Computing and Informatics, Vol. 42, 2023, 1339–1359, doi: 10.31577/cai_2023_6_1339

EDGE COMPUTING-BASED VEHICLE DETECTION IN INTELLIGENT TRANSPORTATION SYSTEMS

Hao Pan, Shaopeng Guan*, Xiaoyan Zhao, Yuewei Xue

School of Information and Electronic Engineering
Shandong Technology and Business University
264005 Yantai, China
e-mail: phao0622@163.com, konexgsp@gmail.com

Abstract. Vehicle detection in intelligent transportation systems usually adopts cloud computing mode. The increasing amount of traffic surveillance video has brought challenges to the storage, communication, and processing of intelligent transportation systems based on cloud computing models. In this paper, we propose a vehicle detection scheme based on edge computing. First, the traffic surveillance video is preprocessed at the edge device. Using the frame difference algorithm based on structural similarity (SSIM) to remove video redundant frames, and avoid repeated frames in the subsequent extracted key frame sequence. Then, a frame difference algorithm based on local maxima is used to extract key frames as the basis for subsequent vehicle detection. Finally, the YOLOv5s is improved and used for vehicle detection. The efficient channel attention mechanism (ECA) is introduced to enhance the important features of the vehicle and suppress the general features to strengthen the detection network's ability to identify vehicle targets. At the same time, the Focal loss function is introduced to solve the positive and negative sample imbalance problem and improve the detection speed. The experimental results show that the scheme has more advantages than the original YOLOv5s in terms of precision, recall, and mean average precision.

Keywords: Intelligent transportation systems, vehicle detection, edge computing, key frames, YOLOv5s

Mathematics Subject Classification 2010: 68U10

* Corresponding author

1 INTRODUCTION

In recent years, with the improvement of people's living standards, the number of social motor vehicles has grown rapidly. While enjoying the convenience brought by vehicles, people are also facing a series of problems such as traffic congestion and frequent accidents. Intelligent Transportation System (ITS), as a large-scale and comprehensive transportation management system, can greatly alleviate traffic congestion and effectively reduce the occurrence of traffic accidents [1].

The current intelligent transportation system mainly adopts the cloud computing mode, that is, the traffic images and videos are captured by camera equipment, and transmitted to the cloud computing center for subsequent processing and analysis [2]. With the rapid increase in the number of camera devices deployed, monitoring data exploded. A large amount of data transmission has brought huge pressure on the network bandwidth, resulting in delays in data transmission and other issues. At the same time, the processing of massive data will also greatly increase the computing load of the cloud computing center [3]. Therefore, ITS based on the cloud computing mode can no longer meet the needs of efficient and real-time traffic supervision.

Edge computing is a new computing paradigm that performs computational tasks close to the data source. It does not need to transmit all the data generated by the end device to the cloud computing center, and has the advantage of reducing network bandwidth pressure and transmission delay [4, 5]. Edge computing essentially uses the computing power of edge devices to migrate some computing tasks to edge devices for execution, thereby reducing the computing load of cloud computing centers [6]. With the development of information technology, the camera equipment in ITS gradually becomes intelligent and has computing power, and it becomes possible to migrate the task of vehicle detection to the camera equipment side.

Vehicle detection is one of the core tasks of ITS. The purpose is to detect vehicles from video image sequences and provide a basis for subsequent traffic supervision. Traditional vehicle detection methods mainly rely on manually extracting vehicle features and combining the features with a classifier for detection. The detection process is as follows: firstly, a sliding window is used on the image to be detected to obtain the candidate area. Then the features in the candidate area are extracted. Finally, the classifier is trained using the extracted features to detect the vehicle in the input image. The traditional vehicle detection method works well in terms of the accuracy of type recognition, but using sliding windows and manually extracting vehicle features has problems such as complex detection process, slow speed, and low efficiency [7].

Deep learning builds a machine learning architecture model with multiple layers, and after large-scale data training, it can automatically obtain more representative feature information and realize the classification and prediction of samples [8]. Convolutional neural network is one of the representative algorithms of deep learning, which can automatically extract features in vehicle images and can solve the shortage of traditional manual extraction of vehicle features [9]. This method only needs to use labeled vehicle images to train the network, so that it can learn the characteristics of the vehicle type, and then realize the detection of the vehicle. You only look once (YOLO) – is a target detection algorithm that uses convolutional neural networks to achieve fast target detection and recognition and maintain high accuracy by integrating target area prediction and target category prediction in a single neural network model [10]. Because of its advantages of small model size and fast detection speed, it is widely used in the field of object detection [11]. However, due to the influence of background, illumination, occlusion, and other factors, the robustness and accuracy of YOLO for vehicle detection need to be further improved [12].

In this paper, we design a vehicle detection scheme based on edge computing. Firstly, the preprocessing operation of redundant frame removal is performed on the surveillance video. Then, the key frames are extracted and the improved YOLOv5s are used to detect the vehicle.

Our main contributions are summarized as follows:

- A frame difference method based on local maxima is proposed to extract key frames in surveillance video. Using the inter-frame difference pixel intensity maxima of two frames as the key frames extraction criterion reduces the computational complexity while better maintaining the accuracy of key frames extraction.
- A vehicle detection method based on the improved YOLOv5s model is proposed. The efficient channel attention mechanism (ECA) is introduced to improve detection accuracy. At the same time, the introduction of the Focal loss function solves the problem of the imbalance between positive and negative samples, speeds up model convergence, and improves the detection speed.
- The scheme was evaluated in the public data sets. Experimental results in special cases such as dark light and occlusion show that the improved YOLOv5s model has better detection accuracy and higher accuracy.

The rest of the paper is organized as follows: Section 2 reviews related work; Section 3 details the method of key frame extraction and the improved YOLOv5s model; Section 4 evaluates the scheme through experiments; finally, Section 5 concludes the presented work.

