No Author Given

LOC Page

Contents

CHAPTER 1

Road Damage Detection and Classification using YOLOv7

Abdullah As Sami

Chittagong University of Engineering & Technology, Dept. of CSE, Chittagong, Bangladesh.

Saadman Sakib

Chittagong University of Engineering & Technology, Dept. of CSE, Chittagong, Bangladesh.

Kaushik Deb

Chittagong University of Engineering & Technology, Dept. of CSE, Chittagong, Bangladesh.

CONTENTS

Abstract

Deep learning has allowed a simple, convenient method of road infrastructure management to facilitate a safe, cost-effective, and efficient transportation network. Timely repairing roads becomes challenging as manual road inspection is time-consuming and unsafe. Recent advancements in deep learning allow the automatic collection and assessment of road damage data. This paper describes the YOLOv7 object detection model trained on the most recent benchmark Road Damage Dataset, RDD 2022. Few studies examined the efficacy of the new state-of-the-art RDD 2022, which comprised six country's road damage, including Japan, Norway, The USA, Czech, and China, with four common categories of damage, i.e., verticle cracks, horizontal cracks, alligator cracks, and pathhole. We examined different object detection models and concluded that the object detection model YOLOv7 could be the benchmark model for road damage detection and classification. Firstly, we added the Convolutional Block Attention Module (CBAM) module to the network. Secondly, the K-means++ algorithm was employed to determine a suitable anchor box for our dataset. The proposed YOLOv7 exceeded all previous object detection approaches on road damage detection and classification tasks, obtaining 68.61% mAP and a 66.87% F1 score.

Keywords: road damage detection, attention, YOLOv7, K-means++, RDD 2022.

1.1 INTRODUCTION

Modern civilizations are heavily dependent on the transportation of people and goods via roadways, which imposes a continuing degradation on the road surface and demands a robust road maintenance system. Current methods for assessing road damage fall into three categories: manual assessment, automatic assessment, and image processing. Traditional manual assessment requires an excessive workforce, material resources, and time. However, in the real world, the manual process is tedious due to the length of the testing roads and the high resource demands of the task. Road assessment using automatic detection systems is rising with the advancement of technology, for instance, using vehicles fitted with sensor equipment [1]. Since the road condition is complex, it becomes challenging for automatic assessment tools. Image processing techniques can incorporate high effectiveness and low cost. Road surroundings are complex, so manual feature extraction is impossible with conventional image-processing methods. Compared to usual image processing approaches, image processing techniques using deep learning are broadly employed for road damage detection because of their superior accuracy, speed, and embeddability [4]. Deep learning technology has achieved substantial progress in detecting road damage. Naddaf et al. [5] utilized the Faster R-CNN to detect road damage and assessed the effect of various lighting and weather conditions. Mandal et al. [6] proposed to identify road damage using the one-stage YOLO CSPDarknet53 network. It showed another model exploration using the road damage dataset. Despite the mentioned research's progress in the road damage detection problem, the substantial potential for advancement in accuracy remains. We selected The YOLOv7 model as the approach to investigate the Road Damage Detection and Classification problem due to its demonstrated high accuracy and satisfactory Frame per Second (FPS) performance. The contributions to this work can be summarized as follows:

- Evaluating current state-of-the-art object detection methods, their relevance, and associated road damage detection and classification techniques with the most recent benchmark dataset for road damage detection, RDD 2022.

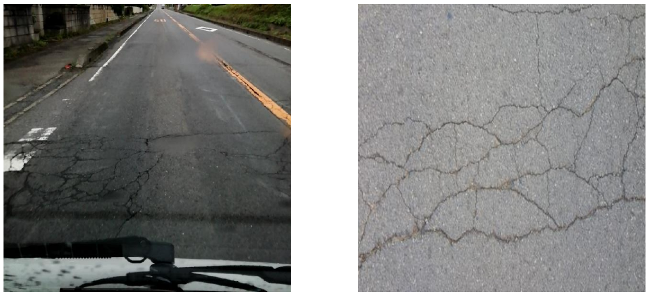
- The integration of the Convolutional Block Attention Module (CBAM) into the network of the model aimed to augment the accuracy of the model.
- The K-means++ algorithm was employed to determine a suitable anchor box for our dataset, as opposed to using the default anchor box based on the COCO dataset.

1.2 RELATED WORK

Researchers used two distinct image types for the road damage detection study. Figure 1.1 shows a snapped photo of the top view, while the other is of the front perspective. 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 focused on top-view damage have great detection accuracy. However, the majority of these models feature only crack damage. Most of the studies did not attempt to categorize various road damage. Yusof et al. [7] fine-tuned CNN's filter sizes to achieve satisfactory accuracy of 94.5% in identifying road photographs of different cracks. Zhang et al. [8] constructed a CrackNet model devoid of a pooling layer to recognize cracks in the road with a 90.1% degree of accuracy. This model performed admirably, achieving 98.00% precision and 97.92% accuracy when categorizing different crack damages. Though the models mentioned above achieve an astounding detection performance, the application of these models is limited. Since the operation of collecting these types of road images using dedicated cars regularly is costly.

The road damage detection model using front-view photos concentrates on developing detection and classification models using images taken by dashboard-mounted cameras. This process of recognizing road damage is more suitable and practical in the actual world to reduce human intervention and extensive research is also being conducted in this area. Nonetheless, the process is more difficult because this type of image has a variety of complicated and obscure damage. In addition to the road surface, the background of these photographs includes atmosphere, geography, and other noise that makes road damage detection more difficult. However, using these images efficiently to detect damage would provide immense advantages in road supervision tasks. The databases of front-view images are currently vast. Moreover, it can be increased quickly and easily. In addition, images covering complete road areas in inspections and camera installation are simple and inexpensive. Due to images' diverse and complicated textures, studies into leveraging this image resource stay narrow despite its numerous benefits. Deep learning-based models may effectively address these challenging requirements. However, only a few models have their performance evaluated using images taken by dashboard-mounted cameras. Researchers use object detection models such as Faster RCNN, YOLO, and SSD to detect and classify road damage. Recently, Maeda et al. [9] proposed combining a Single shot detector(SSD) with an Inception backbone and SSD with a MobileNet backbone for detection and categorizing eight types of road damage, reaching a 71% recall score. Jeong et al. [10] proposed a YOLOv5x-based model, and their model achieved a 67% F1 score.



