} \sum_{j=0}^B \prod_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \prod_{ij}^{obj} (C_i - \hat{C}_i)^2 \quad (4)$$

The symbol \prod_i^{obj} is utilized to ascertain the presence of a detection object in cell i . The \prod_{ij}^{obj} is utilized to ascertain the responsibility of the j th ($j = 0, \dots, B$) prediction box located in cell i ($i = 0, \dots, S^2$) for detecting the given object. The notation $p_i(c)$ is utilized to indicate the classification score of a specific category C , whereas $\hat{p}_i(c)$ denotes the actual value of the category linked with the prediction box. The weight coefficient in the absence of an object is denoted as λ_{noobj} , while C_i represents the confidence score. Additionally, the intersection portion of the prediction box and the actual box is defined by \hat{C}_i .

Binary cross entropy loss (BCE) is mainly used in classification and confidence loss, while CIoU (Complete Intersection Over Union) loss is commonly used for localization loss.

$$L_p(q) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \quad (5)$$

Equation 5 defines the binary cross-entropy loss. In the given Equation, y_i denotes the actual label of the sample in the real-world scenario, whereas $p(y)$ signifies the anticipated likelihood of the data point being affirmative, taking into account all N data points. The augmented form of the localization loss is described in Equation 6.

$$L_{\text{box}}(l, g) = 1 - \text{CIoU}(l, g) = 1 - \text{IoU}(l, g) + \frac{\rho(l, g)^2}{c^2} + \alpha v \quad (6)$$

$$\text{IoU}(l, g) = \frac{l \cap g}{l \cup g} \quad (7)$$

$$\nu = \frac{4}{\pi^2} \left(\tan^{-1} \left(\frac{g_w}{g_h} \right) - \tan^{-1} \left(\frac{l_w}{l_h} \right) \right)^2 \quad (8)$$

$$\alpha = \frac{\nu}{(1 - \text{IoU}(l, g)) + \nu} \quad (9)$$

The symbols l and g in equation represent the ground truth and, prediction box, consequently. The symbol ρ denotes the Euclidean distance. The parameter c represents the diagonal length of the smallest possible closed box that can cover both boxes above. The intersection degree of IoU of two boxes is represented by equation 7. v in equation 8 denotes the aspect ratio's consistency. Finally, the symbol α , denoted in equation 9, represents the trade-off parameter.

3.2 Improved YOLOv5s

The YOLO architecture size varies considerably from nano to extra-large based on multiple-depth and multiple-width parameters, and selecting the appropriate model for training is essential. Accuracy will improve as the model size increases, but model deployment will become problematic. The YOLOv5m and other large versions are unsuitable for real-time road damage detection and classification because they require excessive parameters. So we tried to improve the YOLOv5s model, which contains 7M million parameters suitable for detection speed. Figure 3 represents our improved YOLOv5s network architecture. It satisfies the requirements for road damage detection tasks, particularly regarding detection speed. The primary obstacle in road damage detection and classification is sustaining detection speed while improving accuracy. To accomplish this, we modify the architecture of YOLOv5s significantly. Comparing diverse backbones the default backbone of YOLOv5, CSPDarknet, is chosen as the model's backbone network. First, we incorporated the efficient Channel Attention module (ECA-Net) into the model's backbone and neck network to enhance the model's accuracy. Second, we use label smoothing techniques to strengthen the generalizability of the model. Thirdly, we use the K-means++ algorithm rather than the

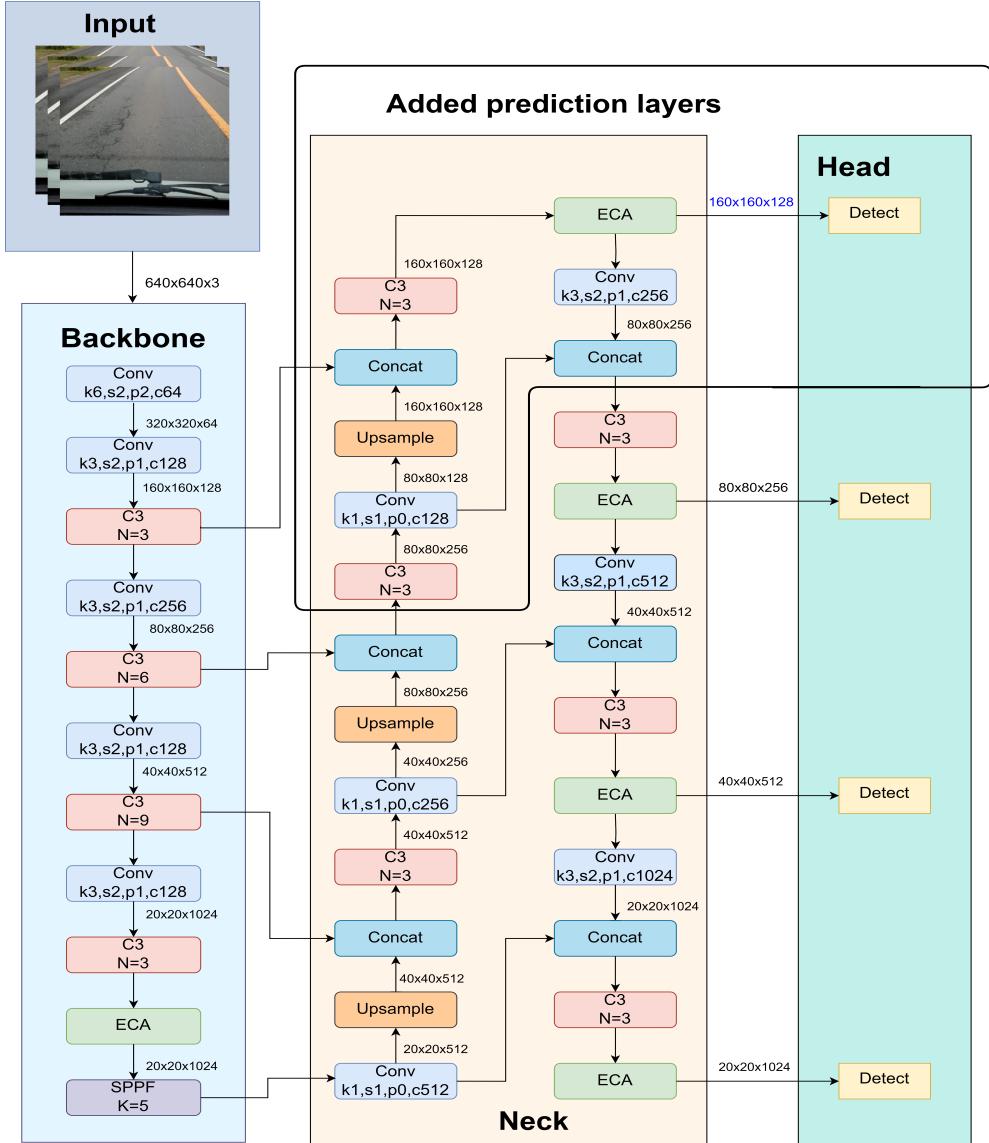


Fig. 3: A brief overview of Improved YOLOv5 Architecture.

default K-means algorithm to iteratively analyze the dataset to identify an appropriate anchor box for our dataset rather than the default COCO dataset-based anchor box. Fourth, we have utilized focal loss instead of Binary cross-entropy loss because it is more effective when there is an imbalance between positive and negative examples. Finally, we have added an additional layer of prediction to improve the model's ability to learn across numerous objects and scales.