2 RELATED WORKS

There exist a large number of redundant frames in surveillance videos, which can easily lead to repeated frames in the extracted key frames sequence [13]. To this end, Wan et al. [14] proposed a motion amplitude detection algorithm based on local spatiotemporal interest points. When the number and position of interest points in the video do not change, it is considered that the content of the video has not changed. Then, this feature is used to remove a large number of unchanging redundant frames existing in long videos. Pasandi and Nadeem [15] proposed to treat all network cameras as a whole and use the temporal and spatial correlation between cameras to eliminate redundant frames. Al-Ani and Hammouri [16] proposed a method based on frame difference to remove video redundant frames, deleting any frame between consecutive frames that is lower than the average value of the difference between frames. The above methods use the spatio-temporal characteristics between video frames for redundant frame removal and achieve better results, but these methods do not take into account the structural similarity between video frames, and some redundant frames still exist.

In order to solve the problem of time-consuming and laborious vehicle detection on a frame-by-frame basis, key frame extraction is required for the surveillance video after removing redundancy [17]. To this end, Gharbi et al. [18] proposed a key frame extraction method based on the combination of local interest point description and repeatability graph clustering. The selection of key frames is performed using graph clustering based on the principle of close modularity, and the experimental results prove that the key frames proposed by this method are representative. Wu et al. [19] proposed a clustering method based on high peak search to integrate important attributes of the video, and clustered similar frames into clusters, and the central value of the cluster was regarded as a key frame. Wolf [20] proposed an optical flow calculation method for identifying motion local minima in shots. The optical flow and a simple motion metric are first calculated for each frame, then the metric is analyzed as a function of time, and finally the frame with the smallest metric is selected as the key frame. Most of the above schemes use clustering or optical flow calculation methods. Although representative key frames can be extracted, there is a problem of high computational complexity.

Carrying out vehicle detection based on the extracted key frames sequence can improve detection efficiency and provide help for real-time handling of traffic accidents and relief of traffic congestion [21]. With the rise of deep learning, detection algorithms based on deep learning have received extensive attention from researchers. Target detection algorithms based on deep learning can be divided into two categories, one is a two-stage detection algorithm represented by R-CNN [22] and Fast R-CNN [23], which use the region candidate network to extract candidate target information. The other is an end-to-end one-stage detection algorithm represented by YOLO [10] and SSD [24]. The accuracy of the two-stage detection algorithm is relatively high, but the speed is much slower than the one-stage detection algorithm. For application scenarios with high real-time requirements such as ITS, a one-stage detection algorithm is often used [25]. However, as the size of the image increases, the operating efficiency of the one-stage detection algorithm SSD will decrease significantly, so the target detection method based on YOLO is more widely used [26]. Li et al. [27] proposed a vehicle detection algorithm based on YOLOv3. The algorithm introduces the channel attention mechanism and spatial attention mechanism into the feature extraction network to improve the feature extraction ability of the network. Bao et al. [28] proposed a YOLOv5-based component defect detection method, which incorporates a coordinate attention module in the original network that can simultaneously capture spatial interactions with precise location information and help the network locate the target of interest more accurately. The above approach combines the attention mechanism with YOLO to improve the overall detection capability, but does not take into account the fact that the amount of parameters becomes larger resulting in the model not applying to resource-constrained edge devices. To this end, Wan et al. [29] proposed an improved YOLOv3 vehicle detection algorithm and deployed it on edge devices. The algorithm introduces the feature extraction network in YOLOv3 into the dense splice block to reduce the model parameters of the feature extraction network structure, which reduces the training complexity and improves the training speed. Liang et al. [30] proposed a vehicle detection algorithm based on the YOLOv3 model, which ensures its efficiency on edge devices by pruning the feature extraction network and compressing the feature fusion network. The above scheme achieves the reduction of the number of network parameters through model compression and satisfies the requirements of edge device deployment, but does not consider the impact of dark light and occlusion on vehicle detection accuracy in practical applications.

3 PROPOSED WORK

Considering the time-consuming and labor-intensive problem of direct vehicle detection on traffic surveillance video, we designed a vehicle detection scheme to improve the efficiency and accuracy of vehicle detection. Aiming at the time-consuming and laborious problem of frame-by-frame detection, we designed a key frame extraction method to improve detection efficiency. In addition, the YOLOv5s model was improved to improve the vehicle detection accuracy of the model.

3.1 Key Frames Extraction Method

We use the local inter-frame difference maxima algorithm to extract key frames from surveillance video. In order to avoid repeated frames from the extracted key frames sequence, we need to remove redundant frames before key frame extraction.

The structural similarity of surveillance video frames will affect the removal of redundant frames, so we use the SSIM-based inter-frame difference algorithm to remove redundant frames [31]. Firstly, the differential operation is performed on two adjacent frames, and then the SSIM values of the two images are calculated. SSIM calculation process is as follows:

Step 1: Given two images, design an $N \times N$ localized viewport on the images.

Step 2: Calculate the SSIM value for the image of the window position, and the resulting value is the local value.

Step 3: Move the window one pixel at a time and calculate the local values until the local values are calculated for each position of the image.

Step 4: Average all local values, which is the SSIM value of the two images.

When the two images do not meet the differential operation conditions and the SSIM value is 1, then the two images will be judged to be redundant, and one of the images needs to be deleted.

Given two images x and y, the SSIM values of the two images are shown in Equation (1) [32]:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{1}$$

where μ_x is the mean of x, μ_y is the mean of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y, and σ_{xy} is the covariance of x and y. $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ is a constant used to maintain stability. L is the dynamic range of pixel values. $k_1 = 0.01$, $k_2 = 0.03$. The SSIM value ranges from -1 to 1. When the two images are the same, the SSIM value is equal to 1.

The purpose of key frame extraction is to find a set of images from a raw video to represent the video content [33]. Analyzing the acquired key frames can solve the time-consuming and labor-intensive problem of analyzing the video frame-by-frame. Therefore, we need to perform key frame extraction before vehicle detection. Key frames usually refer to 2-3 frames that play a decisive role among several frames constituting a video. However, the selection of key frames is a very subjective operation. In addition, the type of video and the length of the video will also affect the selection of key frames. The selection of video key frames usually meets the following aspects [29]:

Repeatability: Low repeatability of selected video key frames.