(a) Front view image (b) Top view image

Figure 1.1: Different types of images used in road damage detection research: (a) Front view image; (b) Top view image.

However, the detection speed could be more satisfactory. Wang et al. [11] detected and classified road damage using a faster R-CNN- model with data augmentation approaches and achieved a 62.5% F1 score.

1.3 METHODLOLOGY

In this section, we described brief overview of proposed Road Damage Detection and Classification architecture.

1.3.1 Dataset description and Collection

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 employed a benchmark dataset called Road Damage Dataset (RDD 2022) to train the road damage detection model. Most top-view datasets lack the class name; therefore, classification is somewhat challenging. 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 [13], and RDD 2018 [9] 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 [12] 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 1.2 depicts the distributions of damage types, i.e., 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.

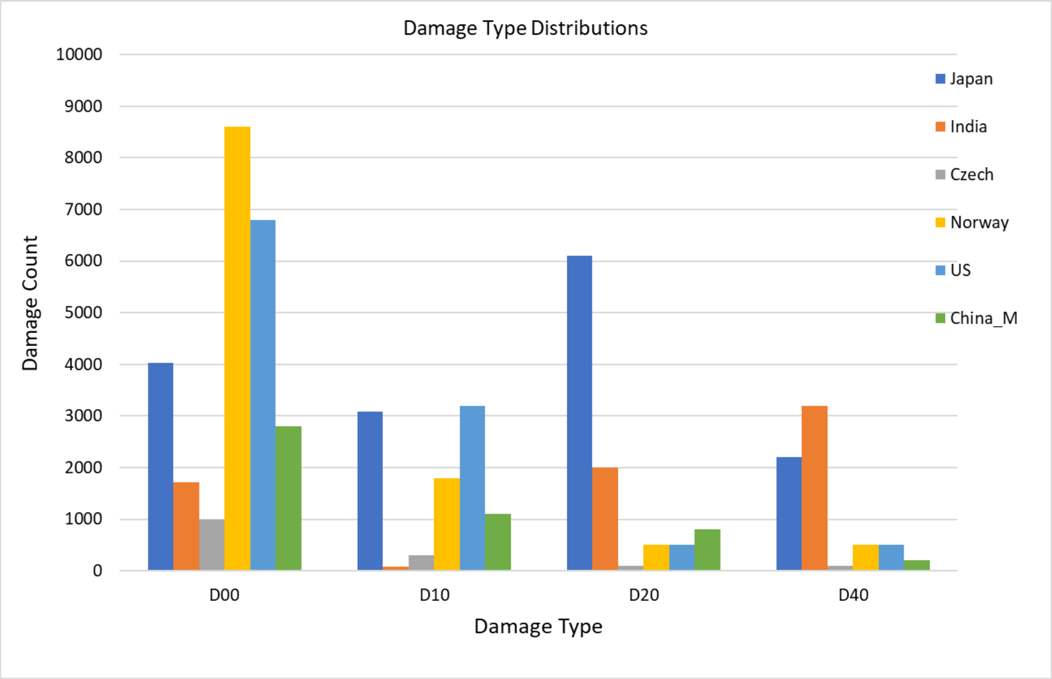
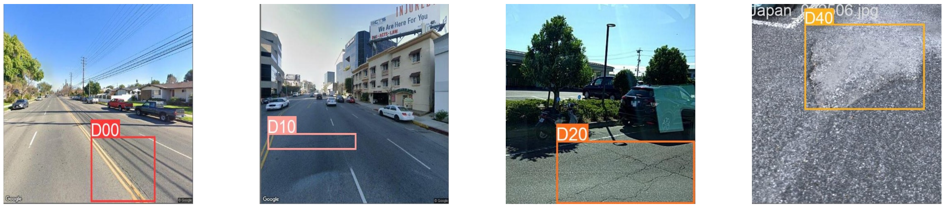


Figure 1.2: Damage category distribution in the current benchmark state-of-the-mark dataset, RDD 2022.

1.3.2 Data Processing

The RDD 2022 dataset has 36,000 photos with an annotation file, most of which have unique labels for a single country. 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 1.3 depicts an example of these four classes. Additionally, there are a large number of photos that do not have any annotation. These images are pointless for the road damage detection model, and hence we 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. For example, the resolution of Norway photos 3072 × 3720 is incompatible with our GPU configuration. Moreover, we omitted China-Drone in our proposed work because it needed to be obtained from a front-mounted vehicle image. We manually scale each image to 640 × 640 to train our YOLOv7 model. The greater efficiency of YOLO with a 640 × 640 image can be linked to the algorithm's built-in design. The YOLO algorithm partitions an image with dimensions of 640 x 640 into a grid consisting of 20 x 20 cells, where each cell measures 32 x 32 pixels. This particular grid dimension facilitates an optimal equilibrium in detecting objects of varying sizes. If the image size is smaller, there will be fewer cells in the grid, making it easier to detect small objects. Conversely, if the image's dimensions are massive, the grid will contain a more significant number of cells, thereby rendering the image



(a) D00:longitudinal Crack

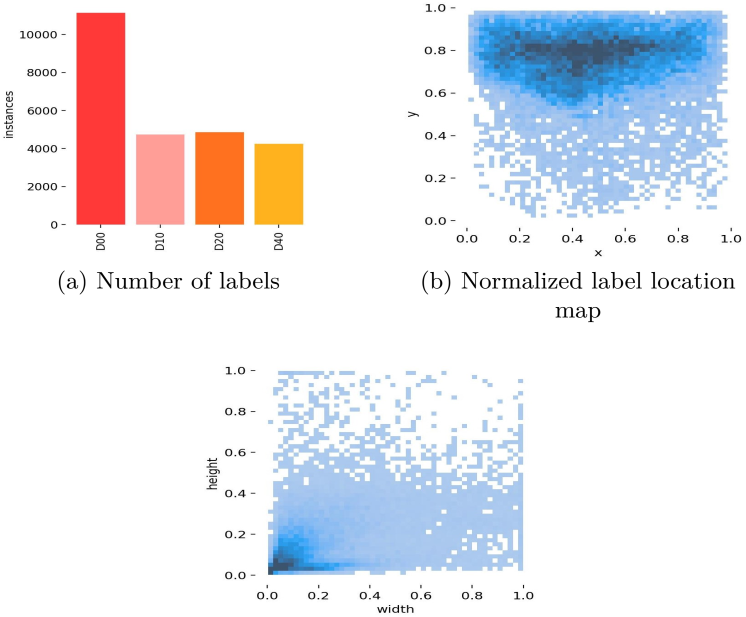
(b) D10:Transverse Crack

(c) D20:Alligator Crack

(d) D40:Pathhole

Figure 1.3: Different kinds of road damages used in the proposed work.