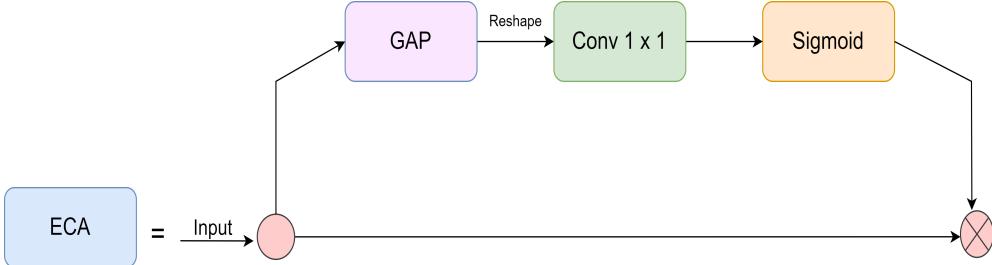


Fig. 4: The ECA-Net structure.

3.2.1 Enhancing Object Detection with Attention Mechanism

The analysis of human vision yielded the concept of attention mechanisms. Thus, attention mechanisms in deep learning are comparable to the selective attention mechanism of humans. Their primary objective is to select more critical information regarding the current task objective from many pieces while disregarding other unimportant details. Spatial, channel and mixed mechanisms are the three primary categories of attention mechanisms.

The ECA-Net is a recently developed attention mechanism characterized by its lightweight and efficient design [37]. It improves the Squeeze and Excitation Networks (SENet) framework [8]. The YOLOv5 model was enhanced by integrating an ECA attention mechanism to enrich defect detection accuracy in complex scenes. Figure 4 depicts the ECA attention mechanism. This integration facilitated the model's focus on object regions while training and reducing insignificant information representation. ECA employs the one-dimensional sparse convolution filter to generate channel weights to replace the full connection layer in the SENet. According to the study's findings, implementing the proposed method can enhance the precision of predictions without imposing additional computational demands. Furthermore, the method can be readily integrated into mobile networks, facilitating widespread deployment.

The ECA-Net model utilizes Global Average Pooling(GAP) to transform the input feature map X , a real-valued matrix with dimensions $W \times H \times C$, into a singular real value. The resulting features are X_{avg} , a matrix with $1 \times 1 \times C$ dimensions. The variables C , W and H correspond to the feature's width, height, and channel, respectively, as indicated in Formula. The feature extraction process from X_{avg} is achieved through a convolutional operation using a one-dimensional convolution kernel of size K . K is represented as equation 11.

$$\bar{X} = \frac{1}{WH} \sum_{i=1,j=1}^W X_{ij} \quad (10)$$

$$K = \frac{\log_2 C + 1}{2} \quad (11)$$

The sigmoid activation function is commonly employed to trigger the output result following convolution to obtain the weight parameter. This parameter reflects the correlation and significance of all the channels. Then, the weight variable W is applied

to the input feature map, resulting in the completion of the capturing of each channel feature of the feature map. This approach amplifies significant characteristics by assigning greater weights, whereas insignificant factors are subdued by assigning lesser weights.

To complete our objective, we examined four attention modules. Various positions were utilized for their incorporation, including after each C3 module in the backbone, solely after the final C3 module in the backbone, the neck portion of the architecture, and the neck and backbone. Our findings indicate that incorporating an attention module into the backbone of the model results in increased weightiness due to the substantial number of parameters present in each C3 block of the backbone. The feature extraction process that passes through these attention modules also contributes to this increased weightiness. We added the attention module right after the final C3 block and before the SPPF. By placing the attention module before the SPPF, the refined feature representation from the attention module can be efficiently combined with the spatially pooled features. This strategic placement facilitates enhanced network performance by enabling selective focus on the most relevant features [40]. Furthermore, including ECA-Net after every C3 block in the downsampling process can improve the network’s performance by augmenting its capacity to capture long-range interdependencies among distinct spatial locations within the feature map [4].

3.2.2 Optimized Anchor Generation

The initial dimensions of the anchor box in YOLOv5 are determined through edge clustering using the K-means algorithm [9] on the MS COCO dataset [14]. However, the MS COCO dataset predominantly comprises large and medium-sized targets. In contrast, this study’s Road Damage Detection dataset encompasses small and medium-sized targets that deviate from the COCO dataset. Hence, the initial dimensions of the anchor box in YOLOv5 are deemed unsuitable for the RDD-2022 dataset samples. To address this issue, We employed the K-means++ clustering algorithm instead of the K-means algorithm to reconfigure the anchor box’s dimensions for the samples present within the RDD 2022 dataset. K-means++ is an enhanced variant of the K-means clustering algorithm that addresses certain limitations of the original K-means algorithm [1]. These limitations include random initialization of centroids, slower convergence speed, and reduced accuracy. The present study employs the K-means++ algorithm to perform clustering on the training set, acquiring 12 anchor boxes that serve as the initial clustering boxes. Upon completing dataset reclustering, a novel anchor box was discovered, comprising the following Coordinates: ‘15, 15, 44, 26, 33, 63’, ‘116, 24, 81, 62, 44, 120’, ‘228, 36, 92, 142, 175, 97’, and ‘191, 190, 126, 464, 410, 199’.

3.2.3 Utilizing Focal Loss Strategy

We employed focal loss to substitute the binary cross entropy loss. The focal loss is a variant of the conventional cross-entropy loss function, which aims to decrease the impact of accurately classified instances on the overall loss while emphasizing the significance of misclassified cases [15]. In addition, focal loss is a technique utilized in object detection to tackle the problem of disproportionate representation of foreground and background classes. YOLOv5 can increase detection performance by employing

focus loss, particularly for small objects that are difficult to identify due to their small size and low contrast. Furthermore, focus loss can reduce the probability of incorrect identifications by assigning relatively lower weights to less complicated instances, reducing the influence of unwanted information on the final detection outputs.

