applied sciences

MDPI

Article

Performance Analysis of the YOLOv4 Algorithm for Pavement Damage Image Detection with Different Embedding Positions of CBAM Modules

Li Li [1,*], Baihao Fang [1] and Jie Zhu [2]

[1] School of Mechanics and Engineering Science, Shanghai University, Shanghai 200444, China
[2] CATS Testing Technology (Beijing) Co., Ltd., Beijing 100029, China
* Correspondence: lilishu@shu.edu.cn

Abstract: One of the most critical tasks for pavement maintenance and road safety is the rapid and correct identification and classification of asphalt pavement damages. Nowadays, deep learning networks have become the popular method for detecting pavement cracks, and there is always a need to further improve the accuracy and precision of pavement damage recognition. An improved YOLOv4-based pavement damage detection model was proposed in this study to address the above problems. The model improves the saliency of pavement damage by introducing the convolutional block attention module (CBAM) to suppress background noise and explores the influence of the embedding position of the CBAM module in the YOLOv4 model on the detection accuracy. The K-means++ algorithm was used to optimize the anchor box parameters to improve the target detection accuracy and form a high-performance pavement crack detection model called YOLOv4-3. The training and test sets were constructed using the same image data sources, and the results showed the mAP (mean average precision) of the improved YOLOv4-3 network was 2.96% higher than that before the improvement. The experiments indicate that embedding CBAM into the Neck module and the Head module can effectively improve the detection accuracy of the YOLOv4 model.

Keywords: pavement maintenance; YOLOv4; crack identification; CBAM

check for updates

Citation: Li, L.; Fang, B.; Zhu, J. Performance Analysis of the YOLOv4 Algorithm for Pavement Damage Image Detection with Different Embedding Positions of CBAM Modules. *Appl. Sci.* **2022**, *12*, 10180. https://doi.org/10.3390/app121910180

Academic Editor: Luís Picado Santos

Received: 7 September 2022
Accepted: 7 October 2022
Published: 10 October 2022

1. Introduction

Cracking is a common type of pavement damage. The appearance of cracks accelerates the deterioration of road performance, and serious cracks may endanger traffic safety [1]. A study by Lee et al., found that 16% of traffic accidents were related to road environmental factors (mainly caused by poor pavement conditions) [2]. The deterioration of pavement surface conditions increases the probability of multi-vehicle collisions and aggravates the severity of accidents [3], so timely repair of pavement cracks is imperative. In addition, early repair of cracks before they develop also helps to reduce road maintenance costs and extend pavement life [4]. Yu et al., found that the continued use of preventive maintenance on asphalt pavements resulted in a 27% reduction in maintenance costs compared to continuous corrective maintenance [5]. Therefore, timely repair of pavement cracks is necessary, and the prerequisite for timely repair is the early detection and proper documentation of cracks.

The initial pavement crack detection was conducted by manual visual inspection, but it had the disadvantages of high subjectivity, low detection efficiency, and high personnel safety risk [6,7]. With the rapid development of information technology over the last three decades, the application of computer vision techniques for pavement crack detection has gained a large amount of attention. Many studies have used image processing techniques for pavement crack detection. Among these techniques, the traditional image processing methods mainly include edge detection [8], threshold segmentation [9], and region growing [10] methods. The edge detection algorithm identifies the edges of pavement cracks by using edge detection operators such as the Sobel operator [11], the Prewitt operator [12], and the Canny operator [13,14]. The threshold segmentation method separates the target cracks from the background by setting a suitable pixel threshold [15,16]. The region growing method is a method to separate cracks by merging neighboring pixels with similar properties (e.g., grayscale, texture color) into one region [17,18]. All of these traditional image processing methods have promoted the development of automatic pavement damage detection and identification technology to different degrees [19,20], but there are also quite a few limitations in practice. Both the edge detection algorithm and the threshold segmentation method are sensitive to image noise [21,22], so it is difficult to separate the crack morphology completely and accurately. The region growing method, on the other hand, has difficulties in the reasonable selection of initial pixel points. In summary, these types of processing techniques based on single image pixel feature analysis have poor generalization ability and low robustness, and thus are difficult to be applied on a large scale in a real sense.

The emergence of deep learning algorithms has given rise to new ideas for pavement image processing in this context. Deep-learning-based target detection techniques can learn the deep features of existing image data and use the similarity between data to predict unknown images. Compared with the traditional image processing methods, this method has higher generalization ability and more accurate detection accuracy. Many studies have been conducted to explore this.

Girshick et al., proposed region convolutional neural networks (R-CNN) in 2014 [23], a deep learning-based target detection and recognition algorithm that is able to detect the desired target on the basis of the extracted features. Then, in 2015, Girshick et al., improved on the R-CNN by proposing the fast R-CNN algorithm [24]. This algorithm enables the classification and localization tasks to not only share convolutional features but also boost each other through region of interest (ROI) pooling and joint multi-task training, and thus the detection accuracy is greatly improved. In the same period, Ren et al., also proposed the faster R-CNN algorithm framework [25]. This algorithm replaced the selective search algorithm with the region proposal network (RPN), which improves the detection speed by sharing convolutional layer features to achieve candidate region extraction and category prediction.

These studies have improved the accuracy and speed of target detection to varying degrees but still cannot meet the requirements of real-time image detection. Therefore, regression-based detection algorithms have been introduced, mainly including the You Only Look Once (YOLO) [26] and Single Shot MultiBox Detector (SSD) [27] algorithms. Redmon et al. proposed the YOLO algorithm in 2015 on the basis of the regression concept. This algorithm first divides the image into a number of equally sized grids and subsequently selects the position among them with the highest probability of target detection. It helps to improve the recognition accuracy in general, but the detection capability for fine targets is still weak and the generalization capability is not satisfactory. To address the above issues and to further improve the recognition accuracy and speed, Redmon proposed YOLOv2 [28] in 2017, which introduced the Anchor box mechanism and used the K-means clustering algorithm to obtain a more suitable Anchor box for the model. Shortly thereafter, Redmon proposed YOLOv3 [29] in 2018, using the darknet-53 feature extraction network, while using multi-category cross-entropy as well as binary cross-entropy to allow the model's classification power to be further improved. In the same period, the SSD algorithm proposed by Liu in 2016 makes full use of the information of the feature maps in each layer of the feature pyramid to improve the detection accuracy of small target objects to some extent. However, compared with the YOLO series algorithm, the generalization ability and detection accuracy of the SSD algorithm still needs to be improved [30–32]. YOLOv4 [33] and YOLOv5 [34] are the most recent versions of the YOLO family of target detection algorithms, both of which have significantly improved accuracy and speed compared to previous versions [35]. YOLOv4 and YOLOv5 have the ability to recognize abstract features of an image by extracting deeper target features through convolutional neural networks, which are able to fuse features at different scales. Thus, both outperform traditional algorithms in terms of speed and accuracy. Although the two are generally similar in terms of performance [36], YOLOv4 is generally considered to be more robust [37,38]. In addition, each component of the Yolov4 model is more independent and thus more suitable for observing the impact of embedding CBAM modules at different locations on YOLO network performance. Therefore, the YOLOv4 algorithm was used in this study to detect the pavement cracking images.

Meanwhile, to improve the detection accuracy, this paper improved the K-means [39] clustering algorithm for obtaining the Anchor box in YOLOv4 and embedded the convolutional block attention module (CBAM) [40].

The CBAM combines the channel attention mechanism and spatial attention mechanism [34], which can enhance crack features in the channel and space of the feature map while suppressing unimportant features such as noise [35]. In addition, the CBAM occupies only a small number of parameters in the network and does not affect the iteration speed of the network model, thus ultimately achieving the effect of improving crack detection accuracy without reducing the computational speed. Similar attempts have been made in some studies. For example, Zhang [41] added a CBAM to the YOLOv3 model to improve the detection accuracy of bridge crack images, but the study did not describe the embedding location of the CBAM. Yang [42] tried to embed the CBAM in the Neck region when using YOLOv4 to detect the number of wheat spikes but did not explain the reason for choosing this location. Therefore, in order to better understand the relationship between the embedding position of the CBAM module and the network performance, this paper tried to embed the CBAM module in the Backbone, Neck, and Head regions of YOLOv4, respectively, and tested the corresponding network performance changes.

In data mining, clustering is one of the most commonly used methods. The K-means algorithm has been widely used due to its inherent advantages of good clustering, simplicity of thought, and fast clustering. However, the shortcoming of the K-means algorithm is also obvious, which is that it is not stable enough in the selection of initial points [43]. The K-means++ algorithm was able to optimize the size and position of the Anchor box, resulting in a YOLOv4 network that is more suitable for pavement crack detection [37]. Therefore, the K-means clustering algorithm in the original model was replaced by the K-means++ clustering algorithm in order to make the model more adaptable to the detection of pavement cracks.