Continuity: The selected video key frames must ensure the continuity of the content.

Regularity: The number of video key frames is as small as possible and the information is as discrete as possible.

Representativeness: The selected video key frames can represent the video content.

Considering that the frame difference method is not sensitive to scene changes such as light, and the algorithm has low complexity and good stability, we adopt a key frame extraction algorithm based on the maximum value of the local inter-frame difference [34]. The two frames are differenced to obtain the inter-frame differential pixel intensity, which is used to measure the magnitude of the change between the two frames. When a certain frame in the video has a large change from the previous frame, it is considered a key frame and extracted. The classic inter-frame difference key frame extraction is to compare the distance D between the two image frames with the set threshold T. If $D > T$, the latter of the two frames will be used as the key frame [35]. The distance D between image frames is shown in Equation (2) [13]:

$$D(f_i, f_j) = \sum_{x,y} |f_i(x, y) - f_j(x, y)|, \tag{2}$$

where f_i and f_j are any two image frames as candidate key frames, x and y are the positions of pixels in the selected two image frames.

The classic inter-frame difference algorithm uses the threshold as the key frame extraction standard. Improper setting of the threshold will easily lead to inaccurate key frame extraction. We use the inter-frame differential intensity maxima as key frame extraction criteria. The size of the local neighborhood is chosen to be 1, that is, the size of each inter-frame difference value and its left and right neighborhood values are checked. If they are larger than the values of their left and right neighborhoods, they are local extremes, and the image frame corresponding to that inter-frame difference value is selected as the key frame.

Video frames generated in traffic surveillance videos are susceptible to noise, resulting in inaccurate inter-frame difference values, which affect key frame extraction results. The Hanning window function can effectively remove random noise. We use the Hanning window to smooth the inter-frame difference value of the image sequence and avoid extracting several frames under similar image sequences as key frames at the same time [36].

3.2 Vehicle Detection

After the redundant frame removal and key frame extraction are performed on the surveillance video, the frame sequence representing the video content is obtained, and vehicle detection can be performed accordingly.

Considering that the vehicle detection based on edge computing belongs to a distributed structure, we adopt a distributed deep learning architecture based on edge computing, and migrate the training and reasoning stages to the edge nodes to reduce the communication overhead generated during the training process [29]. Its architecture is shown in Figure 1.

The architecture migrates the vehicle detection model to the edge node, and uses the acquired key frames as the data of the vehicle detection model. The model in the edge node is trained, and the training result is uploaded to the edge server. The server is responsible for parameter updates, model synchronization, and other operations until the end of training.

The YOLO series algorithm has relatively high detection speed and accuracy and has been widely used for vehicle detection [37]. YOLOv5 is the fifth version of YOLO. It also has the advantages of high detection accuracy and fast inference speed, and its weight file is small, which is suitable for deployment on edge devices for real-time detection [38]. As a derivative version of YOLOv5, YOLOv5s

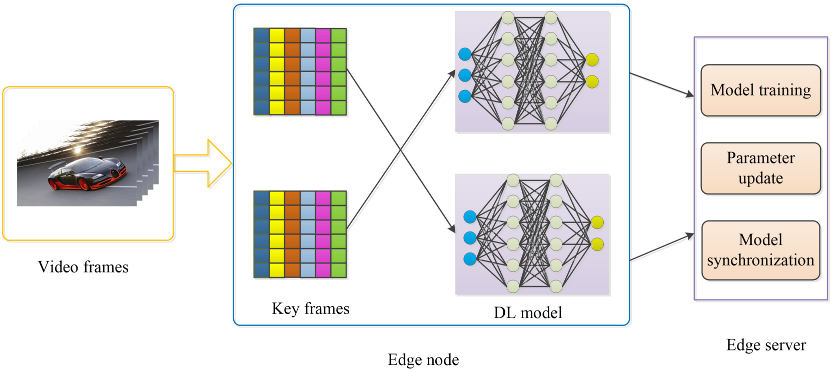


Figure 1. Distributed deep learning architecture based on edge computing

has fewer parameters and a lighter model. In this paper, we use YOLOv5s as the benchmark model for vehicle detection, and its network structure is shown in Figure 2.

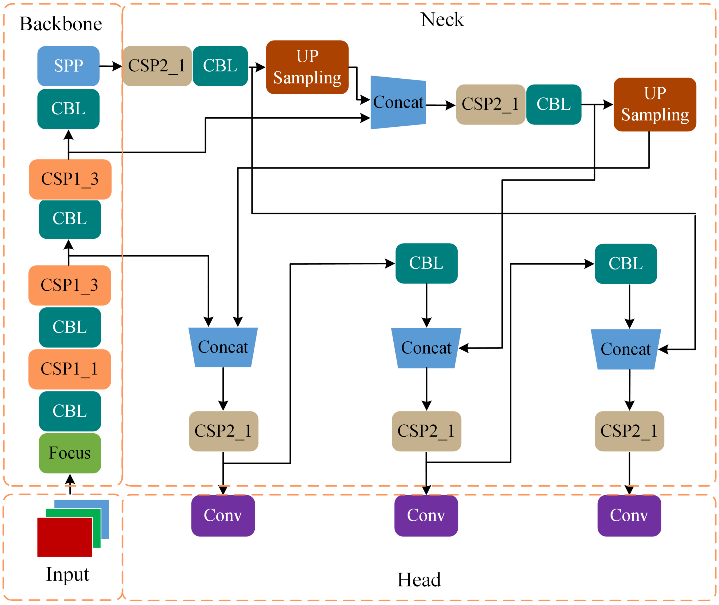


Figure 2. YOLOv5s network structure

The YOLOv5s model is divided into four parts: Input, Backbone, Neck and Head. Among them, the Input end includes an adaptive image scaling function. Backbone uses modules such as Focus and Spatial Pyramid Pooling (SPP) to extract image features. The Neck part is to fuse the image features extracted by the Backbone and give three different scales of feature maps, which are used to make predictions. The Head part inherits the head structure of YOLOv3, including prediction information including object coordinates, category and confidence.