The YOLOv5 model employs the focal loss function on the classification branch of the network. This branch forecasts every anchor box's objectness score and class probabilities. The calculation of the focal loss is as equation 12:

$$FL(p_t) = -(1 - p_t)^\gamma \log(p_t) \quad (12)$$

The equation involves the predicted probability for the actual class, denoted as p_t , and a hyperparameter γ , which modulates the degree of weighting assigned to challenging examples. When γ equals zero, the focal loss function simplifies to the conventional cross-entropy loss.

The modified equation for binary cross-entropy with focal loss is given in equation 13:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) - (1 - \alpha_t)p_t^\gamma \log(1 - p_t) \quad (13)$$

Where p_t is the projected likelihood of the actual class, α_t is a binary class sign (i.e., $\alpha_t = 1$ if the actual class is present and $\alpha_t = 0$ otherwise), and γ is the focusing parameter that regulates how much weight is placed on complex examples.

3.2.4 Label Smoothing Algorithm

Label smoothing is a widely employed regularisation technique in deep neural networks (DNNs) that effectively mitigates the issue of overfitting during training. Its implementation has been observed to enhance the classification accuracy of Deep Neural Networks [43]. Overfitting is a prevalent issue arising during deep learning models' movement. It refers to a scenario where the model exhibits good performance on the training set, but its performance on the test set is unsatisfactory. This poor generalization ability renders the model ineffective in predicting unknown data. It is widely accepted that including mislabeled labels is a crucial measure for reducing the potential for excessive confidence in identifying data used for training.

The technique of label smoothing involves the integration of a uniform distribution, wherein the conventional label vector y_i is substituted with the modified label vector \hat{y}_i . Equation 14 and 15 represent label smoothing.

$$\hat{y}_i = y_i(1 - \delta) + \frac{\delta}{K} \quad (14)$$

The total count of K multiclass groups and the small hyperparameter δ (typically set to 0.1) are the focal points of assessment.

$$\hat{y}_i = \begin{cases} 1 - \delta, & i = \text{target} \\ \frac{\delta}{K}, & i \neq \text{target} \end{cases} \quad (15)$$

In our study, we analyzed the effectiveness of the label smoothing method with a default value of 0.1, which has yielded favorable results in various scenarios [20].

3.2.5 Added Prediction Layer

Initially, YOLOv5 only performed feature prediction on the final three C3 layers. The utilization of multiple feature images with varying dimensions is employed to detect objects of varying sizes. The network acquires three feature images of different dimensions from the initial input image using downsampling at 8x, 16x, and 32x. After deep convolution, the feature image contains lots of semantic information. Nevertheless, the loss of certain positional information about the object occurs during multiple convolutions, rendering it impractical to detect tiny objects. The present study incorporates a 4×downsampling in the spatial resolution of the initial input image utilizing the YOLOv5 backbone network. As illustrated in figure 3, the initial image undergoes a downsampling process four times. It is subsequently inputted into the feature fusion network, acquiring a feature map with a different size. As each C3 has been followed by an ECA-Net, the detection head obtains its feature map from ECA-Net. The image in question exhibits a reduced receptive field and increased location information. By utilizing the multilevel feature information of complex objects, the featured image can enhance the detection efficacy of the model in complex environments.

4 Result Analysis

In this section, we endeavoured to present all of our experiments. Through the execution of these tests, we aimed to conduct a comprehensive evaluation of our proposed model. First, we will discuss the training configuration and dataset employed, followed by a discussion of the implemented study.

4.1 Dataset Description

Various intelligent equipment, such as drones and dashcams, have made the automatic Collection of road damage data much more accessible in recent years. Especially the dashboard-mounted camera is quicker to install and more effective than manual scanning techniques in detecting road damage. Moreover, classifying road damage data manually is a time-consuming process. We employ an existing benchmark dataset to train the road damage detection model. Most top-view datasets don't have the class name; hence, classification is impossible. The road damage detection dataset (RDD 2022) is the largest front view road damage dataset ever compiled about road damage, encompassing four types of damage in six countries. This dataset follows the preceding RDD dataset and is an expanded version of the earlier RDD 2020 [2] and RDD 2018 [17] datasets. While the RDD 2018 dataset has 9,053 photos and the RDD 2020 dataset contains 26,336 images from three countries, the RDD 2022 dataset [3] comprises 47,420 images compiled from numerous sources in six nations. It includes Japan, the United States, Norway, the Czech Republic, India, China Motorbike, and China Drone. Figure 5 depicts the distributions of damage types (of the four main damage types) among the six countries. From a present standpoint, this cutting-edge dataset is the most practical and effective for road damage detection and classification.

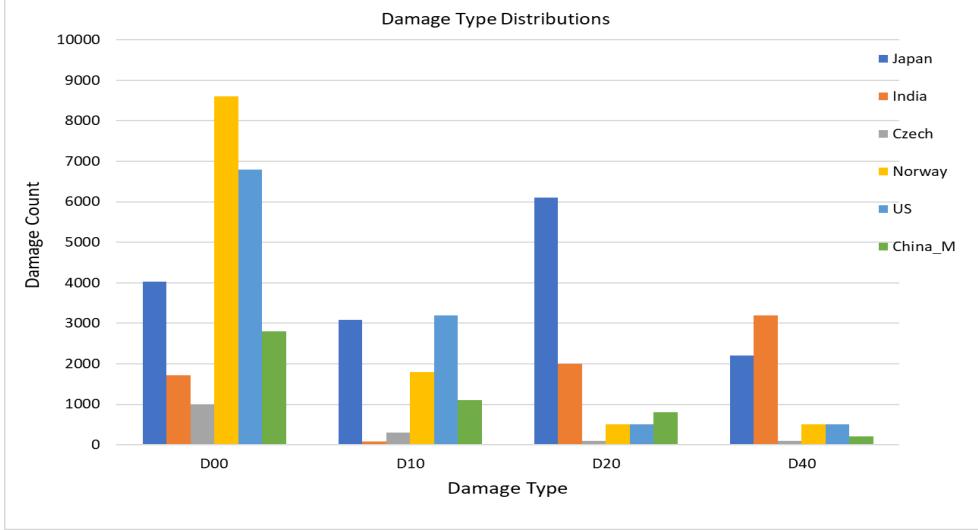


Fig. 5: Damage category distribution in the current benchmark state-of-the-mark dataset.