2. Data Preparation

The image data used in this study were collected by CiCS, a rapid road condition detection system widely used in China. The original image format is "jpg" with a resolution of 96 dpi (3024 × 1887 pixels). A total of 1923 pavement cracking images were used, of which 1538 were used as the training set and 385 were used as the test set. Both the training and test sets were randomly selected.

On the basis of the damage composition of the original pavement image, the cracking images can be classified into four categories: transverse cracks, longitudinal cracks, alligator cracks, and crack sealing. Therefore, in this paper, the open-source deep learning annotation tool LabelImg was used to manually annotate the cracking images as follows (Figure 1), and the labeled region was called the Bounding box.

- Transverse crack: Tcrack;
- Longitudinal cracks: Lcrack;
- Alligator crack: Acrack;
- Cracks sealing: Repair.

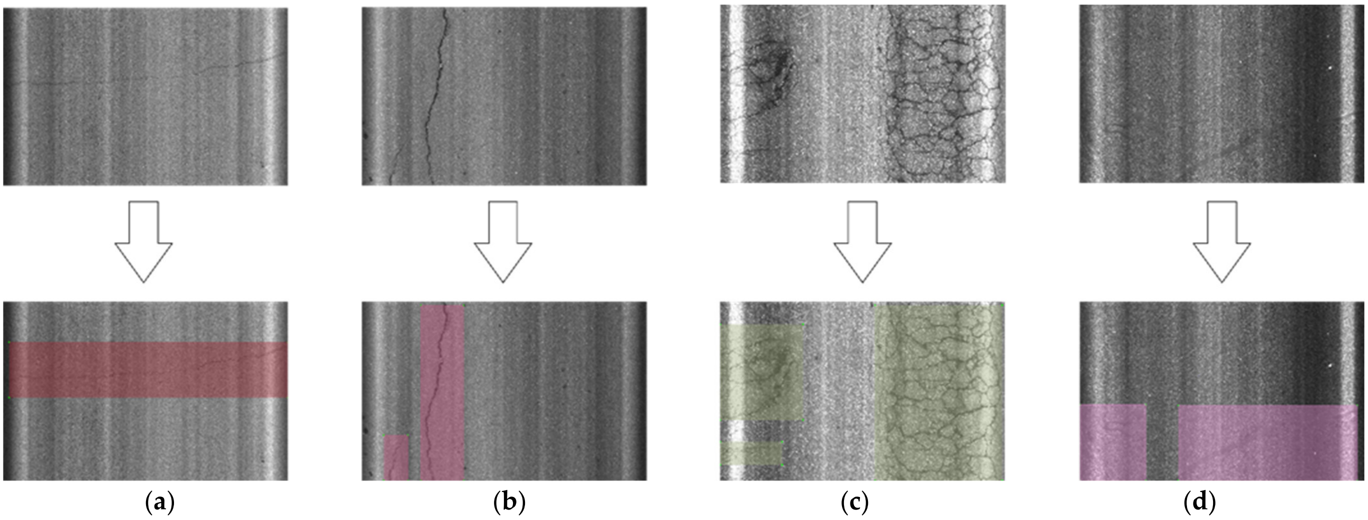


Figure 1. Image annotation visualization. (**a**) Transverse cracks; (**b**) longitudinal cracks; (**c**) alligator crack; (**d**) crack sealing.

3. Image Processing Algorithms

3.1. YOLOv4 Algorithms

YOLOv4 is an end-to-end algorithm for target detection. It was developed by Bochkovskiy on the basis of YOLO, YOLOv2, and YOLOV3 by improving and optimizing the data processing, backbone network, activation function, and loss function.

The network structure of YOLOv4 mainly consists of four parts: Input, Backbone, Neck, and Head, where the Backbone module is the initial feature extraction of the input image, the Neck module is to enhance the extracted image features, and the Head module uses the extracted features for detection, as shown in Figure 2. The Input module compresses random input images to the size of 416 × 416 and delivers them to the Backbone module. The CSPDarknet53 of the Backbone module is composed of the DarknetConv2D_Batch normalization_Mish (CBM) module and the CSP (cross-stage partial network) module; the Backbone module outputs three initial feature layers with feature map height, width, and the number of channels of 52 × 52 × 256, 26 × 26 × 512, and 13 × 13 × 1024, respectively. The Backbone module and the Head module are inserted between the depth feature extraction module, which is called Neck. The Neck module includes the spatial pyramid pooling (SPP) [44] and the path aggregation network (PANet) [45]. Among them, the SPP can fuse features of multiple dimensions together, and its structure uses pooling kernels at different scales of 1 × 1, 5 × 5, 9 × 9, and 13 × 13 to maximize the pooling of the feature layer result output from the backbone convolution, which can effectively increase the acceptance domain range of the backbone features and separate the most significant upper and lower background features. The PANet structure proposed in 2018 is an innovation in the field of the International Conference on Computer Vision and Pattern Recognition (CVPR) image segmentation, having the advantage of accurately preserving spatial information and helping to correctly locate pixel points, and its main role is to perform feature fusion on the three initial effective feature layers so that better and more effective feature layers can be extracted to improve the detection accuracy of the detection module Head. In the Head module, the decoding operation is performed on the obtained feature maps, and YOLO Head presets three anchor boxes to predict the target boxes. The input images are scored and boxed on each of these three scales during detection, and the target to be detected is finally selected.

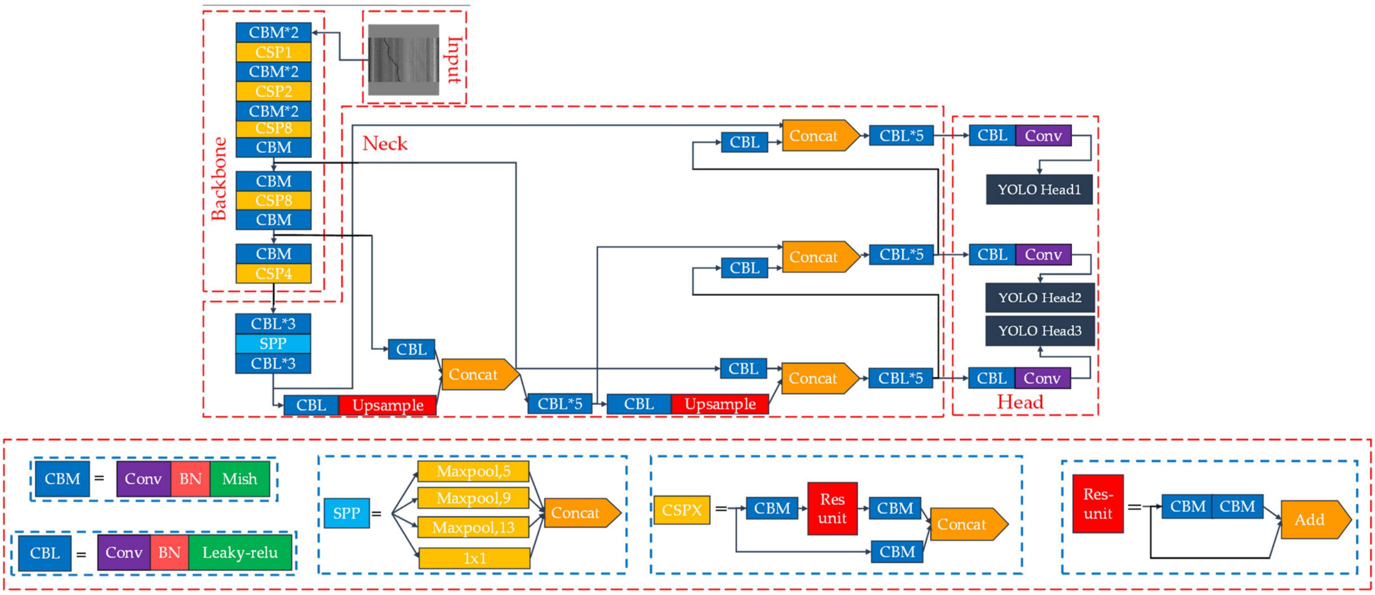


Figure 2. YOLOv4′s structure chart.

3.2. CBAM Module

Embedding CBAM modules in the network of YOLOv4 can effectively improve the network performance by increasing the crack information weight [42] while suppressing other useless information [41]. The CBAM modules can be divided into two categories: channel attention modules and spatial attention modules, which are implemented in different ways.

- Channel attention module

The process of feature map enhanced by attention channels can be displayed in Figure 3. The input feature map F is assumed to be H × W × C, where H, W, and C are the length, width, and number of channels of the feature, respectively. The input feature map is firstly subjected to a pooling layer of 1 × 1 × c (c is the number of channels) and a global average pooling layer of 1 × 1 × c, and then performs the shared MLP (muti-layer perception) operation. Subsequently, feature extraction is performed once again to obtain the maximum pooling channel and the global average pooling channel. The two are superimposed in a one-dimensional vector summation to obtain the attention channel. Finally, the attention channel is multiplied with the initial input feature map to obtain the feature map FM′ enhanced by the attention channel.

