mathematics

MDPI

Article

An Improved YOLOv5 Crack Detection Method Combined with a Bottleneck Transformer

Gui Yu [1,2,3,4] and Xinglin Zhou [1,3,4,*]

1 Key Laboratory of Metallurgical Equipment and Control Technology, Ministry of Education, Wuhan University of Science and Technology, Wuhan 430081, China; yugui@hgnu.edu.cn
2 School of Mechatronic and Intelligent Manufacturing, Huanggang Normal University, Huanggang 438000, China
3 Hubei Key Laboratory of Mechanical Transmission and Manufacturing Engineering, Wuhan University of Science and Technology, Wuhan 430081, China
4 School of Machinery and Automation, Wuhan University of Science and Technology, Wuhan 430081, China
* Correspondence: zxl65@163.com

Abstract: Efficient detection of pavement cracks can effectively prevent traffic accidents and reduce road maintenance costs. In this paper, an improved YOLOv5 network combined with a Bottleneck Transformer is proposed for crack detection, called YOLOv5-CBoT. By combining the CNN and Transformer, YOLOv5-CBoT can better capture long-range dependencies to obtain more global information, so as to adapt to the long-span detection task of cracks. Moreover, the C2f module, which is proposed in the state-of-the-art object detection network YOLOv8, is introduced to further optimize the network by paralleling more gradient flow branches to obtain richer gradient information. The experimental results show that the improved YOLOv5 network has achieved competitive results on RDD2020 dataset, with fewer parameters and lower computational complexity but with higher accuracy and faster inference speed.

Keywords: YOLOv5; crack detection; Bottleneck Transformer; deep learning

MSC: 68T07

check for updates

Citation: Yu, G.; Zhou, X. An Improved YOLOv5 Crack Detection Method Combined with a Bottleneck Transformer. *Mathematics* **2023**, *11*, 2377. https://doi.org/10.3390/math11102377

Received: 21 April 2023
Revised: 15 May 2023
Accepted: 18 May 2023
Published: 19 May 2023

1. Introduction

For most transportation agencies, maintaining high-quality road surfaces is one of the keys to maintaining road safety. Cracks are common pavement diseases that seriously affect the road and traffic safety. Timely detection of road cracks is of great significance for preventing pavement damage and maintaining traffic safety [1].

Early manual visual inspection methods are tedious, time-consuming, subjective, error-prone, unsafe, and obstructive to traffic. Some other traditional methods require the use of 3D lidar or acoustic wave detection to obtain road condition assessments, which are uneconomical, slow, and difficult to deploy [2]. To overcome these disadvantages, automatic detection methods combining image sensors and computer vision algorithms have gradually become the mainstream methods.

Computer vision-based methods use various algorithms to extract features from images to detect cracks. Early methods include thresholding [3–5], edge detection [6–8], wavelet transform [9], mini path selection [10,11], etc. With the rise of machine learning, researchers applied it to crack detection, such as the artificial neural network (ANN) [12,13], support vector machine (SVM) [14–16], random structure forest [17], AdaBoost [18], and so on.

However, all the above methods heavily rely on manual feature extraction. For different scenes and different lighting conditions, different feature extraction models need to be designed. In the complex and changeable road environment, it is difficult to use a unified feature model to effectively extract the features cracks, resulting in poor detection robustness [19].

To achieve automatic feature extraction, the Deep Convolutional Neural Networks (DCNN) is applied to crack detection. The current popular crack detection methods based on DCNN mainly include semantic segmentation methods and object detection methods. The semantic segmentation methods usually employ an encoder–decoder structure to classify each pixel to obtain precise crack regions, such as FCN [20,21], Unet [22], Deep-Crack [23], etc. However, Zhuang et al. pointed out that the success of deep learning models in semantic segmentation comes at the cost of a heavy computation burden [24]. Furthermore, the datasets used in current semantic segmentation research only contain road surface images, without any other scenes outside the road. Such image data collection is usually performed by special inspection vehicles or regular vehicles specially modified with camera brackets so that the cameras face the ground. The labeling cost of these datasets is also very high.

The object detection method is an economical choice when precise measurement of crack size is not required. The object detection method using the vehicle-mounted mobile phone camera is low-cost, simple, and efficient and can be deployed on any ordinary car. Moreover, the scenes contained in its dataset are more complex and diverse. Therefore, this paper focuses on crack detection using an object detection method based on a vehicle-mounted mobile phone camera.

YOLOv5 [25] is a single-stage object detection model with four versions: YOLOv5s, YOLOv5m, YOLO5l, and YOLO5x. The network structures of the four versions are exactly the same, and the two parameters of depth_multiple and width_multiple are used to achieve different network depths and network widths. Its structural feature is to use the methods of Cross Stage Partial (CSP) and Spatial Pyramid Pooling-Fast (SPPF) in the Backbone network and use the methods of Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) in the Neck network. In addition, the Mosaic data augmentation, adaptive anchor frame calculation, and adaptive image scaling techniques are used to improve training effect. YOLOv5 has been widely welcomed by academic and engineering communities since its release. The USC-InfoLab team reaped the GRDDC'2020 championship using YOLOv5 [26], which also proved the effectiveness of YOLOv5 for crack detection. Therefore, we designed an improved network based on YOLOv5 to detect cracks.

Cracks are usually long in length and narrow in width, and their length is much greater than width, showing a slender shape in space. This structure feature of the crack makes the task of crack detection require more long-range dependencies to obtain contextual information.

A Bottleneck Transformer is proposed by Srinivas et al. [27]; by using a Multi-Head Self-Attention mechanism, it can obtain better long-range dependencies than DCNN. Additionally, this is exactly what is needed for the crack detection task. Inspired by YOLOv5 and the Bottleneck Transformer, this paper proposes an improved YOlOv5 network combined with the Bottleneck Transformer. Our main contributions can be summarized as follows.