4.2 Data Processing and Training Configuration

RDD 2022 has 36,000 photos with an annotation file, most with unique labels for a single country. For example, China Motorbike includes a label titled Repair. Japan and India have many labels with the names D01, D11, D43, D44, and D50, which are unknown in other countries. Therefore, we manually deleted these unique classes and retained the four common classes in our research. We included four types of damage: D00: longitudinal cracks, D10: transverse fractures, D20: alligator cracks, and D40: potholes. Figure 6 depicts an example of these four classes. Additionally, there are a large number of photos that don't have any annotation. These images are pointless for our detection model; we also eliminated these images. A close inspection of the RDD 2022 dataset provides us with a few insights for selecting images efficiently. Countries such as Norway and Japan had various image sizes. Such as, the resolution of the Norway photos (3072×3720) is incompatible with our GPU configuration. We manually scale each image to 640×640 to train our YOLOv5 model. Moreover, we didn't include China-Drone in our work because it needed to be obtained from a front-mounted vehicle image. Finally, We divided the dataset into training, validation, and testing sets, with the testing set including 1,542 images. The size of the testing set is adequate for estimating the model's performance on new, unobserved data. The training and validation sets allow the model to learn from enough examples and prevent overfitting. The results of the different Classes of this study and their quantity are presented in figure 7. The provided information comprises three distinct components. Firstly, it includes details about the dataset's type and quantity. Secondly, it encompasses information regarding the distribution of the object's center point's position. Lastly, it comprises information about the distribution of the object's size.

Upon creation of the training set, the YOLOv5s model is subsequently trained through the utilisation of the yolov5s.pt pretraining weights, and we have created our “custom.yaml” configuration file to specify the model architecture. Logarithmic and weight files are recorded and stored throughout the training process. The training configuration is given in table 1.

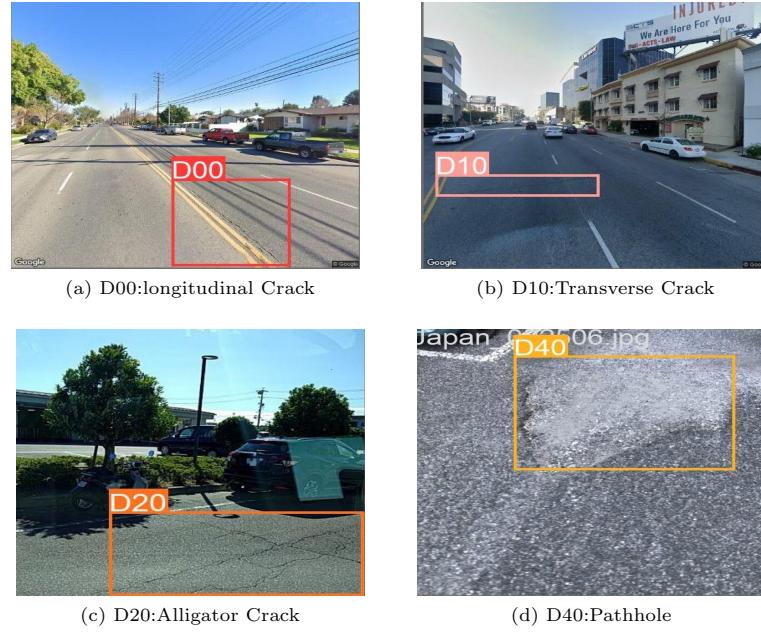


Fig. 6: Different kinds of damages used in this research.

4.3 Evaluation Metric

This research utilizes various evaluation metrics, including precision, recall rate, F1 score, mean Average Precision (mAP), number of parameters, and GFlops (i.e., One

Table 1: Training configuration of our improved YOLOv5s.

Hyper parameter	Value
Optimizer	SGD
Epochs	100
Batch size	16
Image Size	640 x 640
workers	6
Auto Anchor	False (Default=True)
Label_smoothing	0.1 (Default=None)

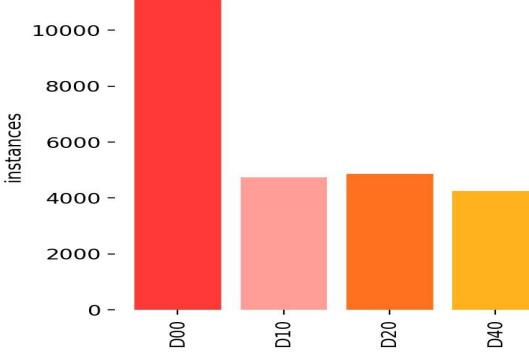


Fig. 7: Different Class of this study and their quantity.

billion Floating-Point Operations per second). The precision metric of a model determines the proportion of correctly predicted positive labels compared to the overall number of predicted positive labels. The precision equation is presented in equation 16.

$$\text{Precision Score} = \frac{TP}{(FP + TP)} \quad (16)$$

Recall is a metric that evaluates the accuracy of a model's ability to predict positive outcomes based on actual positive results, in contrast to other performance metrics. The metrics of True Positive (TP), False Positive (FP), and False Negative (FN) are employed in the calculation of precision and recall. The Recall equation is provided in equation 17.

$$\text{Recall Score} = \frac{TP}{(FN + TP)} \quad (17)$$

The F1 score is a metric that evaluates a model's effectiveness by considering both its precision and recall scores. The F1 score is a performance metric that assigns equal importance to Precision and Recall when assessing the effectiveness of a machine learning model. The technical expression for this is the harmonic mean of the precision and recall scores. The mathematical expression for the F1 score is presented in equation 18.

$$F1 \text{ Score} = \frac{2 * \text{Precision Score} * \text{Recall Score}}{(\text{Precision Score} + \text{Recall Score})} \quad (18)$$

The Mean Average Precision (mAP) is a typically used metric for assessing object detection models. Initially, the precision and recall metrics are calculated for every detection within the category across a spectrum of confidence threshold values. Subsequently, the mean Precision is determined by utilizing the region beneath the class's precision-recall curve (AUC). Upon computation of the AP score for every class, the mAP score is derived using averaging these individual scores. The mAP score is a

significant metric that provides a comprehensive evaluation of the detection performance of the model across all categories in the dataset. The mathematical expression for mAP is presented in equation 19.

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (19)$$

where AP_K = The AP of class k and n = The number of classes.

GFlops are employed for assessing the efficacy of a model or algorithm. Typically, a lower GFlops value corresponds to reduced computational demand for model rendering, lower hardware performance requirements, and increased feasibility for implementation in low-end devices. Measuring parameters is crucial in assessing a model's complex nature and computational requisites. A model's complexity is directly proportional to its number of parameters. A more significant number of parameters signifies the complexity of a model, which could require more computational resources for training and deployment. Hence, it is essential to evaluate the efficacy of object detection models by examining their parameter count and performance on relevant metrics such as mAP, Precision, and Recall.

4.4 Evaluation of Performance

In this section, we evaluate our proposed method's effectiveness through various experiments.