processing computationally tricky. The image size of 640 x 640 has been determined to be effective in facilitating the efficient and precise detection of objects of diverse sizes within an image by YOLO [14]. Finally, we partitioned the dataset into three distinct subsets, namely training, validation, and testing, in a ratio of 70:20:10. The training set comprised 13039 images, the validation set contained 3619 images, and the testing set was composed of 1842 images. The testing set is sufficient for measuring the model's performance on unseen data. In addition, this research requires a large validation set because the dataset needs to be more balanced, with fewer samples in some classes than in others.



(a) Number of labels

(b) Normalized label location map

(c) Normalized label size

Figure 1.4: The statistical results of the RDD 2022 dataset

Figure 1.4a depicts a graph where the vertical axis represents the number of labels, while the horizontal axis represents the names of the labels. The distribution of labels is illustrated in figure 1.4b. The abscissa, denoted as x, represents the ratio between the label center's abscissa and the image's width. Similarly, the coordinate y represents the ratio between the label center's abscissa and the image's height. Figure 1.4c illustrates the abscissa width as the ratio between the label and image widths. At the same time, the coordinate height represents the ratio between the label height and the image height.

1.3.3 Architecture Description

The Yolov7 network structure comprises four distinct modules: the input terminal, backbone, neck, and head, alongside several other components like CBS (convolution, batch normalization, and Sigmoid Linear Unit layer for feature extraction), Efficient Layer Aggregation Networks (ELAN), MP (Composition of max pool and CBS layer), and Spatial Pyramid Pooling layer with CSP net (SPPCSPC). Figure 1.6 represents the whole Yolov7 architecture. The YOLOv7 network initiated image pre-processing, then resizing the image to 640 × 640 × 3 before feeding it into the backbone network.

The CBS module, ELAN (efficient layer aggregation networks) module, and MP module sequentially downsized the feature map by a factor of 1/2 in length and width dimensions while doubling the number of output channels relative to the number of input channels. In addition, the CBS composite module executed a sequence of convolution, batch normalization, and activation function operations on the input feature map. The SiLU activation function was employed for this purpose. The Equation of the SiLU activation function is given in equation 1.1:

$$\mathrm{SiLU}(x) = \frac{x}{1+e^{-x}} \tag{1.1}$$

MP module comprises two components: the CBS module and the maximum pooling. In addition, the ELAN and ELANW modules are the primary computational components of YOLOv7, assigned with extracting features. The SPPCSPC module can acquire object information at multiple scales while maintaining the feature map size. The Repconv structure involves re-parameterization, extending the training period and enhancing the inference outcome [23].

The ELAN module is utilized to expand, shuffle and merge cardinality to enhance the model's learning capacity while preserving the initial gradient path. The ELAN architecture consisted of various convolutions. The utilization of group convolution is employed to expand the computational block's channel and cardinality within its architecture. In addition, various groups of computational units are directed towards acquiring a more comprehensive range of features. If the ELAN/W module contains two N connections, it is called ELANW. Otherwise, it is defined as ELAN. The main feature extraction task in YOLOv7 is accomplished using four ELAN modules. After adjusting the number of channels, three ELAN modules are directed towards the Neck region. The Neck component of YOLOv7 utilizes a feature pyramid structure, incorporating one SPPCSPC module and four ELANW modules to perform feature extraction. The three ELANW modules on the right side are directly connected to the Head part. Ultimately, the network executes feature output via three detection