Vehicles in traffic surveillance videos have special conditions such as occlusion and denseness, which will affect the accuracy of detection. The integration of the attention module into the convolutional neural network can effectively enhance the rich representation ability of the convolutional neural network, and can better capture various discriminative features of vehicle targets [39]. Therefore, we introduced the ECA attention module in YOLOv5s to improve the problem of low detection accuracy in special cases such as occlusion and denseness.

ECA is an efficient channel attention mechanism. It improves channel attention based on Squeeze-and-Excitation Networks (SENet), which can reduce the complexity of the model while improving performance, with less computational cost Improving the performance of detection networks [40]. The structure of the ECA module is shown in Figure 3.

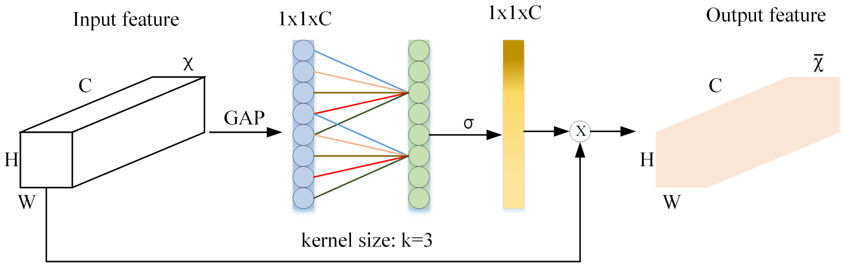


Figure 3. ECA module structure

ECA removes the Fully Connected (FC) layer of SENet, uses Global Average Pooling (GAP) to compress the spatial features of the feature map, and uses one-dimensional convolution to perform channel feature learning on the compressed feature map. Where, k is the convolution kernel size, which represents the coverage of local cross-channel interactions, C is the number of channels in the feature map, while X and $\bar{X}$ distinguish between input and output features. Finally, use the Sigmoid function to generate the weight ratio of each channel, and then combine the original input features with the channel weights to obtain features with channel attention.

The introduction of different positions by the attention mechanism will produce different effects. We designed two fusion methods for different positions, that is, integrating ECA into the Backbone and Neck parts respectively. As a result, two network models were generated, which are respectively denoted as YOLOv5s_A and YOLOv5s_B.

The ECA is fused to the Backbone part to form YOLOv5s_A. The main function of Backbone is to extract the deep features in the image through a relatively deep convolutional network. As the number of network layers deepens, the width of the feature map becomes smaller and smaller. Cross Stage Partial (CSP) aggregates different hierarchies, and the ECA is placed after the CSP for channel attention reconstruction of the feature maps at different locations, and their structure is shown in Figure 4.

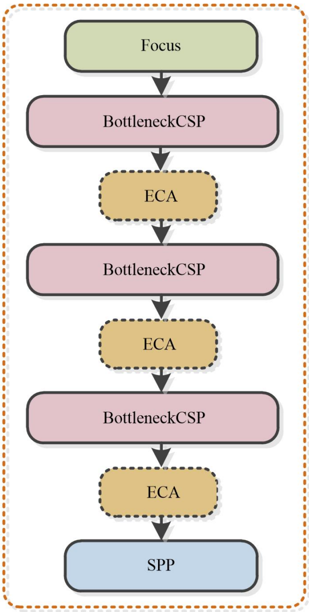


Figure 4. The structure of the Backbone after adding the ECA module

In Figure 4, the Backbone part introduces ECA modules and puts these modules after the CSP structure to obtain vehicle features with channel attention.

Fuse the ECA module into the Neck part to form YOLOv5s_B. The PAN and FPN structures in Neck can transfer semantic information and positioning information from bottom to top, and integrate deep and shallow information through four Concat operations, so the ECA module is placed after Concat. The structure after adding the ECA module is shown in Figure 5.

In Figure 5, four ECA modules are added to the Neck part, and each ECA module is placed after Concat to reconstruct the channel attention of the feature map and pay more attention to the vehicle information in the channel domain.

In the one-stage target detection algorithm, the imbalance between positive and negative samples is more prominent. The percentage of background in the traffic road images is significantly larger than the percentage of vehicles, and the loss

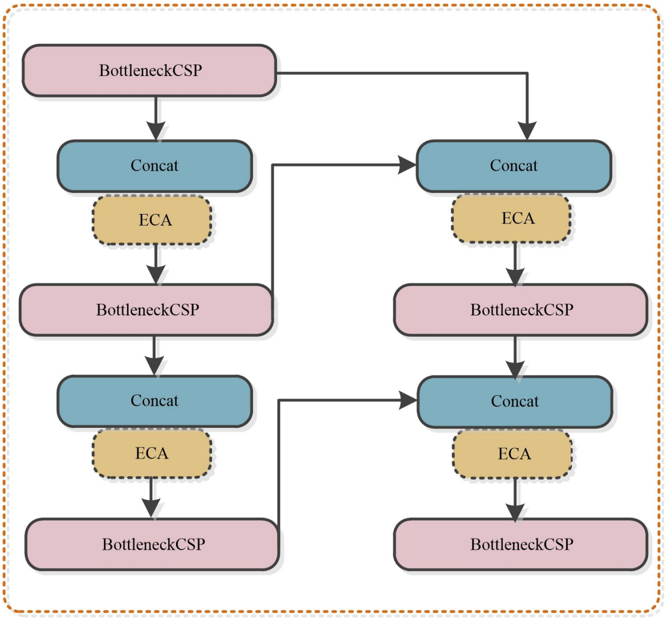


Figure 5. The structure of the Neck after adding the ECA module

function yields mostly negative sample background loss. And most of the negative sample background is simple and easy to divide, which has almost no effect on the convergence of the model. We introduce the Focal loss function, using parameters to balance the effect of positive and negative samples on the loss and divide the samples into hard-to-classify samples and easy-to-classify samples. By reducing the weight of easy-to-classify samples, the model is made to focus more on hard-to-classify samples during training to optimize the training process. The Focal loss function is shown in Equation (3) [41]:

$$Focal\ loss(p, y, \alpha, \gamma) = \begin{cases} -\alpha(1-p)^{\gamma}\log(p), & y = 1, \\ -(1-\alpha)p^{\gamma}\log(1-p), & y = 0, \end{cases} \tag{3}$$

where α is the balance factor, and p is the probability of output through the Sigmoid activation function. y is the real sample label, which takes the value of 0 or 1. $(1-p)^{\gamma}$ and p^{γ} are the modulation factors. The balance factor is responsible for solving the problem of uneven positive and negative samples in the detection model, and the modulation factor controls the impact of the difference between difficult and easy samples on the loss.