1. We modified YOLOv5 in combination with Bottleneck Transformer and proposed an end-to-end pavement crack detection network for efficient detection of crack regions.
2. The C2f module proposed in the state-of-the-art object detection model YOLOv8 is introduced to optimize the network. In addition, we compared the effect of introducing the C2f module at different locations in the model on the performance of network.
3. We achieved competitive results on the evaluation dataset with fewer parameters and lower Giga Floating-point Operations Per second (GFLOPs).

The rest of the paper is organized as follows: Section 2 reviews the previous work on pavement crack object detection based on deep learning. Then, in Section 3 we describe the network architecture of our model and the specific composition of modules. Next, in Section 4 we perform experimental verification and discuss our method, including its ablation study. Finally, in Section 5, we summarize our work and point out future research plans.

2. Related Work

2.1. Deep Convolutional Nerual Network Methods

To overcome the problem of manual feature extraction in traditional methods, deep learning methods automatically extract various features of cracks. In practice, object detection methods are divided into two categories: one-stage algorithms and two-stage algorithms.

The two-stage algorithms first generate a series of candidate bounding boxes as samples and then classify the samples through the convolutional neural network. Typical representatives of such algorithms are Fast R-CNN [28], Faster R-CNN [29], Cascade R-CNN [30], etc. Nie et al. [31] proposed a crack detection model based on Faster R-CNN, using the transfer learning method of parameter fine-tuning to realize the detection of cracks, looseness, deformation, and other pavement diseases. Hascoet et al. [32] used Faster-RCNN for crack detection and improved detection performance using techniques such as label smoothing and also presented their efforts in deploying their model on local road networks. Vishwakarma et al. [33] introduced the tuning strategy of Faster-RCN based on deep residual network (Resnet) and feature pyramid network (FPN) backbones of different depths. Furthermore, they compared it with a single-stage YOLOv5 model with a cross-stage part network (CSPNet) backbone. Pei et al. [34] applied Cascade R-CNN to crack detection and proposed a Consistency Filtering Mechanism (CFM) with a self-supervised approach to fully utilize the available unlabeled data.

However, the one-stage algorithms take object detection as a regression task, which directly regresses the bounding box and predicts the categories of multiple locations in the entire image to obtain more comprehensive information [35], such as SSD [36], YOLO series [37–39], Centernet [40], EfficeientDet [41], etc. Mandal et al. [42] proposed an automated pavement distress analysis system based on the YOLOv2. Shao et al. [43] adopted the YOLOv3 framework and proposed a PTZ camera-based image-processing pipeline for crack size measurement. Zhang et al. [44] combined YOLOv3 with Adaptive Spatial Feature Fusion (ASFF) to improve the adaptability to cracks of different scales. The literature [45] adopted YOLOv4 as the base network to detect cracks and proposed a generative adversarial network (GAN) for data augmentation. At the same time, the effects of tricks such as data augmentation, transfer learning, and optimized anchors and their combinations were evaluated. Mandal et al. [46] used YOLOv4 for crack detection and studied the impact of various backbones. Liu et al. [47] conducted ensemble learning of YOLOv4 and Fast R-CNN. Hu et al. [19] employed YOLOv5 to detect cracks and compared the model size, detection speed, and accuracy performance of the four versions of YOLOv5. Guo et al. [1] proposed to use the lightweight network MobileNetv3 to replace the backbone network of the original YOLOv5s model to reduce model parameters and GFLOPs and adopted Coordinate Attention to optimize the model.

Although these CNN-based methods have achieved good results, the CNN receptive field is usually small, which is not conducive to capturing global features. However, due to the large span and slender features of cracks, global features are exactly what is needed for the crack detection task. Therefore, researchers began to explore the combination of CNN and Transformer that can better capture global features.

2.2. CNN and Transformer Combined Methods

In recent years, transformers have made great breakthroughs in CV. Dosovitskiy et al. proposed the Vision Transformer (ViT), which used Multiple Self-Attention (MSA) to capture long-range dependencies [48]. The success of CNN relies on its two inherent inductive biases, translation invariance and local correlation, while Vision Transformer architectures usually lack this property, resulting in the fact that a large amount of data is usually required to exceed the performance of CNN. The limited receptive field of CNN makes it difficult to capture global information, while the Transformer can capture long-distance dependencies. Therefore, after the emergence of ViT, many excellent works have tried to combine CNN and Transformer, allowing the networks to inherit the advantages of CNN and Transformer, so as to preserve both global features and local features as much as possible. Jing et al. [49] and Zhu et al. [50] integrated Transformer into YOLOv5 to process UAV captured images. Lei et al. [51] replaced the YOLOv5 backbone network with Transformer for underwater object detection. Xiang et al. [35] designed an improved YOLOv5 by embedding the Vision Transformer into the backbone network, which can detect cracks accurately with fast inference speed. The methods of CNN combined with transformer can capture the long-range dependencies of crack objects while maintaining real-time performance, which is also the focus of this paper.

3. Method

3.1. Network Architecture

Considering the excellent performance of YOLOv5 in crack detection, we designed an improved YOLOv5m network, named YOLOv5-CBot, by combining Bot-transformer and C2f module, as shown in Figure 1.