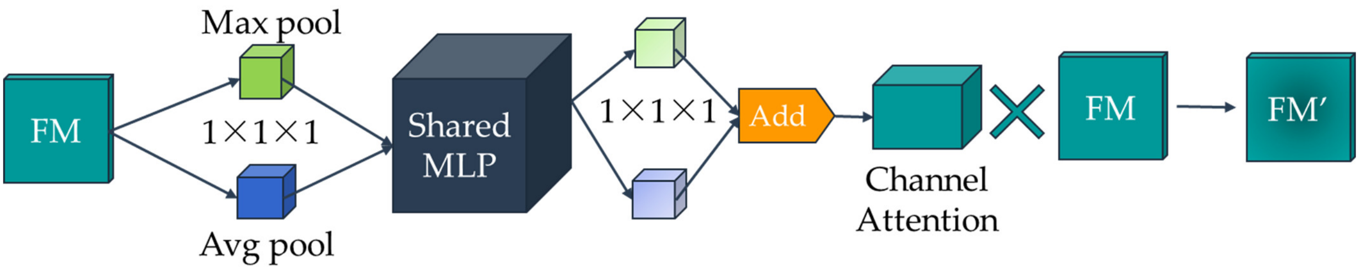


Figure 3. Flow of the channel attention module (when a feature map is used, the value of H × W × C is shown as 1 × 1 × 1).

- Spatial attention module

The process of a feature map enhanced by spatial channels is shown in Figure 4. The input layer of the spatial attention module (FM′) is exactly the output layer of the channel attention module. The input layer is merged over the channels to obtain the

H × W × 2 features after global average pooling and maximum pooling. After that, the spatial attention kernel of H × W × 1 is obtained after 7 × 7 convolution and sigmoid function activation. Finally, by multiplying the spatial attention kernel with the input FM′ through the broadcast mechanism, the feature map FM″ enhanced by the CBAM module is obtained.

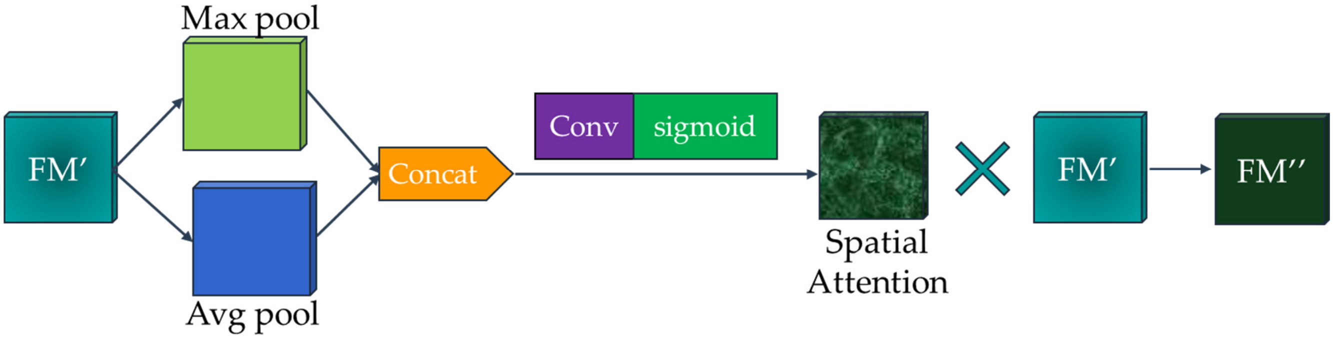


Figure 4. Spatial attention module process.

3.3. K-Means++ Clustering Algorithm

In this study, the K-means++ clustering algorithm was used to cluster and analyze the Anchor box of targets in the pavement crack dataset. Compared with the K-means clustering algorithm used in the original YOLOv4 network, K-means++ optimizes the selection of initial points and can obtain the size of the Anchor box that is more suitable for the target dataset, thus improving the accuracy of target detection [46].

The K-means algorithm is an unsupervised learning and partitioning-based clustering algorithm [36]. The Euclidean distance is commonly used as a measure of data object similarity. The distance between data objects is inversely proportional to similarity. The algorithm requires specifying the initial number of clusters in advance, i.e., specify k initial clustering centers. Then, the positions of clustering centers are continuously updated on the basis of the similarity between data objects and the clustering centers, and the sum of squared clustering errors are continuously reduced. The clustering process ends when the squeezing channel-wise and exciting spatially (SSE) index stops changing or the objective function converges, and the final result is obtained. However, the choice of k initial center locations has a significant impact on the final clustering results and running time. If the selection is completely random, it may lead to slow convergence of the algorithm.

The K-means ++ algorithm is optimized for this problem [43]. The K-means++ algorithm, after selecting the initial clustering center, subsequently prioritizes the points that are farther from the initial point to avoid the situation where two points overlap.

To further analyze and compare the clustering effects of these two algorithms, a test dataset including 4000 samples was created in this paper. Four sample centroids were set in the dataset with coordinates of [−2, 0], [0, 2], [2, 0], [0, −2], and the variances of each centroid were [0.1, 0.5, 0.2, 0.2], respectively. Then the dataset was clustered by K-means and K-means++ algorithms. The results of both are shown in Figure 5 when the numbers of cluster centers were 1, 4, and 7. It can be found that when the number of cluster centers was 7, there was overlap in the cluster centers obtained by the K-means algorithm, while the cluster centers of the K-means++ algorithm were more uniformly distributed.

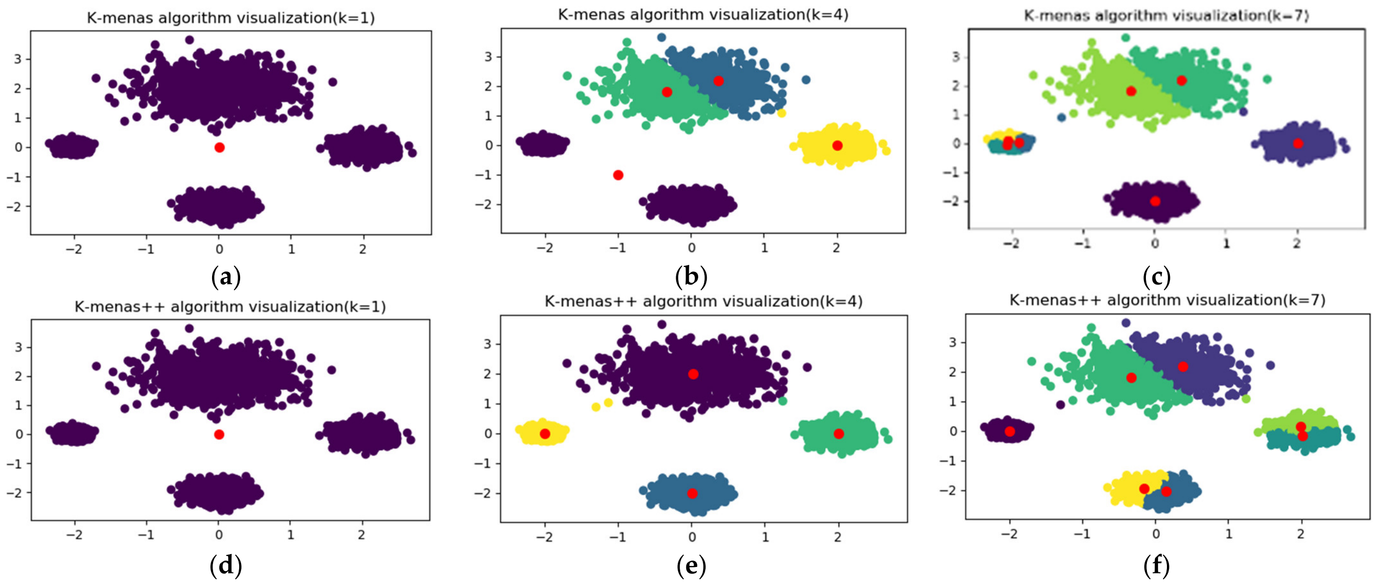


Figure 5. Comparison between K-means and K-means++ algorithms. (**a**) One cluster centers (K-means). (**b**) Four cluster centers (K-means). (**c**) Seven cluster centers (K-means). (**d**) One cluster centers (K-means++). (**e**) Four cluster centers (K-means++). (**f**) Seven cluster centers (K-means++).

4. Improved YOLOv4 Algorithm

Theoretically, a CBAM module can be embedded as a universal module at any location in the network structure [46], and embedding a CBAM module in a deep learning network can help to improve the performance of the network in general [41,42]. However, the degree and mechanism of the influence of embedding location on the performance enhancement effect of CBAM modules are still not fully understood, so the performance of CBAM modules embedded into different locations in the network model will be investigated in this paper. The K-means++ clustering algorithm will also be used to cluster and optimize the Anchor box.