4.4.1 Finding Suitable Backbone

The first study aimed to determine the most effective backbone for the given task. The objective of our study was to maximize the outcome by limiting the parameter count. We tried different kinds of backbone to meet our requirements. Table 2 represents this experiment. The baseline yolov5s model yields a mean average precision of 65.91% while utilizing 7.01 million parameters and a GFlops of 16. With the Swin transformer, the result drops to 3.44% in terms of mAP, attaining 62.47% and a 4.07% decrease in the F1 score, achieving 61.14%. Moreover, the GFlops increased hugely, reaching a value of 102.00. The parameter count displayed a rise of 1.59 million. Using the RepVGG backbone resulted in a mean average precision (mAP) of 61.32% and an F1 score of 60.51%. We have also tried mobile net v3 and Mobileone backbone and found that their parameters count and GFlops are relatively less than other backbone

Table 2: Experimental result of the different backbone with YOLOv5s architecture

Backbone	Precision (%)	Recall (%)	F1 Score (%)	mAP@0.5 (%)	No. of Params. (M)	GFlops
CSPDarknet	66.70	63.80	65.21	65.91	7.01	16.00
Swin Transformer	64.11	58.45	61.14	62.47	8.60	102.00
RepVGG	63.12	58.24	60.51	61.32	7.30	17.14
Mobilenet v3	59.43	54.16	56.36	56.72	5.21	10.00
MobileOne	61.27	56.36	58.67	58.80	8.15	10.14

Table 3: Experimental result of different attention mechanism with YOLOv5s architecture

Attention Mechanism	Precision (%)	Recall (%)	F1 Score (%)	mAP@0.5 (%)	No. of Params. (M)	GFlops
CBAM	66.34	63.02	65.15	64.81	8.35	18.41
SE	65.56	62.31	63.89	64.72	7.41	17.56
CA	66.67	62.57	64.68	65.57	7.27	17.10
ECA	66.85	63.93	65.35	65.99	7.18	16.00

architectures. The MobileNet V3 model achieved a mean average precision (mAP) of 56.72%. The model’s parameter count and GFlops were recorded at 5.21 million and 10, respectively. Mobileone, a recently invented novel architecture by APPLE, got 58.81% mAP and a 58.67% F1 score. The parameter count for this is higher than Mobilenet v3 though both have the same GFlops of 10. Finally, we selected Yolov5’s default backbone CSPDarknet, as our backbone, representing the best performance while containing an acceptable amount of parameters.

4.5 Experiment of Different Attention Mechanism

The study revealed that the CBAM necessitates more parameters and GFlops, encompassing 8.35 million parameters, significantly higher than the other three attention modules. This is attributed to the CBAM providing both spatial and channel attention, despite not portraying the optimal result. Table 3 depicts this experiment. In our study, it was determined that the utilization of Squeeze and Excitation (SENet) and Convolutional Block Attention Module (CBAM) did not yield significant benefits compared to Channel Attention (CA) and Efficient Channel Attention (ECA) in our specific scenario. The results indicate that CBAM produces a mAP of 64.81%, whereas SE and CA yield mAP values of 64.72% and 65.57%, respectively. Therefore, the ECA attention module was identified as the optimal choice due to its relatively low parameter count compared to other attention modules and its ability to yield the highest performance metrics of 65.99% mAP and 65.35% F1 score when compared to other attention modules.

4.6 Ablation Experiment

To assess the reliability and importance of individual enhanced modules within the model, we utilise YOLOv5s as a benchmark and gradually integrate improved modules to perform ablation experiments. The assessment criteria used in this investigation comprise Precision, Recall, F1 Score, mAP, Number of parameters, and frames per second (FPS). Table 4 displays the results of the experimental findings. The default version of YOLOv5s exhibits a mAP of 65.91 and comprises 7.06 parameters. Including ECA-Net in the architecture has resulted in a 0.08% improvement in mAP. Additionally, there was an increase of 0.12 million parameters and a reduction of 9 FPS. Incorporating focal loss resulted in a 0.27% increase in mAP, while the number of parameters and FPS remained constant. Incorporating an adaptive anchor box by utilizing the K-means++ algorithm resulted in a 0.31% increase in mAP and a 0.97% increase in precision. Reducing parameters by 110,000 resulted in a corresponding

increase of 7 FPS. Upon increasing the label smoothing value by 0.01, a notable enhancement of 0.39% was observed in the Recall metric. Ultimately, incorporating an additional detection layer resulted in a 67.81% mAP score, an F1 score of 66.51% accompanied by a parameter increase of 1.05 million. The obtained FPS value in this experiment is 58, with an increased mAP of 1.9% from the Yolov5s baseline. Figure 8 represents the precision-recall curve of our proposed improved YOLOv5s model. Our newly added prediction layer helps to locate tiny objects, thus improving the detection performance. Figure 9 represents our model’s effectiveness in detection performance. This implies that this model is reliable in detecting and classifying road damage in real-time scenarios. Figure 10 demonstrates our proposed model detection results.

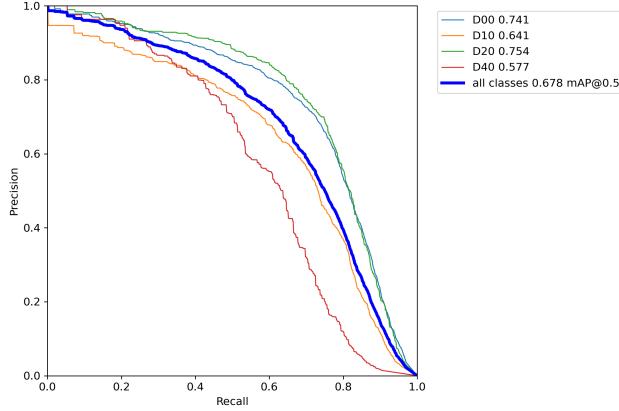


Fig. 8: Precision-Recall Curve of our proposed improved YOLOv5s.