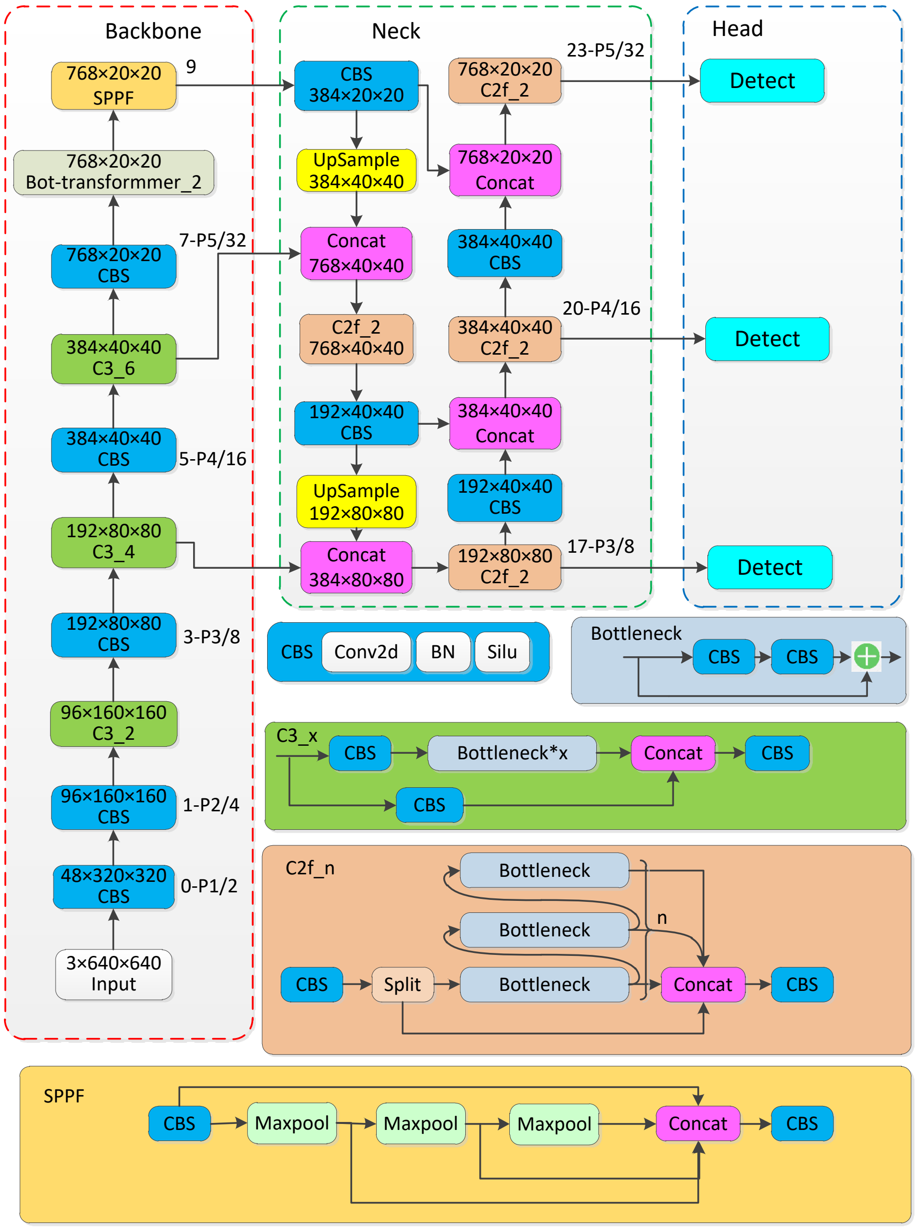


Figure 1. Improved network architecture based on YOLOv5m introduced by the Bot-transformer Block and C2f module. *x means there are x number of same block stacked.

The YOLOv5-CBot mainly consists of three parts, the backbone network (Backbone), bottleneck layer network (Neck), and detection layer (Head). The Backbone part of the network is mainly composed of a standard convolution module (CBS), C3 module, Bot-transformer module, and spatial pyramid pooling module (SPPF). The Neck part of the network consist of the CBS module, C2f module, and a series of concatenating operations. There are two main differences between YOLOv5-CBoT and the original YOLOv5.

1. The last C3 module in the original YOLOv5 backbone network, that is, the C3 module before the SPPF module, is replaced by the Bot-transformer module.
2. All C3 modules in the Neck networks are replaced by C2f modules.

The CBS module is a basic unit that constitutes the entire network, mainly completing the downsampling, dimensionality rise and reduction, normalization, and nonlinearity process of feature maps.

The C3 module is an important feature extraction module, which consists of three CBS modules and several stacked Bottlenecks. C3_x means there are x number of bottleneck block stacked. As shown in Figure 1, after the feature map is input into the C3 module, it is divided into two branches. One branch passes through CBS and Bottlenecks, the other passes through one CBS only. Finally, the two branches are concatenated and then passed through a CBS module. There are two CBS modules in the Bottleneck block; the first CBS is a 1×1 convolution, which reduces the channel to half, and the second is a 3×3 convolution, which doubles the number of channels. Reducing the dimension first helps the convolution kernel to better understand the feature information, and increasing the dimension will help extract more detailed features. Finally, the residual structure is used to add the input and output to avoid the problem of gradient disappearance. The main function of C3 is to increase the depth and receptive field of the network by stacking basic CBS module and residual connections and improve the ability of feature extraction.

The actual pavement cracks mainly include longitudinal cracks and transverse cracks, which occupy large areas in the horizontal or vertical direction, respectively. For such cracks, the receptive field of conventional convolutions is too small to cover the entire crack region. Although the receptive field can be expanded by stacking multiple convolutional layers, it is still insufficient for feature extraction of large-span crack objects. To this end, we introduced a Bot-transformer structure to capture the long-range dependencies of crack objects and significantly extract contextual semantics. The details of Bot-transformer will be introduced in Section 3.2.

The function of the neck part of the network is to combine shallow graphic features with deep semantic features to obtain more complete features. As can be seen from Figure 1, the left part of neck upsamples the feature map by interpolation, making the scale of feature map gradually larger to facilitate the fusion of feature maps from different network layers of backbone; the right side of neck continues to downsample, on the one hand to obtain feature maps of different scales and the other hand to better fuse shallow graphic features with deep semantic features.