4 EXPERIMENT AND ANALYSIS

Considering the limited resources of edge devices, we used an ordinary laptop to evaluate the performance of the proposed scheme. The configuration of the laptop was Intel(R) Core(TM) i7-11800H with 16 GB of memory.

4.1 Key Frames Extraction Experiment

The video dataset used in the experiment comes from Cityflow [42], which is a traffic camera dataset that contains more than 3 hours of synchronized high-definition video from 40 cameras at 10 intersections. The dataset covers a wide range of scenes, perspectives, vehicle models, and Urban traffic flow conditions. To verify the effectiveness of the key frame extraction algorithm based on the local inter-frame difference maximum value, we compared it with K-means and DT algorithms on this dataset. Both algorithms are widely used in key frame extraction. The experiment uses F1-measure as the evaluation standard, and its calculation equation is shown in Equation (4) [43]:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}, \tag{4}$$

where *Precision* represents the percentage of key frames selected by the algorithm that are true key frames, *Recall* represents the ratio of the number of key frames extracted by the algorithm to the actual number of key frames in the total number of frames. The smaller the total number of key frames selected by the algorithm, the higher the accuracy of key frame matching and the lower the corresponding recall rate. If the total number of selected key frames is too large, the recall rate will increase and the accuracy rate of key frame matching will decrease. The $F1$ value is a compromise between matching accuracy and recall, and is used to balance the two. The experimental results of the key frame selection algorithm are shown in Table 1.

Algorithms	**Precision (%)**	**Recall (%)**	**F1-measure (%)**
K-means	73.0	59.0	65.0
DT	67.0	54.0	60.0
Our method	76.0	58.0	66.0

Table 1. Performance evaluation results of key frame extraction algorithm

It can be seen from Table 1 that our accuracy is higher than the other two methods, because our method selects fewer key frames and has a higher accuracy of key frame matching. The total number of key frames selected by the K-means method is higher than ours, so its recall rate is slightly higher than our method, but our method has achieved the better performance in the balance between the two.

4.2 Vehicle Detection Experiment

4.2.1 Experimental Environment and Dataset

The experiment is based on the Ubuntu 18.04 operating system, using the Pytorch 1.8.0 framework, training through the NVIDIA 3070 graphics card. And the video memory is 8 GB. Python version is 3.8 and CUDA version is 11.1.1.

We use the UA-DETRAC dataset for vehicle object detection and tracking at the University at Albany, USA as the experimental object [44]. The dataset consists of 10 hours of video captured with a Cannon EOS 550D camera at 24 different locations in Beijing and Tianjin, China. Videos were recorded at 25 frames per second with a resolution of 960 × 540 pixels. There are more than 140 000 frames in this dataset, with 8 250 vehicles manually annotated, and a total of 1.21 million labeled object bounding boxes. To avoid too small image changes between adjacent frames, we use the method of selecting one frame every 10 frames to obtain 13 000 images, and select 10 000 images with large traffic flow as the experimental data set.

4.2.2 Evaluation Metrics

The experiment uses Precision, Recall, and mAP to measure the detection performance of the model, and the specific calculation formula is as follows [45].

Precision and Recall are shown in Equations (5) and (6):

$$Precision = \frac{TP}{TP + FN} \times 100\,\%, \tag{5}$$

$$Recall = \frac{TP}{TP + FP} \times 100\,\%, \tag{6}$$

where TP (True Positives) is the number of vehicles detected correctly, FN (False Negatives) is the number of vehicles not detected correctly, and FP (False Positives) is the number of vehicles detected incorrectly.

The AP value is the area enclosed by the horizontal and vertical coordinates of the accuracy rate and the recall rate, and its calculation equation is:

$$AP = \int_0^1 P(R)\,\mathrm{d}R. \tag{7}$$

Average the AP values of all categories to get the mAP value, and its calculation equation is:

$$mAP = \frac{\sum AP}{m}. \tag{8}$$

In the Equations (7) and (8), m represents the number of categories of all samples, P is the accuracy rate, R is the recall rate, and $P(R)$ is the accuracy rate and recall rate curve.

4.2.3 Experimental Results

We randomly divide the dataset into training and test sets in the ratio of 9:1, and none of the experiments use pre-trained models. The training process uses the same parameter configuration, and the input image size is 640 × 640. The experimental results are shown in Table 2.

Model	Precision (%)	Recall (%)	mAP (%)
Origin YOLOv5s	92.6	86.2	90.5
YOLOv5s_A	92.9	84.6	88.5
YOLOv5s_B	93.1	87.1	91.9