4.1. Adding Attention Modules

In this study, six experimental models were obtained on the basis of the summary of existing studies [41,42,47–51], as shown in Figure 6. Among them, Figure 6a–d represents the modules of the original YOLOv4; Figure 6e,f shows the CBAM attention modules embedded after the CBM and CBL modules in Figure 6a,b, respectively; and Figure 6g,h shows the replacement of the CBM modules in Figure 6c,d with the CBMC modules.

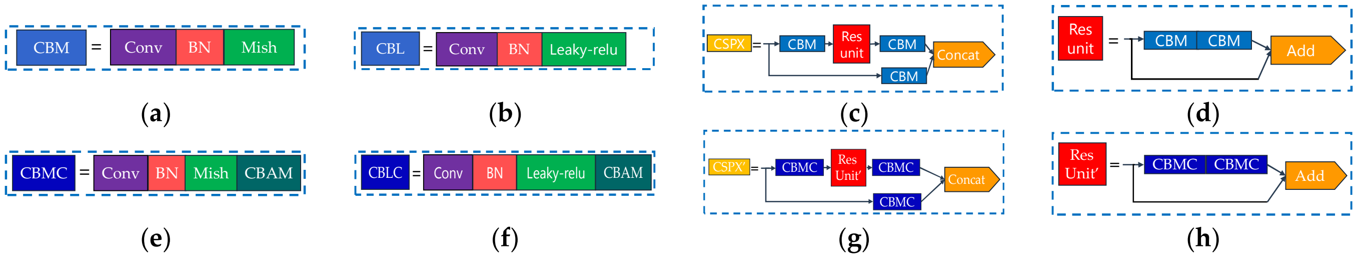
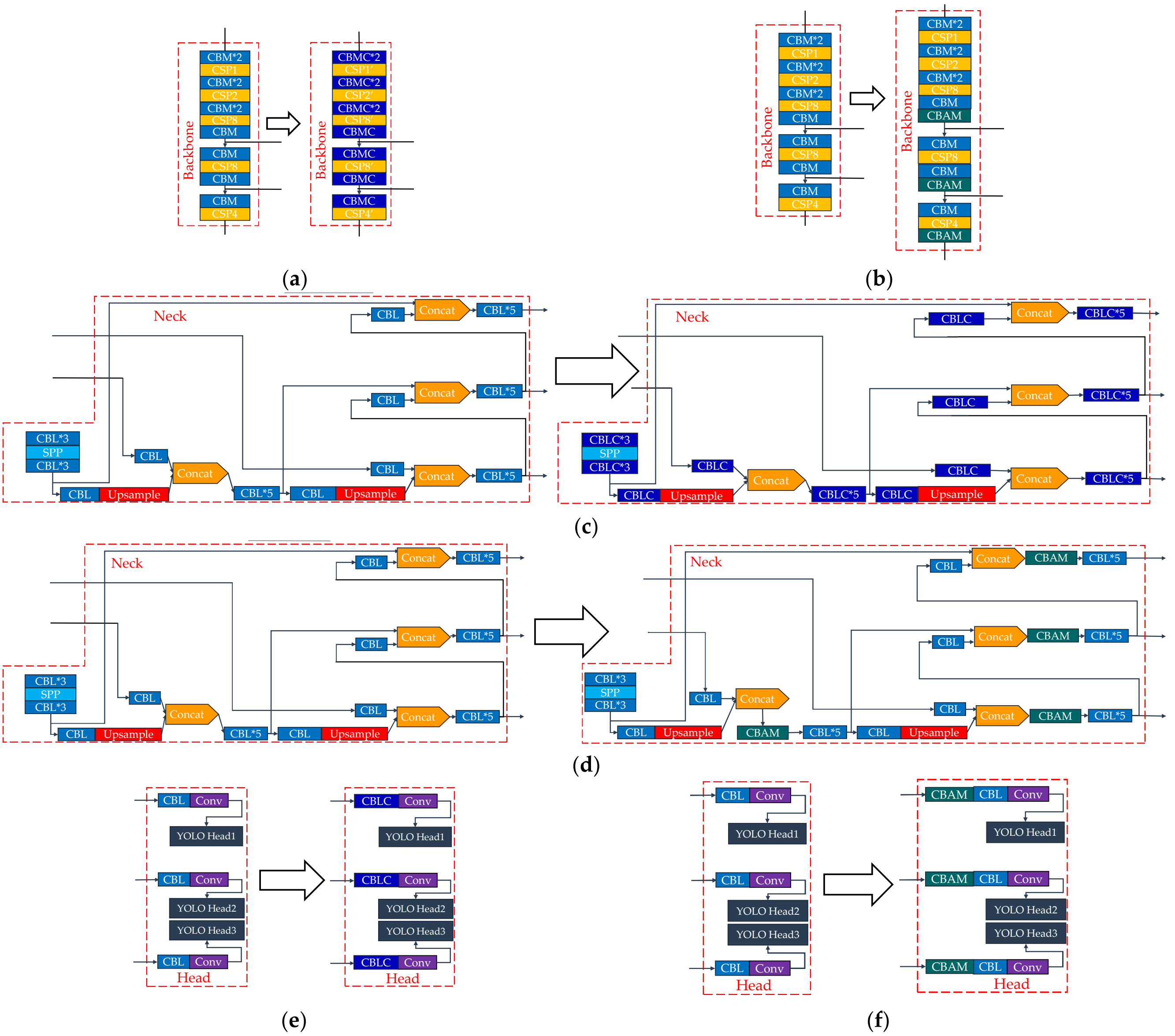


Figure 6. The original module and the improved module. (**a**) CBM module. (**b**) CBL module. (**c**) CSP module. (**d**) Resunit module. (**e**) CBMC module. (**f**) CBML module. (**g**) CSPX′ module. (**h**) Resunit′ module.

Subsequently, the modules were replaced separately for each region of the new module, and a total of three new models were obtained, as shown in Figure 7, namely/YOLOv4B1 (for the Backbone, Figure 7a), YOLOv4N1 (for the Neck, Figure 7c), and YOLOv4H1 (for the Head, Figure 7e). After that, YOLOv4B2 was obtained by adding CBAM modules to the three feature layers in the Backbone region that were to be input to the Neck region (Figure 7b). YOLOv4N2 was obtained by placing the CBAM module after each of the Concat layers in the Neck region (Figure 7d). Moreover, YOLOv4H2 was obtained by placing the CBAM module after the three feature layers in the Head region that would be accepted from the Neck region (Figure 7f).



(a) (b)

(c)

(d)

(e) (f)

Figure 7. Adding CBAM modules in the Backbone, Neck, and Head regions. (**a**) YOLOv4B1. (**b**) YOLOv4B2. (**c**) YOLOv4N1. (**d**) YOLOv4N2. (**e**) YOLOv4H1. (**f**) YOLOv4H2.

4.2. Anchor Box Optimization

When compared to the traditional K-means clustering algorithm, the K-means++ algorithm optimizes the initial Anchor box selection, which can significantly improve the error of classification results in order to obtain the size of Anchor box that is most appropriate for this crack dataset and improve the accuracy of small target detection.

The original K-means++ clustering algorithm of the YOLOv4 network was first used to obtain nine pre-selected boxes (Anchor box = 48,181; 53,401; 85,398; 124,156; 12,861; 169,402; 23,195; 388,166; 39,499) for the 52 × 52, 26 × 26, 13 × 13, and 13 × 12 scales. These

Anchor boxes are obtained by clustering the Bounding box annotated with LabelImg on the dataset.

The following are the steps for obtaining the Anchor box using the K-means ++algorithm:

1. Extract the width and height of the rectangular boxes of all Bounding boxes;
2. Selected an Anchor box as the initial clustering center at random from all Bounding boxes;
3. It calculates the distance $\mathrm{D}(x_i)$ between the centroids of all Bounding boxes and the centroids of existing Anchor boxes, and thus calculates the probability $\mathrm{P}(x_i)$ of each Bounding box being selected as the next clustering center; the further the bounding box was from the initial clustering center, the more likely it was to be selected. $\mathrm{P}(x_i)$ is calculated as shown in Equation (1):

$$\mathrm{P}(x_i) = \frac{\mathrm{D}(x_i)^2}{\sum_{i=1}^{n} \mathrm{D}(x_i)^2} \tag{1}$$

4. After that, the IOU value of each bounding box and each anchor box is calculated as shown in Figure 8, and the Anchor box with the largest IOU value is selected in each Bounding box and attributed to that Anchor box;

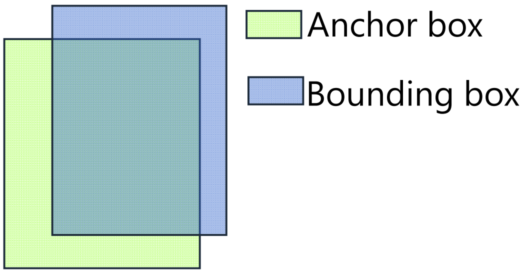


Figure 8. Calculation of the IOU.

