Pattern Recognition 121 (2022) 108146

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Edge computing enabled video segmentation for real-time traffic monitoring in internet of vehicles

Shaohua Wan[a,b,*], Songtao Ding[c], Chen Chen[d]

[a] School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China
[b] State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[c] School of Computer Science & Technology, Xi'an University of Posts & Telecommunications, Xi'an 710121, China
[d] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China

ARTICLE INFO

Article history:
Received 17 December 2020
Revised 21 May 2021
Accepted 27 June 2021
Available online 12 July 2021

Keywords:
Video segmentation
Key frames extraction
Edge computing
YOLOv3

ABSTRACT

In the Internet of Things enabled intelligent transportation systems, a huge amount of vehicle video data has been generated and real-time and accurate video analysis are very important and challenging work, especially in situations with complex street scenes. Therefore, we propose edge computing based video pre-processing to eliminate the redundant frames, so that we migrate the partial or all the video processing task to the edge, thereby diminishing the computing, storage and network bandwidth requirements of the cloud center, and enhancing the effectiveness of video analyzes. To eliminate the redundancy of the traffic video, the magnitude of motion detection based on spatio-temporal interest points (STIP) and the multi-modal linear features combination are presented which splits a video into super frame segments of interests. After that, we select the key frames from these interesting segments of the long videos with the design and detection of the prominent region. Finally, the extensive numerical experimental verification results show our methods are superior to the previous algorithms for different stages of the redundancy elimination, video segmentation, key frame selection and vehicle detection.

1. Introduction

As a basic part of smart cities and Internet of Vehicles (IoV), intelligent sensing devices exploit smart sensors/cameras to collect and analyse data. The widespread deployment of these devices has resulted in an exponential increase in network traffic. They create a large amount of video and image data that requires to be transmitted to the cloud center, which in turn increases the bandwidth of the network. Particularly, a monitoring camera can generate almost 25–30 frames per second in video transmission; a low frame rate (1.25Hz) moving image sequence generates more than 100M data per second [1]. The considerable computation and storage resources need to handle the massive snapshot data produced by the video sensors in the smart surveillance applications. The departments face the severe challenge of providing the real time video analysis results to serve urban public safety and security. Obviously, cloud computing is incapable of to meet the rapid response of the video processing. The main reason is that tasks such as monitoring and tracking require fast and real-time responses while centralized cloud computing processes raw data from widely distributed video sensors, leading to uncertainty in the data transmission delay and bringing huge pressure and delay to the communication network bandwidth [2]. Moreover, this may provide more attack opportunities for the adversary and cause data security and privacy issues. Therefore, current video surveillance applications are suitable for offline forensic analysis, rather than blocking suspicious activities before causing damage. Due to the massive adoption of cameras in video surveillance systems, the demand for video preprocessing continues to grow. In the past few decades, people have made a lot of exploration. For example, researchers have investigated automatic anomaly detection algorithms using machine learning and statistical analysis methods [3,4]. However, these smart methods require a lot of expensive computing resources, though they are powerful. Besides, researchers are also trying to use event-driven visualization mechanisms to help operators extract video, reconfigure network cameras, and map regular real-time images to 3D cameras [5,6]. Nevertheless, the traditional cloud center detection solution still faces the huge challenge of the delay sensitive of the monitoring system due to the lack of the effective performance upgradation. Edge computing enabled video processing is the only practicable way to meet the real-time requirements of a large volume of video streaming [7,8].

* Corresponding author.
E-mail addresses: shaohua.wan@ieee.org (S. Wan), dst@xupt.edu.cn (S. Ding), cc2000@mail.xidian.edu.cn (C. Chen).

https://doi.org/10.1016/j.patcog.2021.108146
0031-3203/

Edge computing is an emerging computing architecture which migrates computation power from a centralized cloud center to the edges closer the user end and mobile devices, and performing part or all of the calculations at the edge can reduce latency, provide a real-time response, lower network bandwidth load, weaken the risk of privacy leakage, improve data security, and alleviate cloud center performance bottlenecks. In the cloud computing architecture, the collected video source data will be uploaded to the cloud for the further analysis; then, the results will be sent back to the client. However, this structure results in the high latency and the increasing network bandwidth consumption because all video data must be shipped to the cloud. As a typical application of edge computing, video surveillance will migrate more computing tasks to mobile intelligent sensing devices at the edge nodes. Compared to cloud computing, edge computing has the superiority in the following respects:

1. Real-time response: the tasks can be processed directly at the edge which is closer to users, and communication delay is diminished, which is crucial for real-time tasks (such as intelligent video surveillance);
2. Reduced network bandwidth: the original video data created by the smart cameras is pre-processed at the edge or the intelligent end devices, instead of being transmitted to the remote cloud center, which decreases the communication overheads than that of the cloud computing model;
3. Security and privacy: the less data transmitted, the fewer opportunities of the data being attacked.

With the rise and popularity of deep learning (DL), more and more research in the field of computer vision has begun to follow this trend. Video analysis using DL is increasingly being adopted on the Internet of Vehicles monitoring, where the analysts usually make use of cloud center resources for the video training and inference. However, due to the sharp increase in video data from snapshots, the high round-trip latency between edge devices and remote cloud centers in traditional cloud computing models cannot meet the quality of experience requirements related to network bandwidth and latency constraints. Moreover, this has also exacerbated the increase in the cost of centralized cloud computing resource management. Thus, edge intelligence has emerged [9]. On the Internet of Vehicles, the video surveillance system is the main source of information, and surveillance cameras deployed at various locations on the road can obtain traffic videos in real-time. Among them, target detection in traffic videos is the foundation for a series of follow-up operations, such as traffic statistics, intersection monitoring, vehicle and pedestrian location and identification, driving status, road conditions, accident determination, and tracking of specific vehicles and pedestrians [10,11]. In this context, there is an urgent need and a wide range of application scenarios for the research on the key technologies of video surveillance on IoV.

In the complex and changeable IoV traffic environment, the characteristics of weak equipment processing capabilities and limited storage greatly restrict the real-time processing of large-scale information collected by vehicles, severely affecting the safety and reliability of traffic. The continuous emergence of new intelligent applications such as augmented reality and traffic situation awareness poses more severe challenges to the computing power and rapid inference and response in the intelligent road network traffic environment. Video analysis is a complex problem, which can usually be logically composed of multiple steps, including motion detection, key frame extraction, target object detection and identification. By using the edge computing model to split the DL model at the edge of the initial inspection and the cloud fine-grained inspection, the specific scenarios of the Internet of Vehicles can be initially analyzed at the edge, the key videos that match the current mission objectives can be filtered out, reducing the computing load of cloud servers, decreasing the pressure on network bandwidth and storage, and improving the processing of video analysis effectiveness. Edge computing is not intended to substitute the cloud, but to complement it, providing a better computing platform for mobile computing, IoT, etc. In summary, the main contributions of this research are as highlighted below:

- By analysing the motion magnitude of the videos, the proposed spatio-temporal interest points algorithm could remove a lot of redundant vehicle video at the edge, thereby decreasing the total of video frames which are required to be processed subsequently.
- The multi-modal linear features combination for the video segmentation method divides the video sequence into segments of interests, and then extracts the video clips from these segments.
- An optimized YOLOv3 vehicle detection algorithm is presented based on edge computing, which greatly improves the detection speed and further meets the low latency and high accuracy requirements.
- The extensive numerical experimental verification results show our methods may achieve better performance, compared with the existing algorithms for different stages of the redundancy elimination, video segmentation, key frame selection and vehicle detection.

The organization of this paper is structured as follows. Video analysis based on edge computing and the state of the art works are presented in **Related work**. Furthermore, our proposed algorithms of the redundancy elimination, super frame segmentation and key frames extraction are elaborated in **Methods**. Moreover, we perform the experiments to verify the effectiveness of our methods in **Experiments**. Finally, we summarize this research in **Conclusions**.