Furthermore, we use the latest effective module C2f introduced in YOLOv8 to replace the C3 module at the neck part of network. The C2f module refers to the ideas of the C3 module and ELAN [52]. As can be seen from Figure 1, there are two differences between C2f and C3.

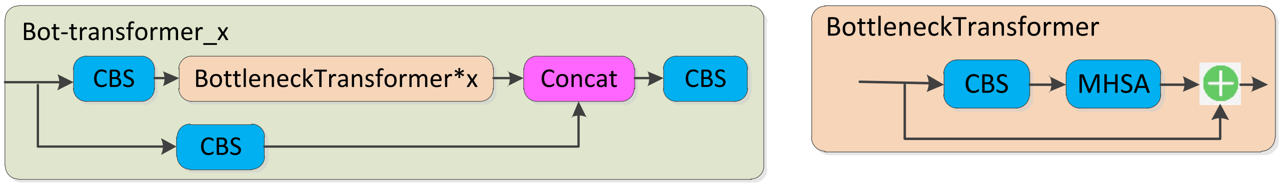
1. C2f reduces a standard convolutional module (CBS), which contributes to the lightweight of the network.
2. In addition to the serial stacking of the Bottleneck module similar to C3, a parallel concatenating operation of the Bottleneck module is added in C2f, which helps to obtain rich gradient flow information.

Based on the above two points, the role of C2f is to obtain richer gradient flow information while ensuring lightweight.

3.2. Transformer Structure

3.2.1. Bot-Transformer Module

BoTNet, proposed by Srinivas et al. [27], is a simple but powerful backbone that incorporates self-attention into various computer vision tasks. By only replacing the 3×3 convolution with Multi-Head Self-Attention (MHSA) in ResNet, without any other changes, the network performance for instance segmentation and object detection is significantly improved, while the network parameters are also reduced. The Bot-Transformer module proposed in this paper, as shown in Figure 2a, is modified from the C3 module of YOLOv5, by changing its Bottleneck block to Bottleneck Transformer block.



(**a**) (**b**)

Figure 2. The architecture of Bot-transformer block. (**a**) BottleneckTransformer*x means there are x number of BottleneckTransformer block stacked, and each BottleneckTransformer is showed in (**b**). (**b**) Structural diagram of submodule BottleneckTransformer of Bot-trsnformer block.

The difference between the Bottleneck Transformer block and original Bottleneck block is that the standard 3×3 convolution module (CBS) is replaced by MHSA block, as show in Figure 2b, which is consistent with the idea of BotNet. Therefore, the key to Bottleneck Transformer lies in the use of MHSA, which will be introduced in Section 3.2.2.

3.2.2. MHSA Block

As described in Figure 2, the MHSA block is the core component of Bottleneck Transformer. The structure of the MHSA block is shown in Figure 3. In the figure, q, k, v, and r represent query, key, value and position encodings, respectively. The input size of MHSA is H $\times$ W $\times$ d, where H, W, and d, respectively, represent the height and width of the input feature matrix and the dimension of a single token.

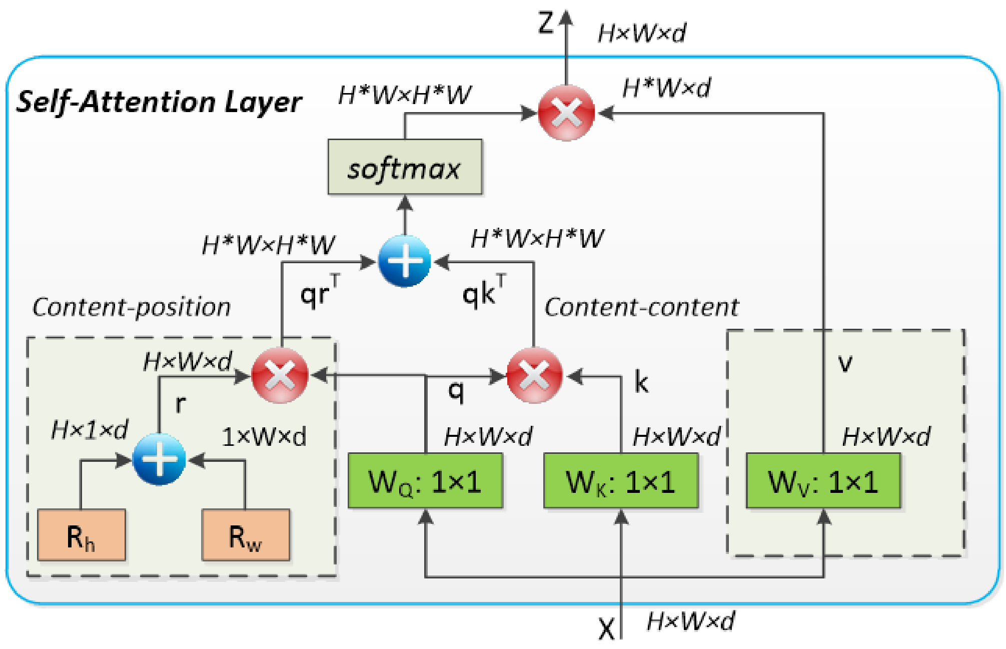


Figure 3. Multi-Head Self-Attention (MHSA) Layer used in the Bot-transformer Block.