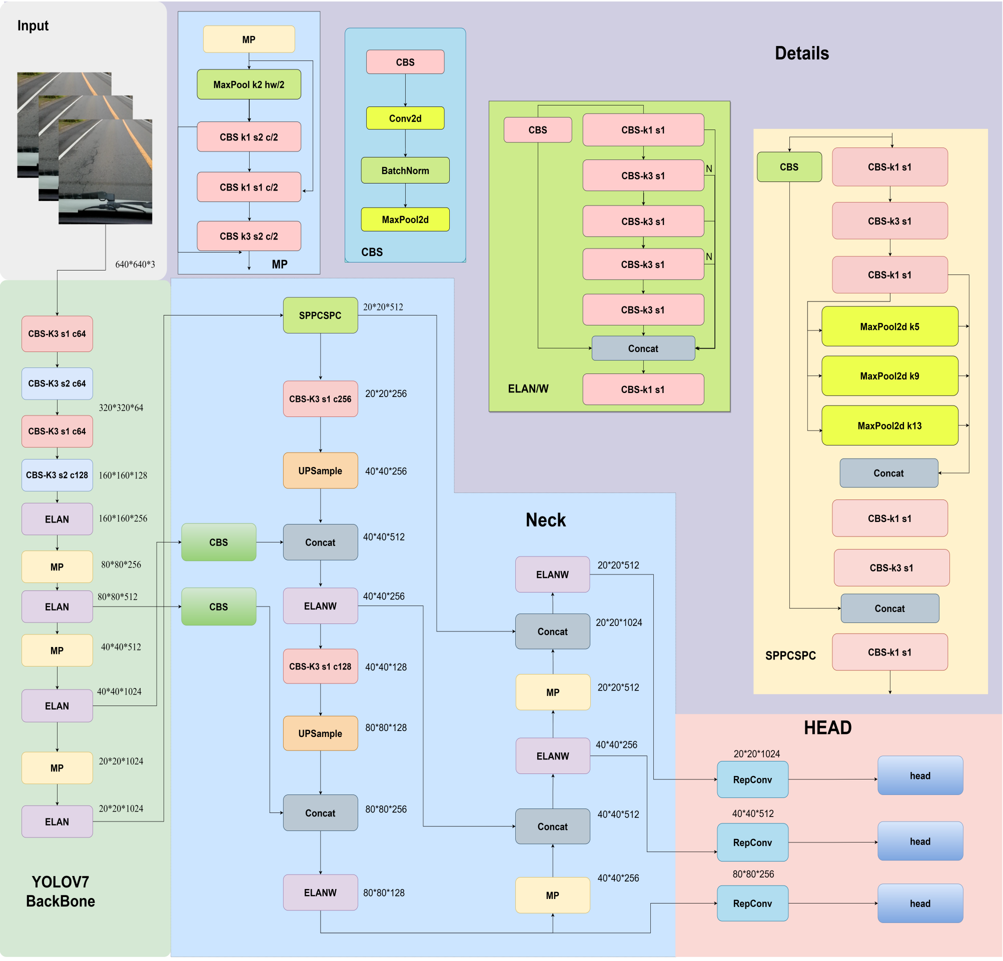


Figure 1.5: A brief overview of YOLOv7 Architecture.

heads. Consequently, the output comprises three distinct dimensions of feature maps measuring 20×20, 40×40, and 80×80. After the convolution process, the output feature maps are transformed into a one-dimensional vector called the fully connected layer. This one-dimensional vector can be used to predict the targets in the image.

The YOLOv7 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 1.2 represents the full loss function.

$$\text{LOSS} = a \times L_{\text{obj}} + b \times L_{\text{cls}} + c \times L_{\text{box}} \tag{1.2}$$

The weighting factors denoted by a, b, and c correspond to the three partial losses. 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.

Equation 1.3 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 1.4.

$$L_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \tag{1.3}$$

$$L_{\text{loc}}(l, g) = 1 - \text{CIoU}(l, g) = 1 - \text{IoU}(l, g) + \frac{\rho(l, g)^2}{c^2} + \alpha v \tag{1.4}$$

$$\text{IoU}(l, g) = \frac{l \cap g}{l \cup g} \tag{1.5}$$

$$\nu = \frac{4}{\pi^2}\left(\tan^{-1}\left(\frac{g_w}{g_h}\right) - \tan^{-1}\left(\frac{l_w}{l_h}\right)\right)^2 \tag{1.6}$$

$$\alpha = \frac{\nu}{(1 - \text{IoU}(l, g)) + \nu} \tag{1.7}$$

The symbols l and g in equation 1.4 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 two boxes is represented by equation 1.5 of IoU. v in equation 1.6 denotes the aspect ratio's consistency. Finally, the symbol α, denoted in equation 1.7, represents the trade-off parameter.

1.3.3.1 Refining Detection through Attention Module

In the context of deep learning and neural networks, attention refers to a sophisticated mechanism inspired by the human cognitive process. It involves selectively focusing on specific elements or regions of input data while allocating varying degrees of importance to different parts based on their relevance to the task.

We comprehensively analyzed four attention modules to fulfill our objective by incorporating attention in several network positions. Our findings show that adding an attention module to the model's backbone increases its weight since each ELAN block contains many parameters. The YOLOv7 network structure was enhanced by including the CBAM attention mechanism [23], as depicted in figure 1.7. The strategic positioning of this element contributes to improved network performance by allowing for targeted attention to the most relevant characteristics. citewu2023lightweight.

CBAM, also known as the Convolutional Block Attention Module, is an attention-driven mechanism meticulously crafted to augment the representational prowess of CNN. A discerning eye selectively accentuates salient features across spatial and channel dimensions, empowering the network to extract and encapsulate pivotal information more effectively. By strategically highlighting the most significant aspects within the input data, CBAM drives CNN's capacity to determine complex patterns and gain deeper insights. The Channel Attention Module in equation 1.8 of CBAM

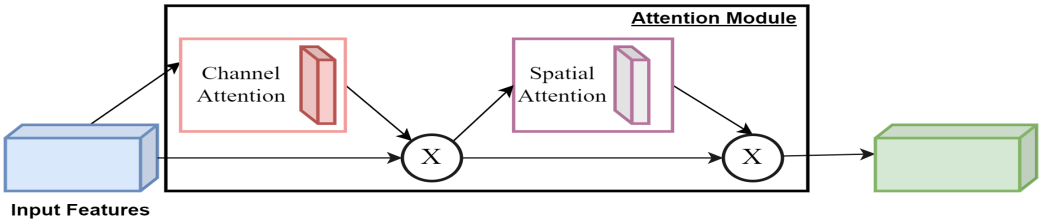


Figure 1.6: A overview of Attention Module Architecture.

is designed to capture the intricate interdependencies among channels within feature maps meticulously. By selectively assigning weights to different channels, the module enables the network to focus more on significant channels while downplaying the influence of less informative ones. On the other hand, the Spatial Attention Module in equation 1.9 within CBAM is engineered to capture the complex relationships among various spatial locations within feature maps. This module empowers the network to selectively amplify or suppress specific spatial locations based on their relevance.
Thus, the Channel and Spatial Attention Module of CBAM in equation 1.10 form a powerful attention-based mechanism that enhances the representation power of CNN. By adaptively adjusting channel weights and selectively amplifying or suppressing spatial locations, CBAM allows the network to selectively focus on crucial information and improve its ability to extract meaningful features from complex data.

$$\omega_c = \sigma(C_w \cdot F \tag{1.8}$$

$$omega_s = \sigma(GP(F) \tag{1.9}$$

$$F_{\text{cbam}} = F \cdot \omega_c \cdot \omega_s \tag{1.10}$$

The symbols C_w and F in equation 1.8 weight matrix of the channel attention module and input feature map respectively. In equation 1.9 $GP(F)$ global average pooling of F.