5. Repeat the four step until the classification of the Bounding box no longer changes, and obtain the final Anchor box.

The IOU is obtained from the intersection of the Anchor box and the Bounding box by combining them, as shown in Equation (2):

$$\mathrm{IOU} = \frac{A \cap B}{A \cup B} \tag{2}$$

A larger IOU value indicates that the Anchor box obtained by the K-means algorithm is closer to the value of the various types of labeled Bounding boxes.

5. Results and Analysis

Intel (R) Core (TM) i5-9300H.CPU@2.40 GHz.gtx1650, Windows 10, 64-bit OS was the experimental environment in this paper. The validation set to training set ratio was 1:9, the learning rate was 0.001, the batchsize was 4, and the training Epochs were 50.

5.1. Evaluation Criteria

The most commonly used evaluation criteria for the target detection task are Precision (P), Recall (R), F1-Score, mean average precision (mAP), and Detection Rate (fps, number of frames recognized in one second) [50]. When performing transverse crack detection, transverse cracks are defined as positive samples, while non-transverse cracks are defined as false samples (class of disturbances). The samples can be classified into four types on the basis of the combination of true and predicted values [50]:

- True positives (TP): the positive sample is correctly identified as a positive sample (i.e., the transverse crack image is correctly identified);

- True negatives (TN): negative samples are correctly identified as negative samples (i.e., the non-transverse crack images are correctly identified as non-transverse cracks);
- False positives (FP): negative samples are incorrectly identified as positive samples (i.e., the non-transverse crack images are incorrectly identified as the transverse cracks);
- False negatives (FN): positive samples are incorrectly identified as negative samples (i.e., the transverse crack images are incorrectly identified as the non-transverse cracks).

Therefore, the above evaluation indicators can be described as follows.

Precision: it is used to measure the accuracy of the model for positive samples and indicates the ability of the classifier to discriminate between positive and negative samples, which can be defined as

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3}$$

Recall: it is used to measure the completeness of a classifier for positive samples and indicates the sensitivity or coverage of the classifier for positive samples, which can be defined as

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4}$$

F1-Score: the summed average of the Precision and Recall metrics, defined as

$$\mathrm{F1} = \frac{2 \times \mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{5}$$

AP: the performance of a model is measured by examining both Precision and Recall metrics, representing the performance of the model in identifying a particular category, with the following expression:

$$\mathrm{AP} = \int_0^1 \mathrm{P}(\mathrm{R})\mathrm{dR} \tag{6}$$

mAP: the mean value of AP values for each category, reflecting the average classification performance of the model for all categories, defined as

$$\mathrm{mAP} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{AP_i}}{\mathrm{n}} \tag{7}$$

5.2. Test Results

5.2.1. Comparison of the Effect between K-Means++ and K-Means Algorithms

To examine the effect of the K-means++ clustering algorithm on the model alone, the attention module was not embedded in this experiment.

The change in the target detection performance of the YOLOv4 algorithm for pavement cracks after optimizing the Anchor box using the K-means++ clustering algorithm was first verified using the test dataset established above. The test results are shown in Table 1. From Equations (3) and (4), it can be seen that accuracy and recall are mutually constrained; the higher the accuracy, the lower the recall will be in relative terms. Therefore, the recall obtained by using the K-means++ algorithm is substantially increased, and instead the accuracy will be slightly decreased. The F1-Score as a comprehensive index is to balance the influence of accuracy and recall; the higher the F1-Score, the higher the accuracy of the model. A total of 11.5% improvement in F1-Score indicated that the model was more comprehensive after optimizing the Anchor box. It was also because the K-means++ algorithm optimized the parameters of the Anchor box that the recall rate of the model was improved significantly. Moreover, the mAP was improved by 0.93%, which also indicates that the improved network model also had good accuracy improvement.

Table 1. Comparison of K-means++ and K-means algorithms.

Types Methods	P (%)	R (%)	F1-Score (%)	mAP (%)	FPS (f/s)
YOLOv4 (K-means)	**96.47**	37.92	54	79.99	**15.43**
YOLOv4 (K-means++)	95.81	**50.48**	**65.5**	**80.92**	15.38

Four types of crack images were randomly selected, and their comparison results are shown in Figure 9. YOLOv4 (K-means) and YOLOv4 (K-means++) had similar APs for transverse cracks and alligator cracks; however, the latter had 25 percentage points higher AP for longitudinal cracks and 23 percentage points higher AP for right-hand crack sealing and detects crack sealing that are not detected by the former. It indicates that the Anchor box calculated by K-means++ has more stability of detection than K-means and is more suitable for detecting pavement damage. It can be seen that F1-Score enhancement allowed the model to detect longitudinal cracks and crack sealing with better accuracy.

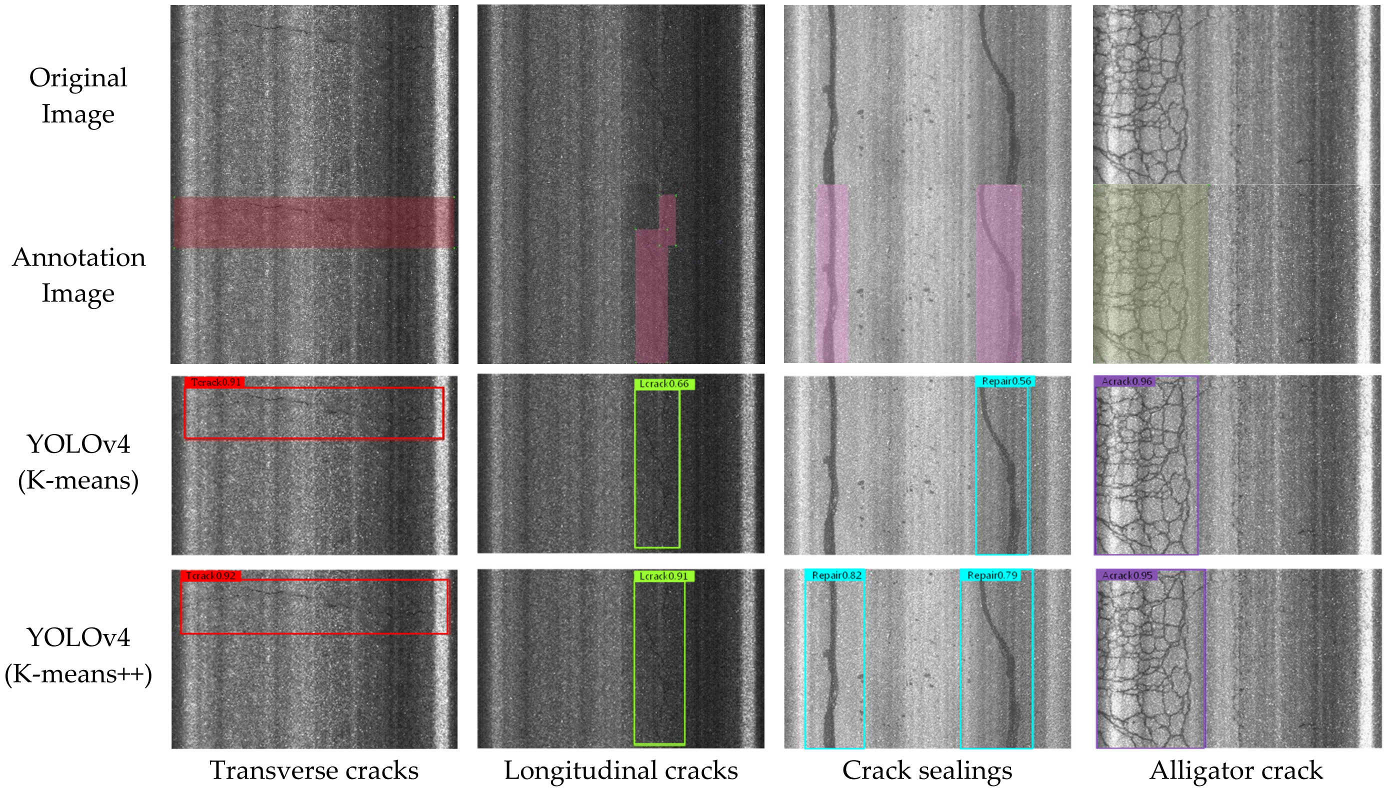


Figure 9. Detection effect of YOLOv4 (K-means) and YOLOv4 (K-means++).