It should be noted that the MHSA here is different from the Multi-Head Self-Attention of the traditional transformer, such as Vision Transformer (ViT). The biggest difference is that the left content-position module introduces two-dimensional position encoding. Two learnable parameter vectors, R_h and R_w, represent the position encoding of the height and width of different positions, respectively. They are added through the broadcast mechanism, and the encoding of the (i, j) position is the sum of two d-dimensional vectors R_{hi} and R_{wj}. Matrix multiplication is performed between the position encoding and the query matrix to obtain a part of the attention, which is qr^T. In the content-content part, the query matrix and the Key matrix are multiplied to obtain another part of the attention, namely qk^T. Finally, the attention logits are $qk^T + qr^T$. The entire MHSA layer can be expressed as:

$$\text{softmax}\,(qk^T + qr^T) \times v \tag{1}$$

4. Experiment

4.1. Dataset

The public dataset we used in this research is the RDD2020 dataset [53], which consists of road damage images from three countries: Japan, the Czech Republic, and India. The dataset now contains 21,041 images, 10,506 from Japanese pavement, 7706 from Indian pavement, and 2829 from Czech pavement, covering eight road conditions including longitudinal cracks, transverse cracks, crosswalk blur, etc. Since this study focuses on crack detection in pavement maintenance, we only consider four types of crack damage: longitudinal cracks (D00), transverse cracks (D10), alligator cracks (D20), and Pothole (D40). Due to a large number of images in the dataset not containing the four types of detection targets, we screened the dataset and finally retained 12,195 images, 1072 from the Czech Republic, 3223 from India, and 7900 from Japan, all of which contain four types of detection targets: D00, D10, D20, and D40. The same filtering process is also adopted in [1,54]. The number of various detection targets contained in the crack images of the three countries is shown in Table 1.

Table 1. Relevant statistics of experimental dataset.

Country	D00	D10	D20	D40
Czech	764	300	129	154
India	1109	60	1758	1530
Japan	2297	2240	4714	1390
Total	4670	2600	6601	3074

4.2. Evaluation Metrics

We use precision and recall as basic evaluation metrics in this study, and they can be defined as:

$$P_r = \frac{TP}{TP + FP} \tag{2}$$

$$R_e = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2 \times P_r \times R_e}{P_r + R_e} \tag{4}$$

where TP indicates that the prediction category of the prediction bounding box is consistent with the ground truth and IOU between them is greater than 0.5, FP presents that the prediction bounding box appears in the region without objects, and FN means that the network fails to predict the ground truth object [35]. The F1-score takes both precision and recall into account, so it can comprehensively reflect the overall performance of the network. It is calculated by taking the harmonic average of two indicators, as shown in Equation (4). Like most object detection studies, mAP0.5 and mAP0.5:0.95 are used as important evaluation indicators of network performance. In addition, the model parameter amount and GFLOPs are adopted to measure model complexity, and the Frames Per Second (FPS) is used to evaluate inference speed.

4.3. Experimental Environment and Parameter Settings

The experimental environment for this research is an NVIDIA GeForce RTX3090 GPU, Intel i9-10900X CPU, and 32G RAM. The model algorithm is implemented by Pytorch deep learning framework on the Windows10 operating system.

In the training process, the input image size of the network is set to 640 $\times$ 640, and the model is trained for 100 epochs using the SGD optimizer with batch size of 16, initial learning rate of 0.01, momentum of 0.937, and weight decay of 0.0005. We use a warmup strategy in the first three epochs and then use cosine learning rate attenuation strategy to perform gradient descent more smoothly.

4.4. Results

On the RDD2020 dataset, we compared our experimental results with some excellent work. It can be seen from Table 2 that our algorithm is significantly better than YOLOv3, YOLOv4, and other algorithms.

Table 2. Performance comparison of various methods.

Methods	Precision	Recall	F1	mAP0.5	mAP0.5:0.95	Params/M	GFLOPs	FPS
Faster-RCNN	0.529	0.607	0.565	0.513	0.220	60.1	108.6	29
Cascade-RCNN	0.494	0.656	0.564	0.548	0.250	87.9	110.6	24
YOLOv3	0.608	0.612	0.610	0.627	0.308	58.7	154.6	48
YOLOv4-CSP	0.606	0.595	0.600	0.631	0.317	50.1	119.1	49
YOLOv5	0.620	0.606	0.613	0.633	0.321	44.0	107.7	59
YOLOv7	0.629	0.601	0.615	0.640	0.338	34.8	103.2	85
CenterNet	0.500	0.626	0.556	0.510	0.215	14.4	19.3	70
Tood	0.615	0.578	0.596	0.601	0.275	53.2	71.8	17
Ours	0.654	0.591	0.621	0.646	0.331	22.9	55.9	85