1.3.3.2 Refined Anchor Generation

The anchor box's initial dimensions in YOLOv7 are determined by employing the K-means algorithm [24] for edge clustering on the MS COCO dataset [25]. Nevertheless, it is worth noting that the MS COCO dataset primarily consists of objects classified as large or medium-sized. In contrast, this study's Road Damage Detection dataset includes smaller and medium-sized targets, which differ from the objects present in the COCO dataset. Therefore, it is determined that the initial dimensions of the anchor box in YOLOv7 are not appropriate for the samples in the RDD-2022 dataset. To tackle this matter, the K-means++ clustering algorithm was utilized as a substitute for the K-means algorithm to readjust the dimensions of the anchor box for the samples contained within the RDD 2022 dataset. K-means++ is a modified version of the K-means clustering algorithm that aims to overcome specific limitations inherent in the original K-means algorithm [26]. The limits encompass random initialization of centroids, slower convergence speed, and diminished accuracy.

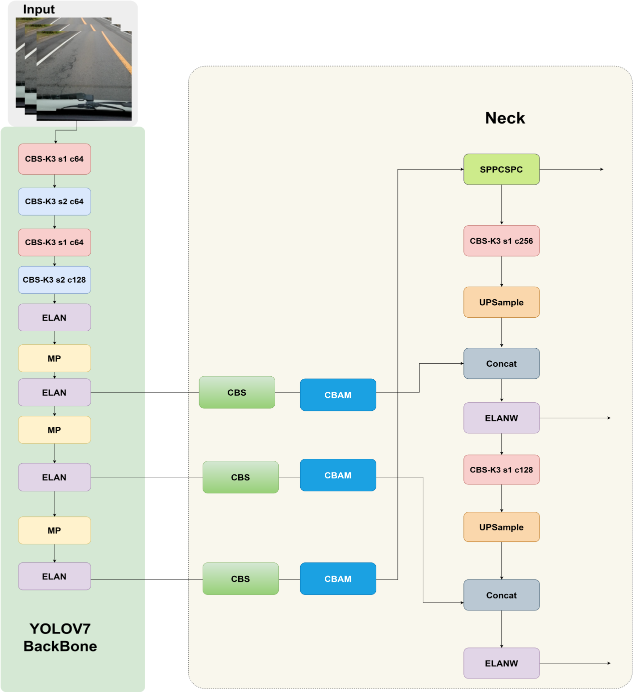


Figure 1.7: Addition of CBAM Attention in YOLOv7 Network Architecture.

To effectively align the anchor box dimensions in YOLOv7 with the characteristics of the RDD 2022 dataset samples, it became evident that the initial measurements needed to be better suited. By embracing the inherent superiority of K-means++, we tackled an iterative process, meticulously recalibrating the anchor box dimensions to seamlessly harmonize with the intricacies exhibited by the samples within the RDD-2022 dataset. K-means++ algorithm can be summarized as follows:

1. Randomly select the first centroid from the dataset.

2. For each remaining centroid (2 to K):
   a. Calculate the minimum squared distance (D_i) from each data point to the nearest centroid, considering all existing centroids up to (i-1).
   b. Select the next centroid by sampling a data point with a probability directly proportional to its corresponding minimum squared distance (D_i).

3. Return the set of K centroids.

1.3.4 Model Training

We trained the RDD 2022 dataset with a few object detection models, such as one-stage object detectors like YOLO and two-stage object detectors like Faster RCNN. We have trained a few recent versions of the YOLO, the YOLOv5, YOLOv6 and the YOLOv7. YOLOv7 has several distinct versions, including YOLOv7tiny, YOLOv7, YOLOv7x, YOLOv7-W6, YOLOv7E6,YOLOv7D6, and YOLOv7E6E. YOLOv7Tiny is the smallest model, whereas YOLOv7E6E is the largest. We proposed YOLOv7 as our model as it outperforms all other object detection models.

We utilized YOLOv7's default image enhancement settings, contributing to greater accuracy. The hyper-parameters can be tuned during YOLO model training. Table 1.1 describes the YOLOv7 image augmentation settings during a training epoch. We trained all models using 16 GB Tesla T4 GPU and 32 GB RAM.

Table 1.1: Image augmentation parameter of the YOLOv7 architecture.

Augmentation parameter	**Value**
Image hsv - hue	0.01
hsv saturation	0.70
hsv value	0.40
Rotation degree	0.00
Translation	0.10
scaling	0.50
Flip left right	0.50
Mosaic Augmentation	1.00

1.4 EVALUATION PARAMETER

Precision, recall rate, F1 score, mean Average Precision (mAP), the number of parameters, and GFlops(One billion Floating-Point Operations per second) are employed as evaluation metrics in this research. The model precision score measures the proportion of positively predicted labels that were successfully predicted. The equation of the precision is presented in equation 1.11.

$$Precision\ Score = \frac{TP}{(FP + TP)} \tag{1.11}$$

In contrast, Recall represents the model's capability to predict positive results based on actual positive results accurately. True positive(TP), False positive(FP), and False negative(FN) are utilized to calculate precision and recall. The Recall equation is provided in equation 1.12.

$$Recall\ Score = \frac{TP}{(FN + TP)} \tag{1.12}$$

The F1 score represents the model's performance based on the Recall and precision scores. The F1 score is an option for Performance measures that give Precision and Recall equal weight while evaluating the performance of a machine learning model. This can be stated technically as a harmonic mean of the precision and recall scores. The F1 score equation is given in equation 1.13.

$$F1\ Score = \frac{2 * Precision\ Score * Recall\ Score}{(Precision\ Score + Recall\ Score)} \tag{1.13}$$

Mean Average Precision(mAP) is a metric employed to assess object detection models. First, the accuracy and recall values for each detection in the class are computed for a range of confidence threshold values. The area under that class's precision-recall curve (AUC) is then used to determine the average Precision. After calculating the AP score for each class, we construct the mAP score by averaging these scores. The mAP score is a valuable aggregate metric that comprehensively assesses the model's detection performance across all classes in the data set. The equation of mAP is given in equation 1.3.

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \tag{1.14}$$

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

GFLOPs are utilized to evaluate the model or algorithm's efficiency. In general, the lower the GFLOPs, the less computational power it requires to depict the model, the lesser the performance requirements for hardware, and the simpler it is to implement in low-end devices. Another crucial indicator for evaluating a model's complexity and computational demands is the number of parameters. A more significant number of parameters indicates a model's complexity, which may involve more computational resources for training and deployment. Therefore, while evaluating object detection models, examining the number of parameters and the model's effectiveness on relevant metrics such as mAP, Precision, and Recall is essential.