Table 2. Comparison of detection results of fusion ECA

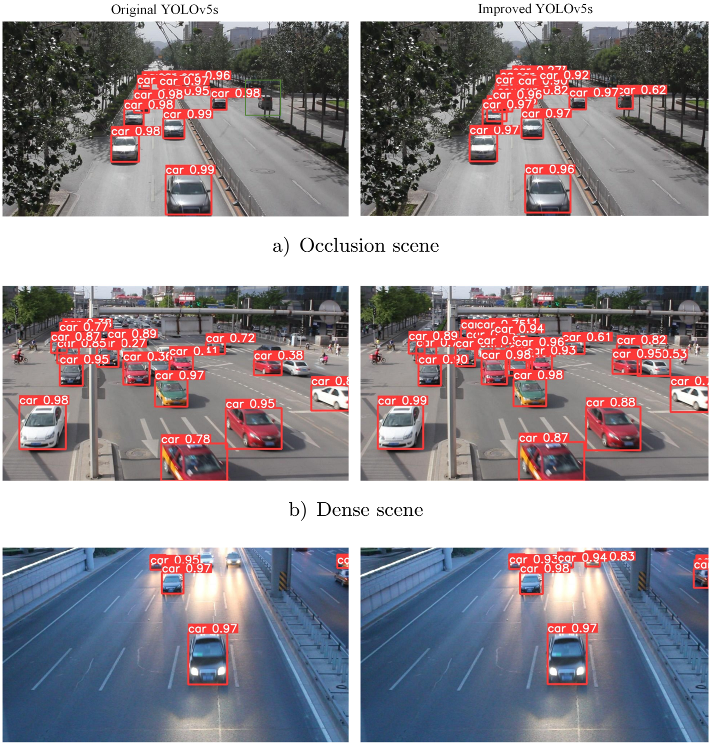
It can be seen from Table 2 that the Precision value of YOLOv5s_A has increased compared with the original YOLOv5s model, but the Recall and mAP values have decreased significantly, and not all vehicles in the image are detected. This is because in Backbone, the features extracted by the network are not sufficient, and only the Precision indicator is improved. Compared with the original network, YOLOv5s_B has improved in all three evaluation indicators, and mAP is also the highest among the three models. This is because in Neck, the network fuses the deep and shallow feature maps. On this basis, it performs attention fusion on the feature maps, recalibrates the importance of different channel features, and achieves the best detection results. Therefore, we use YOLOv5s_B as the basic model for vehicle detection.

In addition, we compared the proposed algorithm with some lightweight detection algorithms on the UA-DETRAC dataset, and the results are shown in Table 3.

Method	Precision (%)	Recall (%)	mAP (%)	Parameters	FLOPs (G)	FPS
YOLOv5s	92.6	86.2	90.5	7.01 M	15.9	111
YOLOv5s_SE	92.8	85.4	89.7	7.22 M	16.2	97
YOLOv5s_CBAM	91.6	86.2	90.6	7.05 M	16.0	102
YOLOv3-tiny	92.2	76.6	88.8	–	–	–
YOLOv4-tiny	90.2	84.4	89.1	5.92 M	–	–
Out scheme	93.1	87.1	91.9	7.02 M	16.0	125

Table 3. Performance comparison of vehicle detection algorithms

It can be seen from Table 3 that the improved YOLOv5s algorithm is ahead of other lightweight detection algorithms in terms of Precision, Recall and mAP values. The parameter amount of the improved algorithm is slightly higher than that of the original algorithm, but it is better than the original algorithm in terms of performance indicators. This is because the ECA we use is a lightweight way to obtain the interaction information between channels, which reduces the complexity of the model while improving performance. At the same time, the proportion of a large number of simple negative samples in training is reduced by introducing the Focal loss function, so that the model can focus more on difficult-to-classify samples during training. In Table 3, FLOPs (Floating-point Operations) are used to measure the complexity of the algorithm, and FPS (Frames Per Second) is the number of images that can be processed in each second. It can be found from Table 3 that the computational effort of the improved algorithm is slightly higher than the original algorithm, but the FPS value is better than the original algorithm. Therefore, the performance overhead of our proposed scheme is satisfying the edge computing condition.



a) Occlusion scene

b) Dense scene

c) Dark light scene

Figure 6. Vehicle detection results in special scenes

Finally, in the UA-DETRAC data set, we selected pictures in three special scenes of occlusion, dense, and dark light to detect vehicles, and the results are shown in Figure 6.

In Figure 6, the detection results of the original YOLOv5s are on the left, and the detection results of the improved YOLOv5s are on the right. It can be seen that the original YOLOv5s missed detection in special scenes. The improved YOLOv5s algorithm not only avoids missed detection in special scenarios, but also maintains a high detection accuracy. This is because ECA is added to our improved model, which can make the network pay more attention to the target to be detected. And the improved YOLOv5s has added the Focal loss function, which can detect a large number of small-scale vehicle targets in a dense street environment, while the original YOLOv5s algorithm cannot effectively identify small-scale targets.

5 CONCLUSIONS

We propose a vehicle detection scheme that can be deployed on edge devices. First, the traffic surveillance video is preprocessed, and redundant frames with similar structures are removed by using SSIM combined with the frame difference algorithm. Then, the key frames are extracted using the inter-frame difference method based on the local maximum value, and the maximum value of the pixel intensity difference between the two frames of images is used as the extraction standard. Finally, we added the ECA module to the Neck part of YOLOv5s to recalibrate the importance of different channel features. This makes the improved network more focused on the extraction of vehicle information, to improve the accuracy of vehicle detection. Experimental results show that our proposed scheme is superior to other models in terms of detection accuracy and recall rate, and still has high detection accuracy in special scenarios such as low light and occlusion.

REFERENCES

[1] Sumalee, A.—Ho, H. W.: Smarter and More Connected: Future Intelligent Transportation System. IATSS Research, Vol. 42, 2018, No. 2, pp. 67–71, doi: 10.1016/j.iatssr.2018.05.005.

[2] Husain, A. A.—Maity, T.—Yadav, R. K.: Vehicle Detection in Intelligent Transport System Under a Hazy Environment: A Survey. IET Image Processing, Vol. 14, 2020, No. 1, pp. 1–10, doi: 10.1049/iet-ipr.2018.5351.

[3] Cao, K.—Liu, Y.—Meng, G.—Sun, Q.: An Overview on Edge Computing Research. IEEE Access, Vol. 8, 2020, pp. 85714–85728, doi: 10.1109/ACCESS.2020.2991734.

[4] Khan, W. Z.—Ahmed, E.—Hakak, S.—Yaqoob, I.—Ahmed, A.: Edge Computing: A Survey. Future Generation Computer Systems, Vol. 97, 2019, pp. 219–235, doi: 10.1016/j.future.2019.02.050.