2. Related work

Video analysis based on edge computing for real-time vehicle monitoring in smart city has been currently attracting a significant amount of attention among practitioners as well as researchers. A large amount of traffic video resources are submitted to Internet. How to delete the redundant video frames, split video sequences correctly and to decrease the unnecessary computation resources consumption is a challenging work in accordance with the original video content [12]. presents a method to detect and localize video anomaly with motion-field shape [13,14]. makes use of an improved Harris-Laplace spatio-temporal interest points to recognize the interesting segments from the big videos, from which the key frames are selected afterwards. The spatio-temporal interest points can be accurately attached around the detected target, and the region of interest construction algorithm can quickly and accurately locate the candidate region containing the target. DL has received widespread attention due to the great success of image classification and target identification, specifically video surveillance, object counting, and vehicle detection [15]. In contrast [16], proposes the calculation of the spatial and temporal feature maps and then detects video saliency. Due to resource constraints (energy consumption, computation and memory), it is still a challenge to deploy these large, powerful video tasks with low latency requirements on the smart end devices. Therefore, it is automatical to consider transferring these computing tasks to the more mighty edge servers or clouds. However, the cloud computing model is not applicable to the edge services which need to execute shortly [17], because offloading tasks to the cloud center increases the network round-trip transmission delay, and the application service requests

do not answer shortly, but the full utilization of powerful computation and memory resources in the cloud will reduce the total response time. Because the edge nodes and edge servers are near to the user and can quickly reply to the user requests, it comes to be the preferred helper [18]. When running computationally intensive tasks on edge servers, multiple terminal device resources need to be effectively managed. The goal is to balance the performance parameters of accuracy, energy consumption, latency, and load balancing. VideoStorm [19] introduced these trade-offs to select the correct configuration for each request under the premise of meeting accuracy and latency goals. For example, in Chameleon [20], the configuration is updated online during streaming video input. However, different model segmentation points based on task division will lead to different calculation delays. Therefore, it is necessary to choose a reasonable cutting strategy to maximize the advantages of end-edge and even cloud collaboration.

When offloading task to the edge server, we could perform the data preprocessing at the edge, reducing redundancy, bandwidth, latency, and the dependence on the cloud center, while improving the efficiency of video analysis. To decrease bandwidth consumption, some academics have proposed the end-edge-cloud collaboration architecture and model compression to eliminate data transmission in different environments. For example [21], proposes that only the data inferred from the edge device is transferred to the cloud for retraining to reduce data transmission[22,23]. proposed to remove redundant data without affecting accuracy to reduce data transmission. Glimpse [24] migrates all DL computing tasks to the close edge servers while using update detection to remove the camera frames that should be offloaded. If there is no change in detection, the frame tracking will be performed locally. This filtering enhances the processing capacity of the system and enables real-time target detection on mobile devices. Vigil [25] proposes a distributed architecture which smartly leverages the processing task between the edge and the cloud to reduce bandwidth consumption in video surveillance. VideoEdge [26] similarly proposes a hierarchical architecture of the edge and cloud to process the video camera streams in order to attain the better trade-off between the multiple objectives and constraints, fairly distributing the resources. We present an edge computing enabled traffic video analysis algorithm in the intelligent transportation systems which performs real-time vehicle identification, where we improve the exsiting YOLOv3 to detect the traffic flow [27,28].

3. Methods

3.1. Redundant video frame processing at the edge

When smart camera collects snapshot data in traffic monitoring systems, the sampling rate is generally 25 frames per second (FPS), which makes sure we smoothly watch the pictures on the screen. Since traffic monitoring requires 24-hour continuous video data collection of traffic scenes, they produce terabyte video data at a time. How to handle the extremely large amount of video streaming, mine and extract value costs much run time, not to mention low latency. It can be revealed through observation of behavioral events in common video surveillance environments that long videos often contain considerable useless static video frames, and these redundant frames take up a lot of calculation time.