1.5 EXPERIMENTAL RESULT

In this section, we evaluate the effectiveness of our proposed method through a series of experiments.

1.5.1 Comparison among different object detectors

We trained nine different models to validate the efficacy of our proposed suggested YOLOv7 model. We conducted experiments with five distinct models to choose the best model, including three of the most well-known one-stage detectors, YOLOv5, YOLOv6, and YOLOv7. In addition, we implemented a two-stage detector model based on faster R-CNN with two different backbones. Table 1.2 displays the different object detection model's performance on the RDD 2022 dataset. The YOLOv7-tiny version achieves a satisfactory F1 score of 63.68% with less computing. It exhibits the highest frames per second (FPS), while the Faster RCNN with Resnet 101 demonstrates the lowest FPS. The YOLOv5m attains 68.45% mAP while showing a 44FPS. YOLOv5-large and YOLOv6-large version shows more excellent performance of 70.49% and 69.13% mAP; however, the FPS is low. we assessed various YOLOv7 model versions, including YOLOv7-tiny, YOLOv7 ,YOLOv7-X, and YOLOv7-W6. We omitted YOLOv7E6, YOLOv7E6E, and YOLOv7D6 from our experiment since they require many parameters and have greater GFLOPS, which are inappropriate for an embedded system with a dashboard-mounted camera. Ensuring a high frame rate (FPS) is imperative for this particular application while maintaining a notable

Table 1.2: Comparison of the YOLOv7 architecture with the other methods on detection results.

Model	**Precision**	**Recall**	**F1 Score**	**mAP0.5(%)**	**FPS(G)**
Faster R-CNN ResNet-50	69.78	60.25	64.66	66.14	18
Faster R-CNN ResNet-101	70.16	61.26	65.40	68.91	13
YOLOv5-medium	69.55	64.23	66.78	68.45	44
YOLO v5-large	72.34	63.84	67.82	70.49	36
YOLO v6-large	71.26	62.56	66.62	69.13	39
YOLOv7-tiny	66.51	61.14	63.68	65.67	**78**
YOLOv7	70.49	62.87	66.48	67.90	53
YOLOv7-X	71.83	62.92	67.11	69.41	33
YOLO v7-W6	72.55	64.95	**68.53**	**70.81**	29

accuracy level. Yolov7 depicts a good trade-off between speed and performance, containing 67.90% mAP and 53 FPS. That's why we selected YOLov7 for our task and moved forward with further experiments.

1.5.2 Experiment with Different Attention Mechanisms

The study demonstrated that the SimAM(A Simple, Parameter-Free Attention Module) requires fewer parameters and GFlops than the other three attention modules, with 33.17 million parameters. Table 1.8 illustrates the experimental data. The utilization of the Squeeze and Excitation module(SE) resulted in a notable improvement of 33.47% in mean average precision (mAP) and 66.53% in F1 score while maintaining a parameter count of 33.47 million. The SimAM model achieves a precision rate of 71.56% while exhibiting a lower recall rate of 62.14%, despite having a similar parameter count compared to the Yolov7 model. The Coordinate Attention (CA) model achieves a higher mean average precision (mAP) of 68.13% compared to the SimAM model, despite the CA model only having an increment of 0.4 million parameters. The Convolutional Block Attention Module (CBAM) yielded a higher result, yielding a mean average precision (mAP) of 68.49% with 33.83 million parameters. Hence, the CBAM attention module is the most suitable option based on its superior performance to alternative attention modules.

Table 1.3: Experimental result of different attention mechanisms with YOLOv7 architecture

Attention Mechanism	**Precision(%)**	**Recall(%)**	**F1 Score(%)**	**mAP@0.5(%)**	**No. of Params.(M)**	**GFlops**
SE	70.39	63.08	66.53	67.98	33.47	39.63
SimAM	71.56	62.14	66.51	68.24	33.17	39.35
CA	70.67	63.07	66.65	68.13	33.58	39.74
CBAM	70.85	**63.23**	**66.82**	**68.49**	**33.83**	**40.36**

1.5.3 Ablation Experiment

To evaluate the dependability and significance of individual enhanced modules within the model, we use YOLOv7 as a benchmark and execute ablation experiments while gradually integrating improved modules. This investigation's evaluation criteria in-

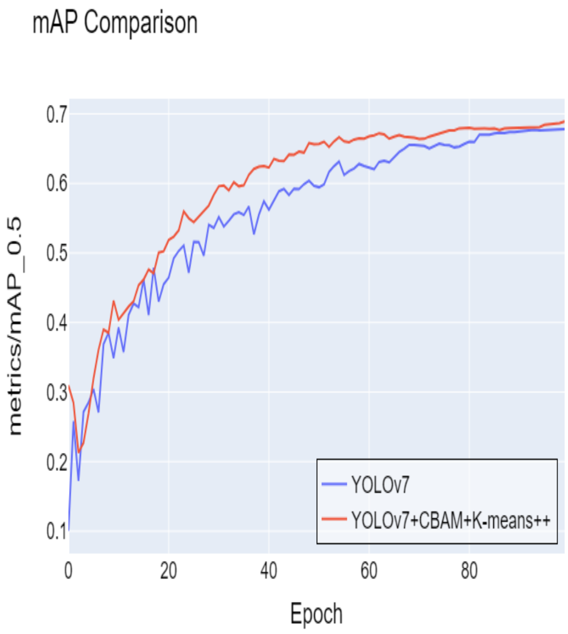


Figure 1.8: Comparison of mAP0.5 between YOLOv7-baseline and our proposed YOLOv7

clude Precision, Recall, F1 Score, mAP, Number of parameters, and frames per second (FPS). The results of the experiment are shown in Table 1.4. The default implementation of YOLOv7 has a mAP of 67.93% and 33.14 parameters. Including CBAM in the architecture has increased mAP by 0.82 %. Moreover, there was an increase of 0.7 million parameters and a decrease of 2 frames per second. Incorporating an adaptive anchor box using the K-means++ algorithm increased mAP by 0.17% and precision by 0.64%. The 110,000 reduction in parameters increased by 7 FPS. Figure 1.10 demonstrates our proposed model's detection results. Figure 1.8 shows the Comparison of mAP@0.5 performance between the YOLOv7-baseline and our proposed YOLOv7.Figure 1.9 depicts the precision-recall curve.