Table 4: Ablation study on improved YOLOv5s

Model	Precision (%)	Recall (%)	F1 Score (%)	mAP@0.5 (%)	No. of Params. (M)	FPS
YOLOV5s-Baseline	66.70	63.80	65.22	65.91	7.06	68
YOLOV5s-Baseline + ECA	66.85	63.93	100	65.99	7.18	59
YOLOV5s-Baseline + ECA + Focal	67.94	62.96	100	66.26	7.18	59
YOLOV5s-Baseline + ECA + Focal + K-means++	68.91	62.64	100	66.57	7.07	66
YOLOV5s-Baseline + ECA + Focal + K-means++ + label smoothing	67.94	63.03	100	66.74	7.07	66
YOLOV5s-Baseline + ECA + Focal + K-means++ + label smoothing + Newly Added Detection layer	70.14	63.24	66.51	67.81	8.11	58

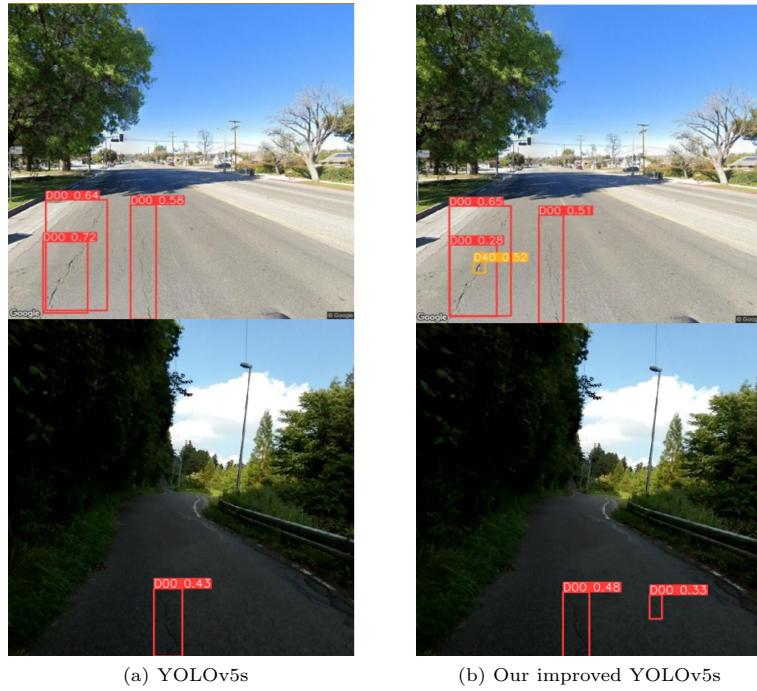


Fig. 9: Demonstration of tiny object detection using our improved YOLOv5s: (a) YOLOv5s; (b) Our improved YOLOv5s

Table 5: Comparison among different object detectors

Model	F1 Score (%)	mAP@0.5 (%)	No. of Params. (M)	GFlops	FPS
YOLOv8l	69.06	71.50	43.70	165.2	23
YOLOv8m	68.63	70.79	25.80	78.70	32
YOLOv5l	68.78	70.90	46.50	109.5	26
YOLOv5m	66.55	69.23	21.20	49.00	36
YOLOv7-tiny	63.68	65.67	6.27	13.80	65
YOLOv5s	65.22	65.91	7.01	16.00	63
YOLOv8s	66.46	67.70	11.21	28.40	53
Ours Improved YOLOv5s	66.51	67.81	8.11	16.30	58

4.7 Comparison of different YOLO-based Object Detector

All state-of-the-art large, medium, and small-sized versions of the Yolo model's results are given in table 5.

The YOLOv8l, YOLOv5l, YOLOv8m, and YOLOv5m represents all the large and medium model that we trained with RDD 2022 dataset where we got the best mAP and F1 score but with a large number of parameters and low FPS. The YOLOv8l attains the highest mAP of 71.50% with 43.7 million parameters among these models. The

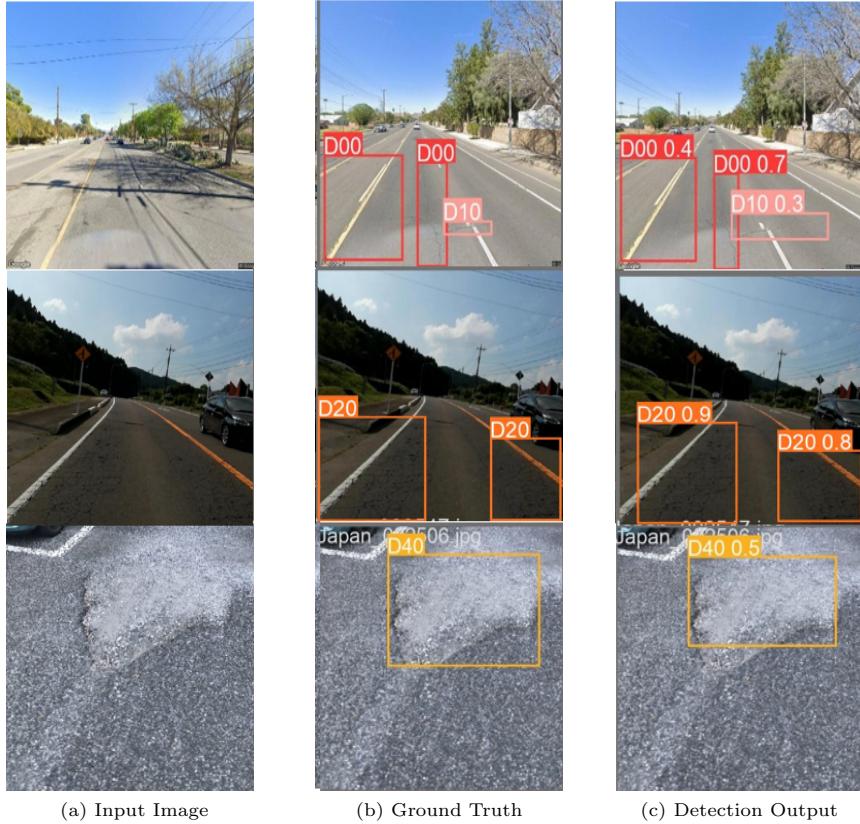


Fig. 10: Demonstration of our improved YOLOv5s model detection results: (a) Input Image; (b) Ground Truth; (c) Detection Output.

YOLOv8m achieved 70.79% mAP and 68.63% F1 score with 25.80 million parameters. With YOLOv5l and YOLOv5m, we got 70.90% and 69.23% mAP, corresponding with 46.50 and 21.2 million parameters. These large and medium-sized models' parameter counts and higher GFlops make the model heavyweight and unsuitable for real-time damage detection. We also experimented with all the small-sized model, which is suitable for real-time damage detection and are more lightweight to integrate into mobile devices. To assess this, we trained the dataset with YOLOv7-tiny, YOLOv5s, YOLOv8s, and our improved YOLOv5s.