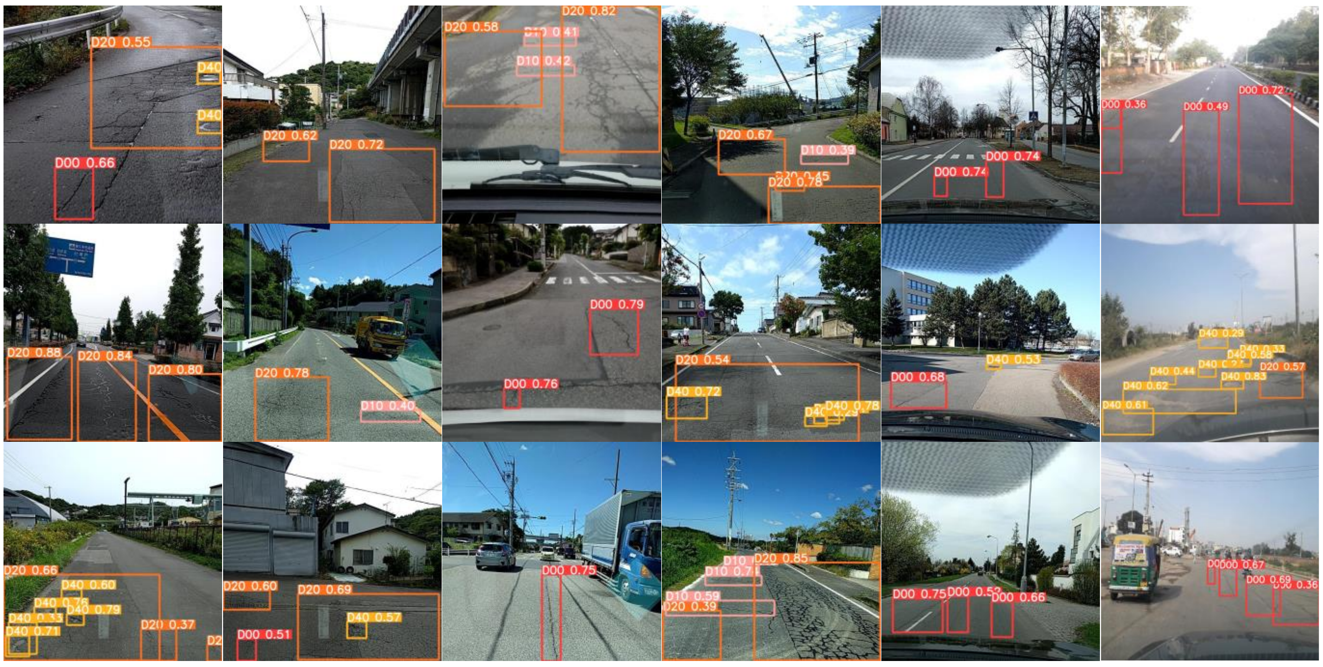
YOLOv5 has been proved to be very effective in crack detection by the USC-InfoLab team. Our algorithm is improved based on YOLOv5m. As shown in Table 2, compared with the original YOLOv5 algorithm, our parameter amount and GFLOPs have been reduced by nearly half, while the F1-score, mAP0.5, and mAP0.5:0.95 are increased by 1.3%, 2.1%, and 3.1%, respectively, and the FPS is increased 1.4 times. Even compared with the state-of-the-art YOLOv7 algorithm, although our parameter amount is reduced by 34% and GFLOPs is reduced by 46%, our F1-score is still increased by 1%, and mAP0.5 is increased by 0.9%. FPS is equivalent to YOLOv7; only the mAP0.5:0.95 indicator is slightly worse. The experimental results show that our algorithm achieves competitive results, with fewer parameters and lower computational complexity but with higher accuracy and faster inference speed.

Figure 4 shows the visualization effect of the detection. Our algorithm can effectively detect cracks of various scales and types in complex and diverse scenes.

4.5. Ablation Study

4.5.1. Effect of C2f Module Location on Network Performance

To verify the effect of introducing C2f modules at different locations on network performance, we designed a set of ablation experiments, as shown in Table 3. C2f-Backbone means that the C3 modules in the backbone network are replaced by the C2f module, while keeping the neck network unchanged. C2f-head represents that C3 modules in the neck network are replaced by a C2f module, keeping the C3 module in the backbone network unchanged and so on. C2f-all stands for that all the C3 modules in the backbone network and neck network are replaced by the C2f module. The experimental results show that replacing C3 with the C2f module in the neck network achieves the best performance with fewer GFLOPs and higher FPS.



(**a**) Japan (**b**) Czech (**c**) India

Figure 4. Visualization results of some detection sample.

Table 3. Performance comparison of introducing C2f module in the different locations of the network.

Methods	Precision	Recall	F1	mAP0.5	mAP0.5:0.95	Params/M	GFLOPs	FPS
C2f-Backbone	0.604	0.602	0.603	0.628	0.320	21	60.6	80
C2f-head (ours)	0.654	0.591	0.621	0.646	0.331	22.9	55.9	85
C2f-all	0.637	0.575	0.604	0.630	0.319	25.4	70.1	74

4.5.2. Effect of the Number of Head of MHSA on Network Performance

To verify the effect of the head number of MHSA block on network performance, we designed a set of ablation experiments, as shown in Table 4. The experimental results show that the network performance is the best when the head number is four. Moreover, the change in the number of heads has relatively little influence on the experimental results. This may suggest that the MHSA structure itself is more important.

Table 4. Performance comparison of different head numbers of MHSA.

Number of Heads	Precision	Recall	F1	mAP0.5	mAP0.5:0.95
Head = 4(ours)	0.654	0.591	0.621	0.646	0.331
Head = 8	0.658	0.576	0.614	0.644	0.327
Head = 12	0.615	0.621	0.618	0.646	0.330
Head = 16	0.652	0.588	0.618	0.642	0.326

4.5.3. Selection of Training Hyperparameters

Training hyperparameters generally have a greater impact on the experimental results. To this end, we designed a set of ablation experiments and compared the two sets of default training hyperparameters of YOLOv5, low-augment hyperparameters group (LAHG) and medium-augment hyperparameters group (MAHG), as shown in Table 5. Considering that these two sets of parameters are optimized on the coco dataset by YOLOv5, we also used parameter evolution although it consumes a huge amount of training time. We used 25 iterations of parameter evolution training based on the low-augment parameter group and the high-augment parameter group, respectively. Each such experiment took 25 times as long as the general experiment. The experimental results show that the hyperparameter evolution based on LAHG can improve the network performance, but the parameter evolution based on MAHG does not. The performance of MAHG is still stronger than the hyperparameter evolution based on LAHG, although the latter consumes more than 20 times the training time. The main reason may be that MAHG uses mixup data augment, but LAHG does not. This also shows that mixup data augment is very effective for YOLOv5. We then used the default MAHG of YOLOv5 for all later experiments.

Table 5. Performance comparison of different hyperparameters group.