[5] Ai, Y.—Peng, M.—Zhang, K.: Edge Computing Technologies for Internet of Things: A Primer. Digital Communications and Networks, Vol. 4, 2018, No. 2, pp. 77–86, doi: 10.1016/j.dcan.2017.07.001.

[6] Shi, W.—Sun, H.—Cao, J.—Zhang, Q.—Liu, W.: Edge Computing – A New Computing Model in the Internet of Everything Era. Journal of Computer Research and Development, Vol. 54, 2017, No. 5, pp. 907–924, doi: 10.7544/issn1000-1239.2017.20160941 (in Chinese).

[7] Wu, X.—Song, X.—Gao, S.—Chen, C.: Review of Target Detection Algorithms Based on Deep Learning. Transducer and Microsystem Technologies, Vol. 40, 2021, No. 2, pp. 4–7 (in Chinese).

[8] LeCun, Y.—Bengio, Y.—Hinton, G.: Deep Learning. Nature, Vol. 521, 2015, No. 7553, pp. 436–444, doi: 10.1038/nature14539.

[9] Gu, J.—Wang, Z.—Kuen, J.—Ma, L.—Shahroudy, A.—Shuai, B.—Liu, T.—Wang, X.—Wang, G.—Cai, J.—Chen, T.: Recent Advances in Convolutional Neural Networks. Pattern Recognition, Vol. 77, 2018, pp. 354–377, doi: 10.1016/j.patcog.2017.10.013.

[10] Redmon, J.—Divvala, S.—Girshick, R.—Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[11] Jiang, P.—Ergu, D.—Liu, F.—Cai, Y.—Ma, B.: A Review of Yolo Algorithm Developments. Procedia Computer Science, Vol. 199, 2022, pp. 1066–1073, doi: 10.1016/j.procs.2022.01.135.

[12] Pan, Q.—Zhang, H.: Key Algorithms of Video Target Detection and Recognition in Intelligent Transportation Systems. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 34, 2020, No. 9, Art. No. 2055016, doi: 10.1142/S0218001420550162.

[13] Olatunji, I. E.—Cheng, C. H.: Video Analytics for Visual Surveillance and Applications: An Overview and Survey. In: Tsihrintzis, G. A., Virvou, M., Sakkopoulos, E., Jain, L. C. (Eds.): Machine Learning Paradigms: Applications of Learning and Analytics in Intelligent Systems. Springer, Cham, Learning and Analytics in Intelligent Systems, Vol. 1, 2019, pp. 475–515, doi: 10.1007/978-3-030-15628-2_15.

[14] Wan, S.—Xu, X.—Wang, T.—Gu, Z.: An Intelligent Video Analysis Method for Abnormal Event Detection in Intelligent Transportation Systems. IEEE Transactions on Intelligent Transportation Systems, Vol. 22, 2020, No. 7, pp. 4487–4495, doi: 10.1109/TITS.2020.3017505.

[15] Pasandi, H. B.—Nadeem, T.: CONVINCE: Collaborative Cross-Camera Video Analytics at the Edge. 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 2020, pp. 1–5, doi: 10.1109/PerComWorkshops48775.2020.9156251.

[16] Al-Ani, M. S.—Hammouri, T. A.: Video Compression Algorithm Based on Frame Difference Approaches. International Journal on Soft Computing (IJSC), Vol. 2, 2011, No. 4, pp. 67–79, doi: 10.5121/ijsc.2011.2407.

[17] Bilal, K.—Erbad, A.: Edge Computing for Interactive Media and Video Streaming. 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC), 2017, pp. 68–73, doi: 10.1109/FMEC.2017.7946410.

[18] Gharbi, H.—Bahroun, S.—Zagrouba, E.: Key Frame Extraction for Video

Summarization Using Local Description and Repeatability Graph Clustering. Signal, Image and Video Processing, Vol. 13, 2019, No. 3, pp. 507–515, doi: 10.1007/s11760-018-1376-8.

[19] Wu, J.—Zhong, S.—Jiang, J.—Yang, Y.: A Novel Clustering Method for Static Video Summarization. Multimedia Tools and Applications, Vol. 76, 2017, No. 7, pp. 9625–9641, doi: 10.1007/s11042-016-3569-x.

[20] Wolf, W.: Key Frame Selection by Motion Analysis. 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, Vol. 2, 1996, pp. 1228–1231, doi: 10.1109/ICASSP.1996.543588.

[21] Yang, Z.—Pun-Cheng, L. S. C.: Vehicle Detection in Intelligent Transportation Systems and Its Applications Under Varying Environments: A Review. Image and Vision Computing, Vol. 69, 2018, pp. 143–154, doi: 10.1016/j.imavis.2017.09.008.

[22] He, K.—Gkioxari, G.—Dollár, P.—Girshick, R.: Mask R-CNN. Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.

[23] Girshick, R.: Fast R-CNN. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[24] Redmon, J.—Farhadi, A.: YOLOv3: An Incremental Improvement. CoRR, 2018, doi: 10.48550/arxiv.1804.02767.

[25] Bochkovskiy, A.—Wang, C. Y.—Liao, H. Y. M.: YOLOv4: Optimal Speed and Accuracy of Object Detection. CoRR, 2020, doi: 10.48550/arXiv.2004.10934.

[26] Liu, W.—Anguelov, D.—Erhan, D.—Szegedy, C.—Reed, S.—Fu, C. Y.—Berg, A. C.: SSD: Single Shot Multibox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.): Computer Vision – ECCV 2016. Springer, Cham, Lecture Notes in Computer Science, Vol. 9905, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[27] Li, J.—Wang, H.—Xu, Y.—Liu, F.: Road Object Detection of YOLO Algorithm with Attention Mechanism. Frontiers in Signal Processing, Vol. 5, 2021, No. 1, pp. 9–16, doi: 10.22606/fsp.2021.51002.

[28] Bao, W.—Du, X.—Wang, N.—Yuan, M.—Yang, X.: A Defect Detection Method Based on BC-YOLO for Transmission Line Components in UAV Remote Sensing Images. Remote Sensing, Vol. 14, 2022, No. 20, Art. No. 5176, doi: 10.3390/rs14205176.