5.2.2. Comparison of the Effect on Adding Different Attention Modules

The six models listed in Figure 7 were evaluated using the test dataset in order to investigate the performance differences resulting from the embedding of the CBAM modules into different locations of the YOLOv4 network. All these models used the K-means++ clustering algorithm in obtaining the Anchor box. The results are shown in Figure 10. As shown in Figure 10, YOLOv4-B1, YOLOv4-B2, and YOLOv4-N1 expressed worse results compared to YOLOv4 (K-means++); the YOLOv4-B1 model almost failed; and for the YOLOv4-B2 model, the detection accuracy of transverse cracks decreased. In the YOLOv4-N1 model, there were transverse cracks, longitudinal cracks, crack sealing, and Alligator cracks, and YOLOv4-N2, YOLOv4-H1, and YOLOv4-H2 were otherwise better than YOLOv4 (K-means++), but the detection of crack sealing showed a weakness after adding the CBAM module.

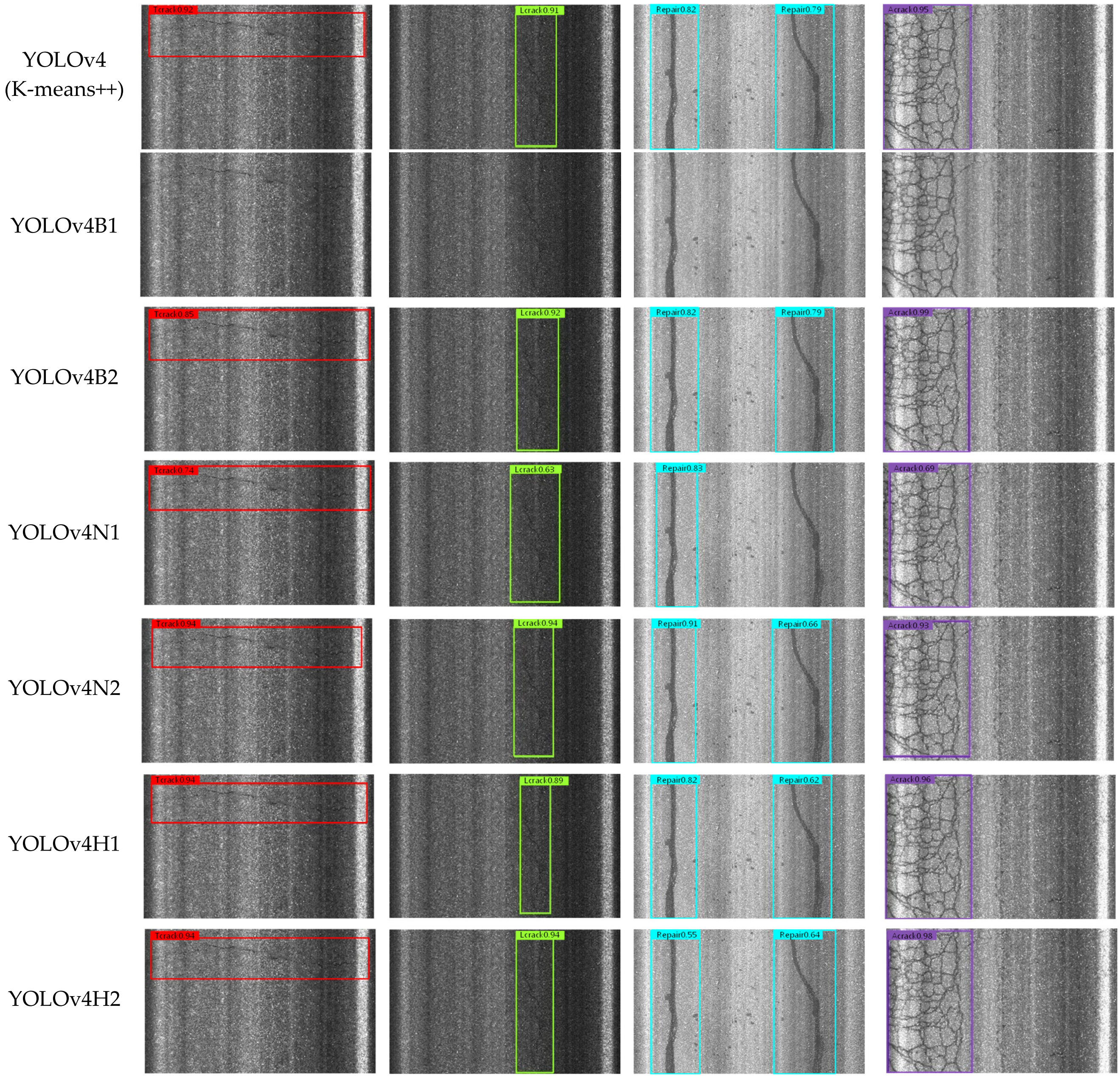


Figure 10. The detection effect of six models with different attention modules.

As shown in Table 2, it can be found that among the six models, YOLOv4B1 was the least effective and directly led to network failure. The reason may be that the weights of useless features were enhanced before the feature data deepened the features through the residual network, which led to the decrease in target weights and poor detection effect. In addition, the accuracy of YOLOv4B2 and YOLOv4N1 also showed different degrees of degradation compared with the original network without the attention module added (YOLOv4 (K-means++)). This phenomenon may have been caused by the presence of too much irrelevant data prior to concatenating the feature data, and the attention mechanism amplified the weight of these irrelevant data.

Table 2. Comparison of six models with different attention modules.

Types Methods	P (%)	R (%)	F1-Score (%)	mAP (%)	FPS (f/s)
YOLOv4 (K-means++)	95.81	50.48	65.5	80.92	**15.38**
YOLOv4B1	4.17	0.09	0	3.18	7.16
YOLOv4B2	95.05	50.38	64.75	75.84	6.44
YOLOv4N1	94.72	36.23	48.5	74.77	5.08
YOLOv4N2	95.84	53.23	68	**82.51**	6.52
YOLOv4H1	95.00	54.39	64.5	81.16	9.52
YOLOv4H2	**96.07**	**57.30**	**70.25**	82.45	14.85

In addition, in terms of mAP, YOLOv4N2 improved by 1.59%, YOLOv4H1 by 0.24%, and YOLOv4H2 by 1.53% compared to YOLOv4 (K-means++). All three models added the attention module after the Concat feature data layer, indicating that the attention module can indeed improve the accuracy of the YOLOv4 algorithm, but the embedding position should be after the Concat layer.

5.2.3. Comparison Experiments of the Four Improved Models

The above analysis shows that YOLOv4N2, YOLOv4H1, and YOLOv4H2 outperformed the original network in all aspects, so these three networks can be combined in different forms to obtain the following four new models and trained iteratively again.

- YOLOv4-1 = YOLOv4N2 + YOLOv4H1;
- YOLOv4-2 = YOLOv4N2 + YOLOv4H2;
- YOLOv4-3 = YOLOv4H1 + YOLOv4H2;
- YOLOv4-4 = YOLOv4N2 + YOLOv4H1 + YOLOv4H2.

The performance of these four networks was compared, and the results are shown in Table 3. Compared with the six models described in the previous section, the performance of YOLOv4-1, YOLOv4-2, and YOLOv4-4 showed little improvement or even a slight decrease, but the detection accuracy of YOLOv4-3 improved significantly and the mAP improved by 2.03% compared to the original YOLOv4 (K-means++) model. This indicates that the attention module works relatively best when embedded in the Head region.

Table 3. Average P, R, F1-Score, and mAP of each model.

Types Methods	P (%)	R (%)	F1-Score (%)	mAP (%)	FPS (f/s)
YOLOv4(K-means++)	95.81	50.48	65.5	80.92	15.38
YOLOv4-1	95.19	54.95	69	81.00	13.81
YOLOv4-2	94.89	52.46	67	81.34	**14.06**
YOLOv4-3	**96.23**	**56.22**	**70.25**	**82.95**	13.90
YOLOv4-4	95.60	56.01	69.75	81.50	13.93

6. Discussion

According to the experimental results, CBAM modules embedded in different positions of YOLOv4 have different effects. Yolov4-3 improves accuracy, robustness, and generalization ability compared to YOLOv4 (K-means ++). Through two experiments, the effect of CBAM on the model performance is discussed. It can be found that by increasing the weight of ROI and adding the CBAM attention mechanism, the performance of the model was effectively optimized, and the detection accuracy of the model was improved. By comparing the results of YOLOV4-3, YOLOV4-H1, and YOLOV4-H2, it can be found that the model proposed by the improved embedding method had a slight improvement in AP and mAP. The effective feature information extraction and multi-scale feature fusion enabled the model to detect pavement damage better. In general, the embedded position of the CBAM module in the YOLO-3 algorithm proposed in this paper effectively improved the detection accuracy, met the actual detection requirements, and verified the feasibility of the model.

To better explain the results of the above experiments, the Grad CAM method [42] was used in this study to visualize and analyze the feature maps of different models. This is a recently proposed visualization method that uses the gradient weights obtained from backpropagation calculations to compute the importance of spatial locations in the convolutional layers. Firstly, the Grad-CAM algorithm was used to calculate the back-propagation gradient weights of the three output feature layers in YOLOv4 to obtain the visualized feature maps with three different weights. Secondly, considering that the three output feature layers were equally important in the target detection stage, the corresponding three visualization feature maps were superimposed with 1:1:1 weight (as shown in Figure 11) to more clearly demonstrate the differences in the visualization feature maps of different models.