To improve the efficiency of video analysis, it is fundamental to identify and eliminate a large amount of redundancy in a video streams. In the work, an effective algorithm is presented to identify the redundant frames of the long video by evaluating the motion magnitude. The algorithm first uses the improved spatio-temporal interest points (STIP) detection method to calculate the spatio-temporal interesting points in the video frames; And then, it combines background suppression with spatio-temporal constraints to decrease the interference from the useless interesting points. In the light of the characteristics of STIP, when the number and positions of STIP in each frame remain unchanged, the experimental observations demonstrate that this frame is superfluous and can be deleted. Therefore, a lot of unnecessary video frames that have not changed in a long video can be removed based on this viewpoint. The focus of traffic target detection is on pedestrian and vehicle targets. Spatiotemporal interesting point suppression is required to further delete the points of interest that are not related to the target. This paper uses Lindeberg's scale selection algorithm, where the scale $S = \frac{\sigma}{4}, \frac{\sigma}{2}, \sigma, 2\sigma, 4\sigma$. This method can optimize the selection of multi-scale interesting points. Specifically, static points of interest can be used as background points and deleted appropriately when moving targets are detected. The static interesting point suppression calculation formula is:

$$P^t_{\sigma,\beta} = Q^t_{\sigma,\beta} - \{Q^t_{\sigma,\beta} \cap Q^{t-1}_{\sigma,\beta}\} \tag{1}$$

Where $P^t_{\sigma,\beta}$ represents the set of non-static interest nodes at time t, $Q^t_{\sigma,\beta}$ represents the set of interest nodes at time t and β is used to remove the static points from time t to $t-1$ frame. A set of interesting points in the significant area of the moving target with scale information is finally obtained through the elimination of redundant points, suppression of background points, and space and time constraints.

3.2. Video frame segmentation

First, the frames in the video are transformed to the grayscale images. Then, low-pass filtering is used to process the converted image and re-sample. Finally, the regional contrast score C is calculated:

$$C = \sum_{\delta} \delta(m,n)^2 P_\delta(m,n) \tag{2}$$

where $\delta(m,n) = |m-n|$, represents the difference of the gray values between the adjacent pixels; $P_\delta(m,n)$ denotes the distribution probability of the difference from the adjacent gray pixels. Clarity score: Clarity is an essential index to evaluate the fitness of video frames and could better reflect those people's subjective emotion. Its calculation of the clarity score also needs to first convert the RGB value of the video frame into a gray value and then compute the difference square of the gray value of two adjacent pixels in each area:

$$D(f) = \sum_j \sum_i |f(i+2,j) - f(i,j)|^2 \tag{3}$$

where $f(i,j)$ denotes the gray value of the homologous pixel in the area, $D(f)$is the area definition calculation result with the maximum value as the definition E.

$$E = maxD(f) \tag{4}$$

Color saturation score: color is an objective expression of human beings to a stimulus and symbol of the surrounding. The spatial relationship will affect the visual saliency, and the outstanding contrast of the neighbouring regions has the high possibility to pay more visual observation to, similarly to compute the contrast score C. The colorfulness saturation score uses the following formula to calculate:

$$S = EMD(P_1, P_2, p(\alpha,\beta)), 0 \le \alpha, \beta \le 63 \tag{5}$$

Attention score: By using dynamic visual saliency based on time gradient, the detection model collects the frames which may draw visual attention and the corresponding attention score A is calculated as follows:

$$A = \theta \times M + (1-\theta) \times \nu \tag{6}$$

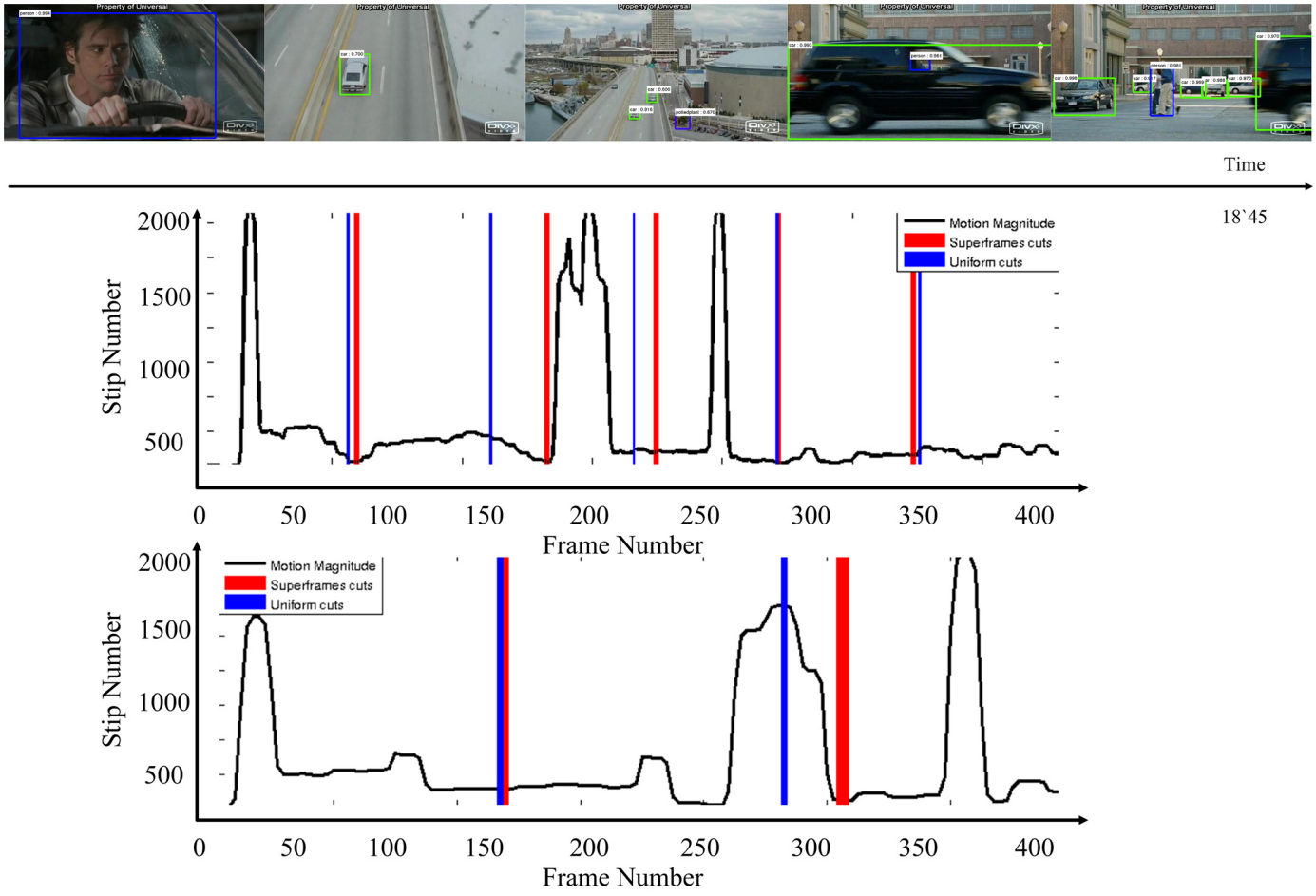


Fig. 1. Splitting a long video into segments of interests by the non-linear combination.

where M is the large frame movement magnitude, and ν denotes the variance. These two parameters are the corresponding average value and variance of the non-boundary frame movement in the super frame segmentation, respectively. Face detection score (F): add feature information based on face detection as part of the reference feature of video segmentation. The facial influence score F of the entire video frame refers to [13]. Finally, a multi-modal linear combination is employed to compute the interest score of the super frame segmentation attention score:

$$I_{score} = \varphi(A) + C \cdot E \cdot S + \psi(F) \tag{7}$$

where $\psi = 0.5, \varphi = 0.25$. An interest score is calculated with the means of the non-linear combination of above-mentioned elements, which measures the fitness of the super-frame segmentation. The traffic video is segmented into some clips, which include the important frames from the original video and are utilized to analyze the vehicle video frames at the edge. As seen in Fig. 1, the video frames is divided into multiple segment of interests in the light of the linear combination scores which are measured by the impactful features elements.