Hyperparameters Group	Precision	Recall	F1	mAP0.5	mAP0.5:0.95	Training Time
LAHG	0.587	0.623	0.605	0.628	0.311	3 h 35 min
MAHG	0.654	0.591	0.621	0.646	0.331	4 h 3 min
evolve based on LAHG	0.641	0.593	0.616	0.637	0.321	89 h 35 min
evolve based on MAHG	0.618	0.608	0.613	0.634	0.334	101 h 15 min

5. Conclusions

With the rapid development of urban traffic, there is an urgent need for efficient and low-cost detection methods for pavement maintenance. In this paper, an improved YOLOv5 network combined with Bottleneck Transformer is proposed, which can capture long-range dependencies to adapt to the long-span, slender features of a crack object. In addition, the C2f module is introduced to further optimize the network. The experimental results show that compared with the original YOLOv5 algorithm, the F1 score of our algorithm is increased by 1.3%, the mAP0.5 is increased by 2.1%, the mAP0.5:0.95 is increased by 3.1%, and the inference speed is 1.4 times faster, but the parameter amount and GFLOPs are reduced by nearly half. To further improve our detection system, we plan to further expand the dataset, study more data augmentation methods, and investigate more attention mechanisms to deal with more complex pavement maintenance work.

Author Contributions: Conceptualization, G.Y.; methodology, G.Y.; software, G.Y.; validation, G.Y.; formal analysis, G.Y.; investigation, G.Y.; resources, G.Y. and X.Z.; data curation, G.Y.; writing—original draft preparation, G.Y.; writing—review and editing, G.Y.; visualization, G.Y.; supervision, X.Z.; project administration, X.Z.; funding acquisition, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant no. 51827812 and 51778509).

Data Availability Statement: Please contact Gui Yu (yugui@hgnu.edu.cn).

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

References

1. Guo, G.; Zhang, Z. Road Damage Detection Algorithm for Improved YOLOv5. *Sci. Rep.* **2022**, *12*, 15523. [CrossRef] [PubMed]
2. Quan, Y.; Sun, J.; Zhang, Y.; Zhang, H. The Method of the Road Surface Crack Detection by the Improved Otsu Threshold. In Proceedings of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 4–7 August 2019; pp. 1615–1620.
3. Oliveira, H.; Correia, P.L. Automatic Road Crack Segmentation Using Entropy and Image Dynamic Thresholding. In Proceedings of the 2009 17th European Signal Processing Conference, Glasgow, UK, 24–28 August 2009; pp. 622–626.
4. Tsai, Y.C.; Kaul, V.; Mersereau, R.M. Critical assessment of pavement distress segmentation methods. *J. Transp. Eng.* **2010**, *136*, 11–19. [CrossRef]
5. Li, P.; Wang, C.; Li, S.; Feng, B. Research on crack detection method of airport runway based on twice-threshold segmentation. In Proceedings of the 5th International Conference on Instrumentation and Measurement, Computer, Communication, and Control, IMCCC 2015, Qinhuangdao, China, 18–20 September 2015; pp. 1716–1720.