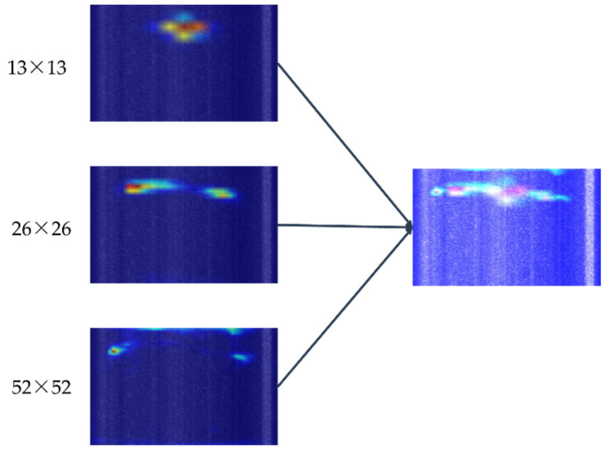


Figure 11. The three output feature layers (**left**); visualized feature maps after superimposition (**right**).

As can be seen in Figure 11, the positions of interest of the feature layers were different at different scales. Therefore, the visual feature maps can be superimposed on all six models built in Section 5.2.3 of this paper to visually compare the effect of embedding the CBAM module in different locations of the network, and the results are shown in Figure 12.

As can be seen from Figure 12, for longitudinal cracks, both YOLOv4 and the improved models can pinpoint the target region, but the YOLOv4 model focused on the target region less than the other four improved models, while the YOLOv4-3 model focused on the target region more. The other YOLOv4-3 model can focus on the target region in the image more accurately and with less background interference. This shows that the improved mechanism in the YOLOv4-3 model can effectively suppress the background noise and enhance the target features, which further proves that the method proposed in this study has a strong attention learning capability and improves the detection performance of the model for cracks.

As for the reasons why the CBAM attention module appears to have very different effects when embedded into different locations of the network, this study speculated that there were the following reasons:

First, in the Backbone network, the semantic features of the feature map were mainly in the extraction stage with a large amount of information to be extracted, while there was more interference information in the pavement image, and various complex texture contour information existed. Therefore, in the use of CBAM modules in the backbone network, non-primary information will increase, causing bad results. Second, long cracks (spanning the whole image area) were often found in pavement damage images. When embedding the CBAM module in the Backbone network, the location information of small targets was more easily noticed, while in the Head regions, the perceptual field was larger and more favorable for detecting large targets such as pavement cracks. Therefore, embedding the

CBAM module in Head regions can better strengthen the spatial features and channel features of the feature maps, thus enhancing the robustness of the network.

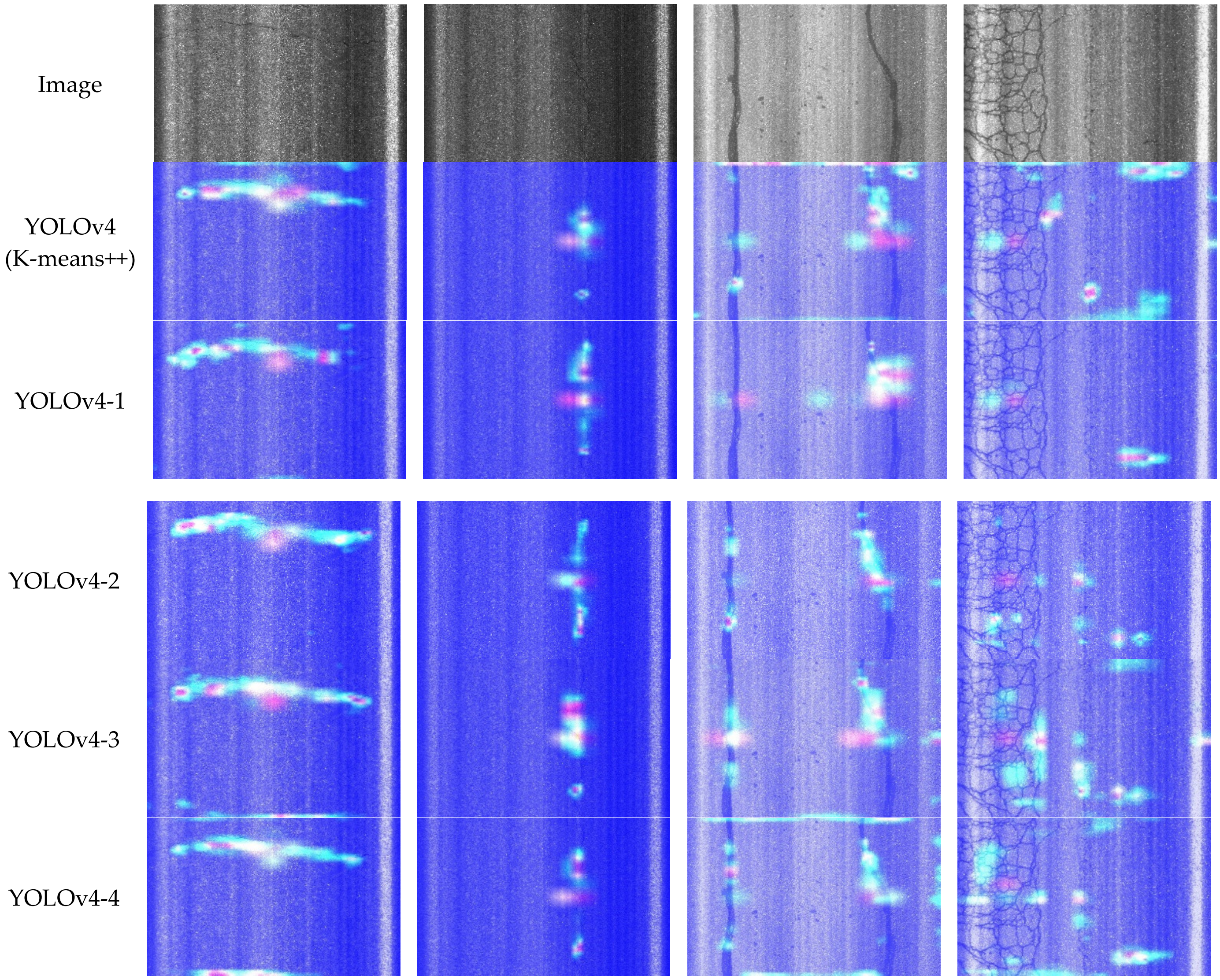


Figure 12. Visualization of the three output feature layers after superimposition.

7. Conclusions

This paper proposes the YOLOv4-3 target detection model, which was based on the YOLOv4 network model combined with CBAM module and the K-means++ clustering algorithm to optimize the YOLOv4 model in order to improve the detection accuracy of cracks in the target detection task. The construction of the improved YOLOv4-3 crack detection model was finally completed, and experimental analysis and evaluation were carried out. The YOLOv4 algorithm was used to study, analyze, and demonstrate that the attention module is not embedded in every location of the network to improve performance. To analyze and understand the impact of the CBAM module on the YOLOv4 model when embedded in different positions of the model, the CBAM module was installed in the Neck, Backbone, and Head regions of the model to verify the stability, noise suppression, and feature extraction of the cracked regions of the improved model. The improved target detection model was tested for comparison. The results showed that the mAP of the improved YOLOv4-3 network was 2.96% higher than that of YOLOv4. Although the performance of the YOLOV4-3 model proposed in this paper has been improved more than that of YOLOv4, there are still many problems that need to be further improved in the future. For example, YOLOV4-N2 had the highest accuracy after embedding the CBAM module, but the accuracy decreased after superimposing the CBAM module in the later stage. For this special case, the robustness of the model can be enhanced by introducing different attention mechanisms in future experiments. This study verified the feasibility of the model, but the experiments in special cases still need to be further improved.