The YOLOv5s attains 65.91% mAP with 7.01 million parameters. YOLOv7-tiny consists of a 6.27 million parameter, and there is a decrease of 0.24% of mAP from the YOLOv5s. Yolov8s consists of 11.2 million parameters, which is 4.2 million more than the v5s and contains larger GFlops than YOLOv5s, and it achieved 67.70% mAP. However, we got a 1.9% improvement from Yolov5 in terms of mAP and 1.29% in terms of F1 while increasing only 1.1 million parameters. Moreover, we got a 0.11% improvement in mAP and 0.05% in F1 score from YOLOV8s while containing 3 million

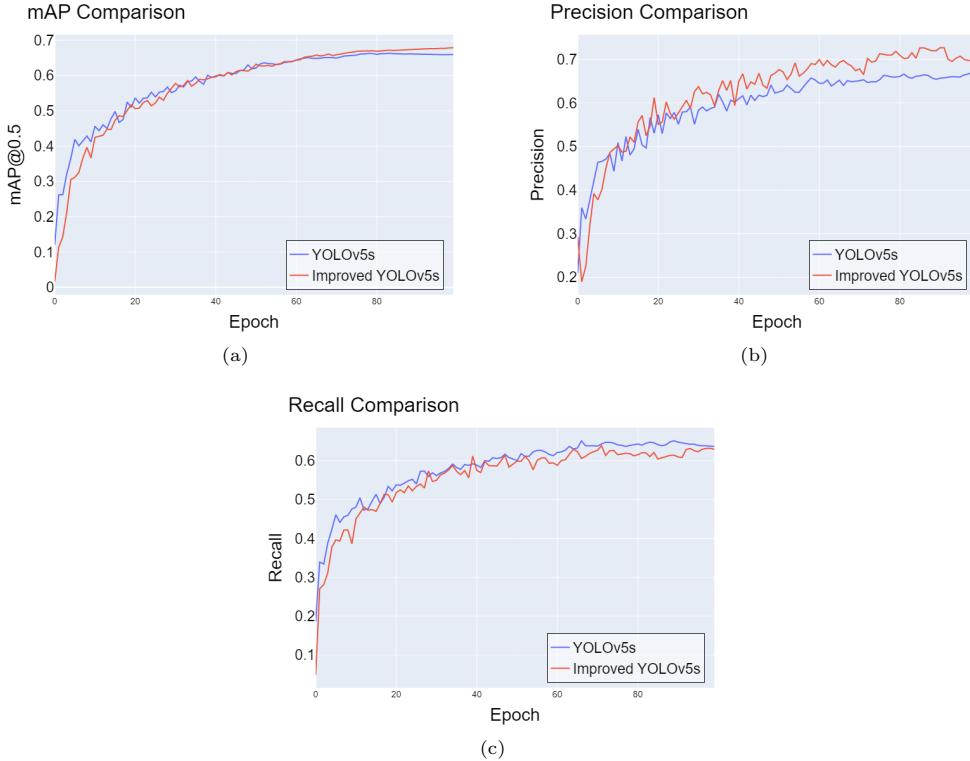


Fig. 11: Comparison of performance between YOLOv5s-baseline and our improved YOLOv5s (a)mAP0.5 Comparison (b)Precision Comparison (c) Recall Comparison

fewer parameters and 12 Gigabytes less GFlops. Figure 11 shows the Comparison of performance between the YOLOv5s-baseline and our improved YOLOv5s.

4.8 Comparison with existing methods regarding road damage detection and classification

Table 6 shows the drastic performance gap between our improved YOLOv5s and other object detection methods on the RDD 2018, RDD 2020, and RDD 2022 datasets. Singh et al. used Mask RCNN with the RDD 2018 dataset to acquire an F1 score of 52.8% [32]. Numerous studies utilized RDD 2020, the successor to RDD 2018, and produced a higher F1 score than previous research [16, 35]. Using the RDD 2020 dataset and other object detection algorithms, the researcher discovered that the YOLO-based approach outperformed most of the other studies [11]. The only literature we found using RDD 2022 got an average of 66.46% F1 score [26] with 28.46 million parameters. We exceeded their average F1 score by 0.05% with 20 million fewer parameters. Moreover, We outperformed all previous studies that used RDD 2018 and RDD 2020 datasets by a significant margin.

Table 6: Comparison with existing methods regarding road damage detection and classification.

Model	F1 Score(%)
Mask R-CNN with RDD 2018 [32]	52.80
Multi-stage Faster R-CNN with Resnet-101 backbones with RDD 2020 [35]	54.26
Ensemble(YOLO-v4+Faster-RCNN) with RDD 2020 [16]	57.07
YOLOv5x with RDD 2020 [11]	57.10
Improved YOLOv5s with RDD 2020 [6]	60.92
YOLOv7 with RDD 2022 [26]	66.30
YOLOv5s with RDD 2022 (ours)	65.22
YOLOv8s with RDD 2022 (ours)	66.46
Improved YOLOv5s with RDD 2022 (Ours proposed model)	66.51

5 Conclusion & Future Work

Rapid and accurate identification of road damage can tremendously benefit the road maintenance industry and significantly contribute to the economy. Applying deep learning techniques becomes crucial as it considerably facilitates road inspection and provides a comprehensive view of the road’s condition. The present study introduces an enhanced approach for detecting road damage utilizing the YOLOv5s model. The proposed method involves training the network model with an extensive amount of data to enable the detection of different kinds of road damage. Initially, our task involved the identification of an optimal backbone for our model that would exhibit excellent accuracy and low computational complexity. Following identifying the model’s backbone, we conducted experiments utilizing diverse attention mechanisms at various locations within the YOLOv5s architecture. Following adopting the ECA attention mechanism, we suggest several alterations that lead to a notable enhancement in the results. The focal loss function is utilized instead of the binary cross-entropy (BCE) loss function, while the K-means++ algorithm is employed to determine the optimal anchor box for our model. Subsequently, a label smoothing algorithm is used. Ultimately, a novel detection head was integrated into a new prediction module at the neck of the YOLOv5s framework. Even though the most common technique for detecting and classifying road damage is object detection, performing instance segmentation is possible and preferable. In the future, we will construct a model for segmenting and classifying front-view images of road damage using an instance segmentation algorithm. This method can substantially enhance the accuracy of the task and precisely pinpoint the location of the damage. Due to the lack of an instance segmentation-based dataset for road damage, this category must still be examined for improved results. Segmenting front-view images of these forms of damage is a complex and laborious process.