Table 1.4: Comparison of performance among different models

Model	**Precision (%)**	**Recall (%)**	**F1 Score (%)**	**mAP@0.5 (%)**	**No. of Params.(M)**	**FPS**
YOLOv7 Baseline	70.59	62.89	66.49	67.93	33.14	53
YOLOv7 Baseline + CBAM	70.85	63.23	66.82	68.49	33.83	51
YOLOv7 Baseline+ CBAM+K-means++	71.31	62.96	66.87	68.61	33.45	56

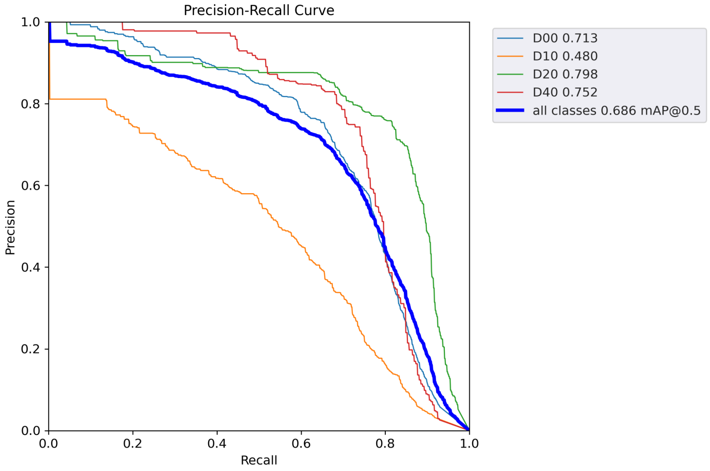


Figure 1.9: Precision-Recall Curve of YOLOv7.

Table 1.5: Results of each road damage classes of our proposed YOLOv7 model

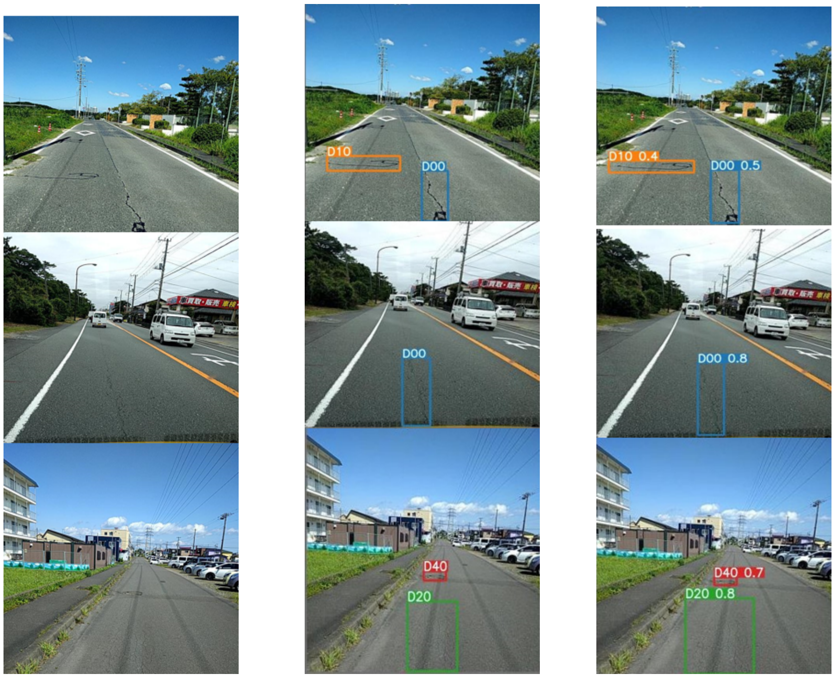
Class	Precision(%)	Recall(%)	F1 Score(%)	mAP@0.5(%)	mAP@0.5-0.95(%)
D00	69.82	68.12	68.94	71.30	35.37
D10	58.95	43.81	50.26	48.33	19.8
D20	77.13	77.26	77.19	79.62	45.57
D40	60.48	76.51	67.55	75.51	39.67

1.5.4 Comparison of different classes of the research

Table 1.5 shows the performance of each class. We observed that the D20 road damage category showed the highest performance of 79.62% mAP. The existence of D10 in the dataset is comparatively lower. That is why we got that impact on achieving the lowest score among all other classes, comprised of 48.33% mAP. The D40 is one of the complicated damage types that are also immense in number in the dataset, and we obtained the second-highest percentage of mAP among all other damage at 75.21% mAP. In addition, we received an mAP score of 71.30% in the D00 class, which ranked third among all other classes.

Table 1.6: Comparison with existing methods regarding road damage detection and classification.

Model	F1 Score (%)
Mask R-CNN with RDD 2018 [18]	52.80
Faster R-CNN with Resnet-101 backbones with RDD 2020 [20]	54.26
Ensemble(YOLO-v4+Faster-RCNN) with RDD 2020 [21]	57.07
YOLOv5x with RDD 2020 [22]	57.10
YOLOv7 with RDD 2022 [19]	66.30
Our proposed modified YOLOv7 with RDD 2022	**66.87**



(a) Input Image (b) Ground Truth (c) Detection Output

Figure 1.10: Demonstration of the YOLO7 model detection results: (a) Input Image; (b) Ground Truth; (c) Detection Output.

1.5.5 Comparison of existing approaches for road damage detection and classification

Table 1.6 shows the drastic performance gap between our proposed YOLOv7 and other object detection methods on the RDD 2018, RDD 2020, and RDD 2022 datasets. Singh et al. [18] used Mask RCNN with the RDD 2018 dataset to acquire an F1 score of 52.3%. Numerous studies utilized RDD 2020, the successor to RDD 2018, and produced a higher F1 score than previous research [20, 21]. Using the RDD 2020 dataset and other object detection algorithms, the researcher discovered that the YOLO-based approach outperformed most of the other studies [22]. We outperformed all previous studies that used RDD 2018 and RDD 2020 datasets by a significant margin. The only literature we came across with RDD 2022 had an average F1 score of 66.46% [19]. We scored 0.85% higher than their average F1 score. The proposed YOLOv7 model can accurately identify road damages with satisfactory confidence.