Author Contributions: Funding acquisition, conceptualization, and methodology, L.L.; data acquisition, J.Z.; model building, experiments, data analysis, and writing—original draft preparation, B.F.; writing—review and editing, L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of Shanghai, China (funding number 19ZR1418800). The authors are grateful for their financial support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Silva, L.A.; Sanchez San Blas, H.; Peral García, D.; Sales Mendes, A.; Villarubia González, G. An Architectural Multi-Agent System for a Pavement Monitoring System with Pothole Recognition in UAV Images. *Sensors* **2020**, *20*, 6205. [CrossRef]
2. Lee, J.; Nam, B. Effects of pavement surface conditions on traffic crash severity. *J. Transp. Eng.* **2015**, *11*, 1–11. [CrossRef]
3. Li, Y.T.; Qin, Y.H.; Wang, H.; Xu, S.; Li, S. Study of Texture Indicators Applied to Pavement Wear Analysis Based on 3D Image Technology. *Sensors* **2022**, *22*, 4955. [CrossRef] [PubMed]
4. Hu, G.X.; Hu, B.L.; Yang, Z.; Huang, L.; Li, P. Pavement Crack Detection Method Based on Deep Learning Models. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5573590. [CrossRef]
5. Yu, J.M.; Lee, C. Survival Model-Based Economic Evaluation of Preventive Maintenance Practice on Asphalt Pavement. *J. South China Univ. Technol. Nat. Sci.* **2012**, *40*, 133–137.
6. Subirats, P.; Dumoulin, J.; Legeay, V.; Barba, D. Automation of Pavement Surface Crack Detection Using the Continuous Wavelet Transform. In Proceedings of the 2006 IEEE International Conference on Image Processing, Atlanta, GA, USA, 8–11 October 2006; pp. 3037–3040.
7. Nguyen, T.S.; Avila, M.; Begot, S. Automatic Detection and Classification of Defect on Road Pavement Using Anisotropy Measure. In Proceedings of the 2009 17th European Signal Processing Conference, Glasgow, UK, 24–28 August 2009; pp. 617–621.
8. Dollar, P.; Zitnick, C.L. Fast Edge Detection Using Structured Forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1558–1570. [CrossRef]
9. Akay, B. A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Appl. Soft Comput.* **2013**, *13*, 3066–3091. [CrossRef]
10. Vo, A.V.; Truong-Hong, L.; Laefer, D.F.; Bertolotto, M. Octree-based region growing for point cloud segmentation. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 88–100. [CrossRef]
11. Wang, Y.; Zhang, J.Y.; Liu, J.X.; Zhang, Y.; Chen, Z.P.; Li, C.G.; He, K.; Bin Yan, R. Research on Crack Detection Algorithm of the Concrete Bridge Based on Image Processing. *Procedia Comput. Sci.* **2019**, *154*, 610–616. [CrossRef]
12. Hong, X.J. Based on Fractional Differential Enhancement New Model of Pavement Crack. *J. Highw. Transp. Res. Dev.* **2016**, *33*, 83–87.
13. Nnolim, U.A. Automated crack segmentation via saturation channel thresholding, area classification and fusion of modified level set segmentation with Canny edge detection. *Heliyon* **2020**, *6*, e05748. [CrossRef] [PubMed]
14. Hoang, N.D.; Nguyen, Q.L.; Tran, V.-D. Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network. *Autom. Constr.* **2018**, *94*, 203–213.
15. Sheng, L. Method of shadow pavement crack extraction based on improved local threshold segmentation. *Wirel. Internet Technol.* **2018**, *20*, 112–113.
16. Zhang, D.; Li, Q.; Chen, Y.; Cao, M.; He, L.; Zhang, B. An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection. *Image Vis. Comput.* **2017**, *57*, 130–146. [CrossRef]
17. Song, M.; Cui, D.; Yu, C.; An, J.; Chang, C.-I. Crack Detection Algorithm for Photovoltaic Image Based on Multi-Scale Pyramid and Improved Region Growing. In Proceedings of the 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Chongqing, China, 27–29 June 2018; pp. 128–132.
18. Li, W.; Sun, Z.Y. Pavement Crack Type Judgment Method Based on Three-dimensional Pavement Data. *China J. Highw. Transp.* **2015**, *28*, 21–28.

19. Mohan, A.; Poobal, S. Crack detection using image processing: A critical review and analysis. *Alex. Eng. J.* **2017**, *57*, 787–798. [CrossRef]
20. Yao, Y.; Tung, S.; Glisic, B. Crack detection and characterization techniques—An overview. *Struct. Control Health Monit.* **2015**, *21*, 1387–1413. [CrossRef]
21. Xiang, X.; Zhang, Y.; El Saddik, A. Pavement Crack Detection Network Based on Pyramid Structure and Attention Mechanism. *IET Image Process.* **2020**, *14*, 1580–1586. [CrossRef]
22. Tsai, Y.C.; Kaul, V.; Mersereau, R.M. Critical Assessment of Pavement Distress Segmentation Methods. *J. Transp. Eng.* **2010**, *136*, 11–19. [CrossRef]
23. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
24. Girshick, R. Fast R-CNN. In Proceedings of the International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
25. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
26. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 27–30 June 2016; pp. 779–788.
27. Chen, L.L.; Zhang, Z.D.; Peng, L. Fast single shot multibox detector and its application on vehicle counting system. *IET Intell. Transp. Syst.* **2018**, *12*, 1406–1413. [CrossRef]
28. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
29. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
30. Li, M.; Zhang, Z.; Lei, L.; Wang, X.; Guo, X. Agricultural Greenhouses Detection in High-Resolution Satellite Images Based on Convolutional Neural Networks: Comparison of Faster R-CNN, YOLO v3 and SSD. *Sensors* **2020**, *20*, 4938. [CrossRef] [PubMed]
31. Morera, Á.; Sánchez, Á.; Moreno, A.B.; Sappa, Á.D.; Vélez, J.F. SSD vs. YOLO for Detection of Outdoor Urban Advertising Panels under Multiple Variabilities. *Sensors* **2020**, *20*, 4587. [CrossRef]
32. da Silva, D.Q.; dos Santos, F.N.; Sousa, A.J.; Filipe, V. Visible and Thermal Image-Based Trunk Detection with Deep Learning for Forestry Mobile Robotics. *Imaging* **2021**, *7*, 176. [CrossRef] [PubMed]
33. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
34. Yang, F. A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5. *Remote Sens.* **2021**, *13*, 1619.
35. Chen, W.; Ju, C.; Li, Y.; Hu, S.; Qiao, X. Sugarcane Stem Node Recognition in Field by Deep Learning Combining Data Expansion. *Appl. Sci.* **2021**, *11*, 8663. [CrossRef]
36. Umair, M.; Farooq, M.U.; Raza, R.H.; Chen, Q.; Abdulhai, B. Efficient Video-based Vehicle Queue Length Estimation using Computer Vision and Deep Learning for an Urban Traffic Scenario. *Processes* **2021**, *9*, 1786. [CrossRef]
37. Zhang, B.; Sun, C.-F.; Fang, S.-Q.; Zhao, Y.-H.; Su, S. Workshop Safety Helmet Wearing Detection Model Based on SCM-YOLO. *Sensors* **2022**, *22*, 6702. [CrossRef]
38. Li, S.; Gu, X.; Xu, X.; Xu, D.; Zhang, T.; Liu, Z.; Dong, Q. Detection of concealed cracks from ground penetrating radar images based on deep learning algorithm. *Constr. Build. Mater.* **2021**, *273*, 121949. [CrossRef]
39. Wong, M.A. Algorithm AS 136: A K-Means Clustering Algorithm. *J. R. Stat. Soc.* **1979**, *28*, 100–108.
40. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 3–19.
41. Zhang, Y.X.; Huang, J.; Cai, F. On Bridge Surface Crack Detection Based on an Improved YOLO v3 Algorithm. *IFAC Pap.* **2020**, *53*, 8205–8210. [CrossRef]
42. Yang, B.; Gao, Z.; Gao, Y.; Zhu, Y. Rapid Detection and Counting of Wheat Ears in the Field Using YOLOv4 with Attention Module. *Agronomy* **2021**, *11*, 1202. [CrossRef]
43. Arthur, D.; Vassilvitskii, S. k-means++: The Advantages of Careful Seeding. In Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007; pp. 1027–1035.
44. He, K.M.; Zhang, X.Y.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. In Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence, Zurich, Switzerland, 6–12 September 2014; pp. 1904–1916.
45. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
46. Sun, X.Q.; Huang, Q.; Li, Y.; Huang, Y. An Improved Vehicle Detection Algorithm based on YOLOV3. In Proceedings of the 2019 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), Xiamen, China, 16–18 December 2019; pp. 1445–1550.
47. Xue, Y.; Ju, Z. Multiple pedestrian tracking under first-person perspective using deep neural network and social force optimization. *Optik* **2021**, *240*, 166981. [CrossRef]

48. Li, H.; Deng, L.; Yang, C.; Liu, J.; Gu, Z. Enhanced YOLO v3 Tiny Network for Real-time Ship Detection from Visual Image. *IEEE Access* **2021**, *9*, 16692–16706. [CrossRef]
49. Qu, Z.; Zhu, F.; Qi, C. Remote Sensing Image Target Detection: Improvement of the YOLOv3 Model with Auxiliary Networks. *Remote Sens.* **2021**, *13*, 3908. [CrossRef]
50. Wang, K.; Liu, M.; Ye, Z. An advanced YOLOv3 method for small-scale road object detection. *Appl. Soft Comput.* **2021**, *112*, 107846. [CrossRef]
51. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. [CrossRef]