[29] Wan, S.—Ding, S.—Chen, C.: Edge Computing Enabled Video Segmentation for Real-Time Traffic Monitoring in Internet of Vehicles. Pattern Recognition, Vol. 121, 2022, Art. No. 108146, doi: 10.1016/j.patcog.2021.108146.

[30] Liang, S.—Wu, H.—Zhen, L.—Hua, Q.—Garg, S.—Kaddoum, G.—Hassan, M. M.—Yu, K.: Edge YOLO: Real-Time Intelligent Object Detection System Based on Edge-Cloud Cooperation in Autonomous Vehicles. IEEE Transactions on Intelligent Transportation Systems, Vol. 23, 2022, No. 12, pp. 25345–25360, doi: 10.1109/TITS.2022.3158253.

[31] Ponni alias Sathya, S.—Ramakrishnan, S.: Non-Redundant Frame Identification and Keyframe Selection in DWT-PCA Domain for Authentication of Video. IET Image Processing, Vol. 14, 2020, No. 2, pp. 366–375, doi: 10.1049/iet-ipr.2019.0341.

[32] Wang, Z.—Bovik, A. C.—Sheikh, H. R.—Simoncelli, E. P.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing, Vol. 13, 2004, No. 4, pp. 600–612, doi: 10.1109/TIP.2003.819861.

[33] Gygli, M.—Grabner, H.—Riemenschneider, H.—Van Gool, L.: Creating Summaries from User Videos. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.): Computer Vision – ECCV 2014. Springer, Cham, Lecture Notes in Computer Science, Vol. 8695, 2014, pp. 505–520, doi: 10.1007/978-3-319-10584-0_33.

[34] Enze, Y.—Miura, Y.: Inter-Frame Differencing in Training Data for Artificial Intelligence: Contour Processing for Inter-Frame Differencing Method. 2020 IEEE International Conference on Consumer Electronics – Taiwan (ICCE-Taiwan), 2020, pp. 1–2, doi: 10.1109/ICCE-Taiwan49838.2020.9258108.

[35] Cheng, Y. H.—Wang, J.: A Motion Image Detection Method Based on the Inter-Frame Difference Method. Applied Mechanics and Materials, Vol. 490, 2014, pp. 1283–1286, doi: 10.4028/www.scientific.net/AMM.490-491.1283.

[36] Podder, P.—Khan, T. Z.—Khan, M. H.—Rahman, M. M.: Comparative Performance Analysis of Hamming, Hanning and Blackman Window. International Journal of Computer Applications, Vol. 96, 2014, No. 18, pp. 1–7, doi: 10.5120/16891-6927.

[37] Zhou, F.—Zhao, H.—Nie, Z.: Safety Helmet Detection Based on YOLOv5. 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), 2021, pp. 6–11, doi: 10.1109/ICPECA51329.2021.9362711.

[38] Yan, B.—Fan, P.—Lei, X.—Liu, Z.—Yang, F.: A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5. Remote Sensing, Vol. 13, 2021, No. 9, Art. No. 1619, doi: 10.3390/rs13091619.

[39] Zhang, C.—Zhu, L.—Yu, L.: Review of Attention Mechanism in Convolutional Neural Networks. Computer Engineering and Applications, Vol. 57, 2021, No. 20, pp. 64–72 (in Chinese).

[40] Wang, Q.—Wu, B.—Zhu, P.—Li, P.—Zuo, W.—Hu, Q.: Supplementary Material for ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11531–11539, doi: 10.1109/CVPR42600.2020.01155.

[41] Lin, T. Y.—Goyal, P.—Girshick, R.—He, K.—Dollár, P.: Focal Loss for Dense Object Detection. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

[42] Tang, Z.—Naphade, M.—Liu, M. Y.—Yang, X.—Birchfield, S.—Wang, S.—Kumar, R.—Anastasiu, D.—Hwang, J. N.: CityFlow: A City-Scale Benchmark for Multi-Target Multi-Camera Vehicle Tracking and Re-Identification. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 8789–8798, doi: 10.1109/CVPR.2019.00900.

[43] Wang, J.—Zeng, C.—Wang, Z.—Jiang, K.: An Improved Smart Key Frame Extraction Algorithm for Vehicle Target Recognition. Computers & Electrical Engineering, Vol. 97, 2022, Art. No. 107540, doi: 10.1016/j.compeleceng.2021.107540.

[44] Wen, L.—Du, D.—Cai, Z.—Lei, Z.—Chang, M. C.—Qi, H.—Lim, J.—Yang, M. H.—Lyu, S.: UA-DETRAC: A New Benchmark and Protocol for Multi-

Object Detection and Tracking. Computer Vision and Image Understanding, Vol. 193, 2020, Art. No. 102907, doi: 10.1016/j.cviu.2020.102907.

[45] Gong, H.—Mu, T.—Li, Q.—Dai, H.—Li, C.—He, Z.—Wang, W.—Han, F.—Tuniyazi, A.—Li, H.—Lang, X.—Li, Z.—Wang, B.: Swin-Transformer-Enabled YOLOv5 with Attention Mechanism for Small Object Detection on Satellite Images. Remote Sensing, Vol. 14, 2022, No. 12, Art. No. 2861, doi: 10.3390/rs14122861.

Hao Pan is a Master student in the School of Information and Electronic Engineering at Shandong Technology and Business University, Yantai, China. His current research interests include big data, deep learning.

Shaopeng Guan is currently serving as Associate Professor with the School of Information and Electronic Engineering, Shandong Technology and Business University, Yantai, China. His current research interests include big data, networks and communications, etc.

Xiaoyan Zhao is a Master student in the School of Information and Electronic Engineering at Shandong Technology and Business University, Yantai, China. Her current research interests include big data, deep learning.

Yuewei Xue is a Master student in the School of Information and Electronic Engineering at Shandong Technology and Business University, Yantai, China. Her current research interests include big data, deep learning.