3.3. Key frame selection algorithm

The D-K-Means clustering model dynamically group the temporal and spatial interesting points near the target to obtain the positions of all cluster centers. In the process of clustering, some pseudo-clustering center points will be generated due to the environment. These false central points are produced by a few interference points. The generation of false central points can be effectively reduced by setting the threshold of the number of interesting points participating in the clustering. The processing method in this paper is to eliminate the point which has the largest distance of the interesting node and the cluster center in each calculation regardless of the distance. We also use the method of comparing the nearest neighbor distance in the experiment. Particularly, the interesting point can also be deleted if the distance ratio of the two interesting nodes belonging to the identified cluster center point is larger than a specific threshold. After determining the cluster centers of multiple targets in the image, the Euclidean distance between the points of interest to which the targets belong to the cluster centers is computed; after completing the many targets center points spacing determination and the closest adjacent identification of interesting points, the detection candidate area is constructed. In the vertical way of the object, the two nodes with the biggest distance of the target center are selected to obtain h1 and h2, and h1+h2 is regarded as the height h of the region of interest. In the same way, the width w of interest is obtained in the horizontal direction of the object. Then, the previously determined object center point is regarded as that of the rectangle; h and w denote the height and width of the region, respectively, to construct the candidate area. Experiments demonstrate that the local area constructed by this method can quickly and effectively enclose the target in it.

In this research, a saliency region detection method based on color contrast is proposed, which selects key frames by comparing the saliency of the region of interest in the image. First, the color contrast detection and spatial relationship calculation are combined for regional saliency detection. Then, the image segmenta-

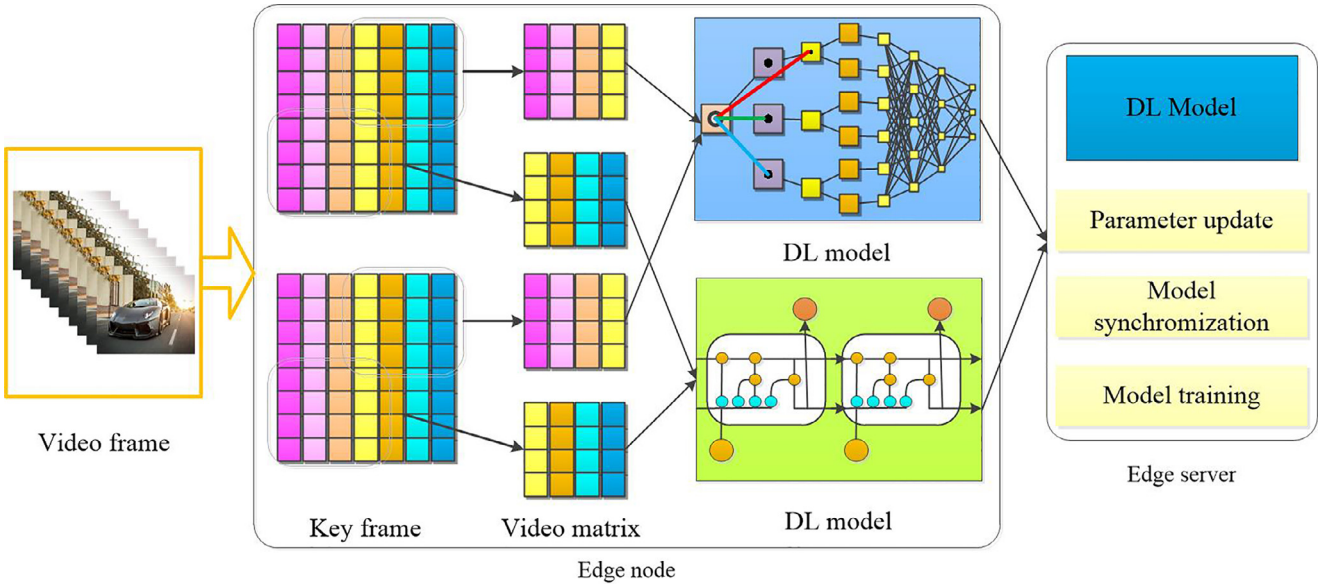


Fig. 2. Deep learning for video analysis at the edge.