Various object detection models were tested, and it was determined that YOLOv7 outperformed most other models with an F1 score of 68.53. Subsequently, we conducted a series of experiments utilizing various iterations of YOLOv7 and determined that the proposed YOLOv7 exhibits superior performance to both mean average precision (mAP) and F1 score with a good FPS.

1.6 CONCLUSION

Rapid and precise road damage identification can greatly benefit the road maintenance industry and contribute significantly to the economy. Applying deep learning techniques becomes vital since it drastically simplifies road inspection and provides a comprehensive view of the road's overall condition. Our proposed work uses multiple state-of-the-art object detection models to analyze the most recent benchmark road damage dataset, RDD 2022. The CBAM attention mechanism was employed in the YOLOv7 network. Simultaneously, the K-means++ algorithm is utilized to ascertain our model's most suitable anchor box. Though the mainstream approach to road damage detection and classification is object detection, doing instance segmentation is possible and better. In the future, we will develop an instance segmentation model to segment and classify front-view images of road damage. This method can significantly improve the task's accuracy and precisely determine the damage's location. Due to the unavailability of the instance segmentation-based dataset for road damage, this category still needs to be examined for better results. Creating segmentation for front-view images of these types of damage is complex and laborious.

Bibliography

[1] Torbaghan, M. E., Li, W., Metje, N., Burrow, M., Chapman, D. N., Rogers, C. D. (2020). Automated detection of cracks in roads using ground penetrating radar. Journal of Applied Geophysics, 179, 104118.

[2] Nguyen, T. S., Begot, S., Duculty, F., Avila, M. (2011, September). Free-form anisotropy: A new method for crack detection on pavement surface images. In 2011 18th IEEE International Conference on Image Processing (pp. 1069-1072). IEEE.

[3] Nguyen, H. T., L. T. Nguyen, and Denis Nikolaevich Sidorov. ”A robust approach for road pavement defects detection and classification.” Journal of Computational and Engineering Mathematics 3.3 (2016): 40-52.

[4] Wang, Y., Song, K., Liu, J., Dong, H., Yan, Y., Jiang, P. (2021). RENet: Rectangular convolution pyramid and edge enhancement network for salient object detection of pavement cracks. Measurement, 170, 108698.

[5] Naddaf-Sh, S., Naddaf-Sh, M. M., Kashani, A. R., Zargarzadeh, H. (2020, December). An efficient and scalable deep learning approach for road damage detection. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 5602-5608). IEEE

[6] Mandal, Vishal, Abdul Rashid Mussah, and Yaw Adu-Gyamfi. ”Deep learning frameworks for pavement distress classification: A comparative analysis.” 2020 IEEE International Conference on Big Data (Big Data). IEEE, 2020.

[7] Yusof, N. A. M., Ibrahim, A., Noor, M. H. M., Tahir, N. M., Yusof, N. M., Abidin, N. Z., Osman, M. K. (2019, November). Deep convolution neural network for

crack detection on asphalt pavement. In Journal of Physics: Conference Series (Vol. 1349, No. 1, p. 012020). IOP Publishing.

[8] Zhang, A., Wang, K. C., Li, B., Yang, E., Dai, X., Peng, Y., ... Chen, C. (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.

[9] Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., Omata, H. (2018). Road damage detection and classification using deep neural networks with smartphone images. Computer-Aided Civil and Infrastructure Engineering, 33(12), 1127-1141.

[10] Jeong, Dongjun. ”Road damage detection using YOLO with smartphone images.” 2020 IEEE international conference on big data (big data). IEEE, 2020.

[11] Wang, W., Wu, B., Yang, S., Wang, Z. (2018, December). Road damage detection and classification with faster R-CNN. In 2018 IEEE international conference on big data (Big data) (pp. 5220-5223). IEEE.

[12] Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Sekimoto, Y. (2022). RDD2022: A multi-national image dataset for automatic Road Damage Detection. arXiv preprint arXiv:2209.08538.

[13] Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Sekimoto, Y. (2021). RDD2020: An annotated image dataset for automatic road damage detection using deep learning. Data in brief, 36, 107133.

[14] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (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).

[15] Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. ”YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors.” arXiv preprint arXiv:2207.02696 (2022).

[16] Hao, Wang, and Song Zhili. ”Improved mosaic: algorithms for more complex images.” Journal of Physics: Conference Series. Vol. 1684. No. 1. IOP Publishing, 2020.

[17] Huang, Z., Wang, J., Fu, X., Yu, T., Guo, Y., Wang, R. (2020). DC-SPP-YOLO: Dense connection and spatial pyramid pooling-based YOLO for object detection. Information Sciences, 522, 241-258.

[18] Singh, Janpreet, and Shashank Shekhar. ”Road damage detection and classification in smartphone captured images using mask r-cnn.” arXiv preprint arXiv:1811.04535 (2018).

[19] Pham, Vung, Du Nguyen, and Christopher Donan. ”Road Damages Detection and Classification with YOLOv7.” arXiv preprint arXiv:2211.00091 (2022).

[20] Vishwakarma, Rahul, and Ravigopal Vennelakanti. ”Cnn model tuning for global road damage detection.” 2020 IEEE International Conference on Big Data (Big Data). IEEE, 2020

[21] Liu, Y., Zhang, X., Zhang, B., Chen, Z. (2020, December). Deep network for road damage detection. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 5572-5576). IEEE.

[22] Jeong, Dongjun. ”Road damage detection using YOLO with smartphone images.” 2020 IEEE international conference on big data (big data). IEEE, 2020.

[23] Woo, S., Park, J., Lee, J. Y., Kweon, I. S. (2018). Cbam: Convolutional block attention module. In Proceedings of the European Conference on computer vision (ECCV) (pp. 3-19).

[24] 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.

[25] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... Zitnick, C. L. (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 (pp. 740-755). Springer International Publishing.

[26] Arthur, D., Vassilvitskii, S. (2007, January). K-means++ the advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (pp. 1027-1035).