6. Santhi, B.; Krishnamurthy, G.; Siddharth, S.; Ramakrishnan, P.K. Automatic Detection of Cracks in Pavements Using Edge Detection Operator. *J. Theor. Appl. Inf. Technol.* **2012**, *36*, 199–205.
7. Yeum, C.M.; Dyke, S.J. Vision-Based Automated Crack Detection for Bridge Inspection. *Comput. Civ. Infrastruct. Eng.* **2015**, *30*, 759–770. [CrossRef]
8. Nisanth, A.; Mathew, A. Automated Visual Inspection of Pavement Crack Detection and Characterization. *Int. J. Technol. Eng. Syst.* **2014**, *6*, 14–20.
9. Zhou, J. Wavelet-Based Pavement Distress Detection and Evaluation. *Opt. Eng.* **2006**, *45*, 27007. [CrossRef]
10. Amhaz, R.; Chambon, S.; Idier, J.; Baltazart, V. Automatic Crack Detection on Two-Dimensional Pavement Images: An Algorithm Based on Minimal Path Selection. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2718–2729. [CrossRef]
11. Zou, Q.; Li, Q.; Zhang, F.; Xiong Qian Wang, Z.; Wang, Q. Path Voting Based Pavement Crack Detection from Laser Range Images. In Proceedings of the 2016 IEEE International Conference on Digital Signal Processing (DSP), Beijing, China, 16–18 October 2016; pp. 432–436.
12. Lee, B.J.; Lee, H.D. Position-Invariant Neural Network for Digital Pavement Crack Analysis. *Comput. Civ. Infrastruct. Eng.* **2004**, *19*, 105–118. [CrossRef]
13. Moon, H.G.; Kim, J.H. Inteligent Crack Detecting Algorithm on the Concrete Crack Image Using Neural Network. In Proceedings of the 28th ISARC, Seoul, Republic of Korea, 29 June–2 July 2011; pp. 1461–1467.
14. Gavilán, M.; Balcones, D.; Marcos, O.; Llorca, D.F.; Sotelo, M.A.; Parra, I.; Ocaña, M.; Aliseda, P.; Yarza, P.; Amírola, A. Adaptive Road Crack Detection System by Pavement Classification. *Sensors* **2011**, *11*, 9628–9657. [CrossRef]
15. O'Byrne, M.; Schoefs, F.; Ghosh, B.; Pakrashi, V. Texture Analysis Based Damage Detection of Ageing Infrastructural Elements. *Comput. Civ. Infrastruct. Eng.* **2013**, *28*, 162–177. [CrossRef]
16. Cha, Y.J.; You, K.; Choi, W. Vision-Based Detection of Loosened Bolts Using the Hough Transform and Support Vector Machines. *Autom. Constr.* **2016**, *71*, 181–188. [CrossRef]
17. Shi, Y.; Cui, L.; Qi, Z.; Meng, F.; Chen, Z. Automatic Road Crack Detection Using Random Structured Forests. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3434–3445. [CrossRef]
18. Cord, A.; Chambon, S. Automatic Road Defect Detection by Textural Pattern Recognition Based on AdaBoost. *Comput. Civ. Infrastruct. Eng.* **2012**, *27*, 244–259. [CrossRef]
19. 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]
20. Yang, X.; Li, H.; Yu, Y.; Luo, X.; Huang, T.; Yang, X. Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network. *Comput. Civ. Infrastruct. Eng.* **2018**, *33*, 1090–1109. [CrossRef]
21. Li, S.; Zhao, X.; Zhou, G. Automatic Pixel-Level Multiple Damage Detection of Concrete Structure Using Fully Convolutional Network. *Comput. Civ. Infrastruct. Eng.* **2019**, *34*, 616–634. [CrossRef]
22. Yu, G.; Dong, J.; Wang, Y.; Zhou, X. RUC-Net: A Residual-Unet-Based Convolutional Neural Network for Pixel-Level Pavement Crack Segmentation. *Sensors* **2023**, *23*, 53. [CrossRef]
23. Zou, Q.; Zhang, Z.; Li, Q.; Qi, X.; Wang, Q.; Wang, S. DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection. *IEEE Trans. Image Process.* **2019**, *28*, 1498–1512. [CrossRef]
24. Zhuang, J.; Yang, J.; Gu, L.; Dvornek, N. Shelfnet for Fast Semantic Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 847–856.
25. Available online: https://github.com/ultralytics/yolov5 (accessed on 5 March 2023).
26. Arya, D.; Maeda, H.; Kumar Ghosh, S.; Toshniwal, D.; Omata, H.; Kashiyama, T.; Sekimoto, Y. Global Road Damage Detection: State-of-the-Art Solutions. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 5533–5539.
27. Srinivas, A.; Lin, T.Y.; Parmar, N.; Shlens, J.; Abbeel, P.; Vaswani, A. Bottleneck Transformers for Visual Recognition. In Proceedings of the IEEE/CVF Conference on Computer vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 16514–16524.
28. Girshick, R.B. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
29. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
30. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into High Quality Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
31. Nie, M.; Wang, K. Pavement Distress Detection Based on Transfer Learning. In Proceedings of the 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, China, 10–12 November 2018; pp. 435–439.
32. Hascoet, T.; Zhang, Y.; Persch, A.; Takashima, R.; Takiguchi, T.; Ariki, Y. FasterRCNN Monitoring of Road Damages: Competition and Deployment. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 5545–5552.
33. Vishwakarma, R.; Vennelakanti, R. CNN Model Tuning for Global Road Damage Detection. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 5609–5615.

34. Pei, Z.; Lin, R.; Zhang, X.; Shen, H.; Tang, J.; Yang, Y. CFM: A Consistency Filtering Mechanism for Road Damage Detection. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 5584–5591.
35. Xiang, X.; Wang, Z.; Qiao, Y. An Improved YOLOv5 Crack Detection Method Combined with Transformer. *IEEE Sens. J.* **2022**, *22*, 14328–14335. [CrossRef]
36. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot Multibox Detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Springer: Cham, Switzerland, 2016.
37. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
38. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
39. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *arXiv* **2022**, arXiv:2207.02696.
40. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6568–6577.
41. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787.
42. Mandal, V.; Uong, L.; Adu-gyamfi, Y. Automated Road Crack Detection Using Deep Convolutional Neural Networks. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 5212–5215.
43. Shao, C.; Zhang, L.; Pan, W. PTZ Camera-Based Image Processing for Automatic Crack Size Measurement in Expressways. *IEEE Sens. J.* **2021**, *21*, 23352–23361. [CrossRef]
44. Zhang, R.; Shi, Y.; Yu, X. Pavement Crack Detection Based on Deep Learning. In Proceedings of the 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; pp. 7367–7372.
45. Zhang, X.; Xia, X.; Li, N.; Lin, M.; Song, J.; Ding, N. Exploring the Tricks for Road Damage Detection with A One-Stage Detector. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 5616–5621.
46. Mandal, V.; Mussah, A.R.; Adu-Gyamfi, Y. Deep Learning Frameworks for Pavement Distress Classification: A Comparative Analysis. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 5577–5583.
47. Liu, Y.; Zhang, X.; Zhang, B.; Chen, Z. Deep Network for Road Damage Detection. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 5572–5576.
48. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
49. Jing, Y.; Ren, Y.; Liu, Y.; Wang, D.; Yu, L. Automatic Extraction of Damaged Houses by Earthquake Based on Improved YOLOv5: A Case Study in Yangbi. *Remote Sens.* **2022**, *14*, 382. [CrossRef]
50. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision 2021, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.
51. Lei, F.; Tang, F.; Li, S. Underwater Target Detection Algorithm Based on Improved YOLOv5. *J. Mar. Sci. Eng.* **2022**, *10*, 310. [CrossRef]
52. Zhang, X.; Zeng, H.; Guo, S.; Zhang, L. Efficient Long-Range Attention Network for Image Super-Resolution. In *Computer Vision –ECCV 2022. ECCV 2022. Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2022; pp. 1–20.
53. Arya, D.; Maeda, H.; Ghosh, S.K.; Toshniwal, D.; Sekimoto, Y. RDD2020: An Annotated Image Dataset for Automatic Road Damage Detection Using Deep Learning. *Data Br.* **2021**, *36*, 107133. [CrossRef]
54. Jeong, D. Road Damage Detection Using YOLO with Smartphone Images. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 5559–5562.