References

- [1] Arthur D, Vassilvitskii S (2007) K-means++ the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pp 1027–1035
- [2] Arya D, Maeda H, Ghosh SK, et al (2021) Rdd2020: An annotated image dataset for automatic road damage detection using deep learning. Data in brief 36:107133

- [3] Arya D, Maeda H, Ghosh SK, et al (2022) Rdd2022: A multi-national image dataset for automatic road damage detection. arXiv preprint arXiv:220908538
- [4] Chen Z, Zhang F, Liu H, et al (2023) Real-time detection algorithm of helmet and reflective vest based on improved yolov5. *Journal of Real-Time Image Processing* 20(1):4
- [5] Cord A, Chambon S (2012) Automatic road defect detection by textural pattern recognition based on adaboost. *Computer-Aided Civil and Infrastructure Engineering* 27(4):244–259
- [6] Guo G, Zhang Z (2022) Road damage detection algorithm for improved yolov5. *Scientific reports* 12(1):1–12
- [7] Hosseini SA, Smadi O (2021) How prediction accuracy can affect the decision-making process in pavement management system. *Infrastructures* 6(2):28
- [8] Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
- [9] Huang Z (1998) Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery* 2(3):283–304
- [10] Jeong D (2020) Road damage detection using yolo with smartphone images. In: 2020 IEEE International Conference on Big Data (Big Data), IEEE, pp 5602–5608
- [11] Jeong D (2020) Road damage detection using yolo with smartphone images. In: 2020 IEEE International Conference on Big Data (Big Data), IEEE
- [12] Li J, Zhao X, Li H (2019) Method for detecting road pavement damage based on deep learning. In: Health Monitoring of Structural and Biological Systems XIII, SPIE, pp 517–526
- [13] Li S, Cao Y, Cai H (2017) Automatic pavement-crack detection and segmentation based on steerable matched filtering and an active contour model. *Journal of Computing in Civil Engineering* 31(5):04017045
- [14] Lin TY, Maire M, Belongie S, et al (2014) Microsoft coco: Common objects in context. In: Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13, Springer, pp 740–755
- [15] Lin TY, Goyal P, Girshick R, et al (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
- [16] Liu Y, Zhang X, Zhang B, et al (2020) Deep network for road damage detection. In: 2020 IEEE International Conference on Big Data (Big Data), IEEE, pp 5572–5576

- [17] Maeda H, Sekimoto Y, Seto T, et al (2018) Road damage detection and classification using deep neural networks with smartphone images. *Computer-Aided Civil and Infrastructure Engineering* 33(12):1127–1141
- [18] Mandal V, Mussah AR, Adu-Gyamfi Y (2020) Deep learning frameworks for pavement distress classification: A comparative analysis. 2020 IEEE International Conference on Big Data (Big Data) pp 5577–5584
- [19] Miller TR, Zaloshnja E (2009) On a crash course: The dangers and health costs of deficient roadways. *Injury Prevention* 15(4):245–248
- [20] Müller R, Kornblith S, Hinton GE (2019) When does label smoothing help? *Advances in neural information processing systems* 32
- [21] Naddaf-Sh MM, Hosseini S, Zhang J, et al (2019) Real-time road crack mapping using an optimized convolutional neural network. *Complexity* 2019(1):1–17
- [22] Naddaf-Sh S, Naddaf-Sh MM, Kashani AR, et al (2020) An efficient and scalable deep learning approach for road damage detection. 2020 IEEE International Conference on Big Data (Big Data) pp 5602–5608
- [23] Nguyen HT, Nguyen LT, Sidorov DN (2016) A robust approach for road pavement defects detection and classification. *Journal of Computational and Engineering Mathematics* 3(3):40–52
- [24] Nguyen TS, Begot S, Duculty F, et al (2011) Free-form anisotropy: A new method for crack detection on pavement surface images. 2011 18th IEEE International Conference on Image Processing pp 1069–1072
- [25] Oliveira H, Correia PL (2014) Crackit—an image processing toolbox for crack detection and characterization. In: 2014 IEEE international conference on image processing (ICIP), IEEE, pp 798–802
- [26] Pham V, Nguyen D, Donan C (2022) Road damages detection and classification with yolov7. arXiv preprint arXiv:221100091
- [27] Radopoulou SC, Brilakis I (2015) Detection of multiple road defects for pavement condition assessment. *Transportation Research Record: Journal of the Transportation Research Board* 2486(1):101–109
- [28] Redmon J, Divvala S, Girshick R, et al (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
- [29] Shim S, Cho GC (2020) Lightweight semantic segmentation for road-surface damage recognition based on multiscale learning. *IEEE Access* 8:102680–102690

- [30] Shim S, Kim J, Lee SW, et al (2021) Road surface damage detection based on hierarchical architecture using a lightweight auto-encoder network. *Automation in Construction* 130:103833
- [31] Shim S, Kim J, Lee SW, et al (2022) Road damage detection using super-resolution and semi-supervised learning with a generative adversarial network. *Automation in Construction* 135:104139
- [32] Singh J, Shekhar S (2018) Road damage detection and classification in smart-phone captured images using mask r-cnn. *arXiv preprint arXiv:181104535*
- [33] Torbaghan ME, Li W, Metje N, et al (2020) Automated detection of cracks in roads using ground penetrating radar. *Journal of Applied Geophysics* 179:104118
- [34] Torres-Mach'i C, Chamorro A, Pellicer E, et al (2015) Sustainable pavement management: Integrating economic, technical, and environmental aspects in decision making. *Transportation Research Record* 2523(1):56–63
- [35] Vishwakarma R, Vennelakanti R (2020) CNN model & tuning for global road damage detection. In: 2020 IEEE International Conference on Big Data (Big Data), IEEE
- [36] Wan F, Sun C, He H, et al (2022) Yolo-lrdd: a lightweight method for road damage detection based on improved yolov5s. *EURASIP Journal on Advances in Signal Processing* 2022(1):98
- [37] Wang Q, Wu B, Zhu P, et al (2020) Eca-net: Efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11534–11542
- [38] Wang W, Wu B, Yang S, et al (2018) Road damage detection and classification with faster r-cnn. In: 2018 IEEE International Conference on Big Data (Big Data), IEEE, pp 5220–5223
- [39] Wang Y, Song K, Liu J, et al (2021) Renet: Rectangular convolution pyramid and edge enhancement network for salient object detection of pavement cracks. *Measurement* 170:108698
- [40] Wu J, Dong J, Nie W, et al (2023) A lightweight yolov5 optimization of coordinate attention. *Applied Sciences* 13(3):1746
- [41] Wu S, Fang J, Zheng X, et al (2019) Sample and structure-guided network for road crack detection. *IEEE Access* 7:130032–130043
- [42] Wu X, Sahoo D, Hoi SC (2020) Recent advances in deep learning for object detection. *Neurocomputing* 396:39–64

- [43] Xu Y, Xu Y, Qian Q, et al (2020) Towards understanding label smoothing. arXiv preprint arXiv:200611653
- [44] Yusof NAM, Ibrahim A, Noor MHM, et al (2019) Deep convolution neural network for crack detection on asphalt pavement. Journal of Physics: Conference Series 1349(1):012020
- [45] Zhang A, Wang KC, Li B, et al (2017) Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network. Computer-Aided Civil and Infrastructure Engineering 32(10):805–819
- [46] Zhang A, Wang KC, Li B, et al (2017) Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network. Computer-Aided Civil and Infrastructure Engineering 32(10):805–819