tion algorithm is used to divide the region, and a color histogram is built for each segmented region. For the divided area O_k, the color contrast between it and other regions is calculated. Then, the weighted sum of each area and other areas is adopted to calculate the color contrast score (C):

$$C(O_k) = \sum w(O_i) D_o(O_k, O_i) \tag{8}$$

where $w(O_i)$ denotes the area weight parameter, $D_o(O_k, O_i)$ represents the color distance between the areas of (O_k, O_i), and it is expressed as

$$D(O_1, O_2) = \sum_{u=1}^{n_1} \sum_{v=1}^{n_2} P(x_1, u) P(x_2, v) D(x_{1,u}, x_{2,v}) \tag{9}$$

where $p(x_l, u)$ is the probability that the u-th color appears in the l-th area. The most significant frame (F^*) in the video clip can be obtained by comparing the color contrast scores C of video frames.

$$F^* = argmax \sum C(O_k) \tag{10}$$

Finally, for each video segment, three video frames are extracted as the key frames of this video. The three video frames are the first frame, the last frame, and the video frame (F^*) of the video clip.

3.4. Edge computing-based YOLOv3 for vehicle detection

In this section, an edge vehicle detection method based on the optimized YOLOv3 [29] is proposed, which is primarily motivated by real-time video surveillance. Our proposed scheme consists of three steps. Firstly, we present a distributed deep learning architecture at the edge, which allocates the distributed training and inference tasks among the edge nodes, greatly decreases the transmission cost and lowers the response time, as depicts in Fig. 2. Secondly, a dense block connection structure is brought into the backbone network, where the sparse training and channel pruning are performed in the model. Meanwhile, the loss function is optimized when training the deep learning model. Thirdly, the multiple-objects tracking and detection assignments are migrated and deployed to the edge nodes.

For the purpose of the more accuracy of the positioning and identification of object detection, YOLOv3 designs a denser convolutional neural network as its backbone, which has 53 convolutional layers named Darknet-53. It heavily makes use of a great number of 1×1 and 3×3 convolution kernels for the elimination of the number of parameters. To enhance the detection results of the small targets, we refer to feature pyramid networks [30] to put forward a multi-scale feature extraction framework. YOLOv3 adopts 3 different scale feature maps to infer the detection effects, and finally to get a 3-d tensor, including bounding box, target evaluation (objectiveness score), and classification prediction on each scale feature map. YOLOv3 auxiliary predicts the coordinations using 9 clusters to obtain the anchor boxes, and 9 anchor boxes of different scales are distributed into 3 groups and exerted on 3 different scale feature maps, that is, the cell grid of each scale feature map utilizes anchor boxes to infer 3 sets of the results. When the input image is 416×416, YOLOv3 will output a total of $(52 \times 52 + 26 \times 26 + 13 \times 13) \times 3 = 10647$ candidate frames.

The Darknet-53 network is too complicated and superfluous, which leads the increase of training complexity and speed. To real-time detect the vehicles in smart city, we need to reduce the trainable model parameters of the feature extraction network structure by exploiting the improved YOLOv3's backbone network, Darknet-53 with the introduction of the dense block connections [27,28,31]. As seen in Fig. 3, the maximum pooling layer with two strides is inserted as the connected structure in the middle of the dense blocks, where the connected structure decreases and denoises the feature map. This dense connection of Darknet-53 is called as Darknet-Dense.

When predicting the bounding box of the object, the prediction result has the same impact on the large target and the small one. However, the real situation is that the sensitivity of the large target is less than that of the small. When the width and height (w, h) of the bounding box change slightly while time elapses, the bounding box of the small will disappear. Therefore, the YOLOv3 uses the square root to calculate the width and height (w, h) of the bounding box. Although this processing method can make the their sensitivity close, but did not consider the difference in the impact of the overall error with the different size of many kinds of vehicles in the traffic videos. The YOLOv3 loss function uses a unified error for the prediction bounding box of large and small vehicles, which is not consistent with the real situation. In order to reflect the different proportions of errors generated by vehicles of different sizes in the entire network, this paper uses a ratio method to opti-

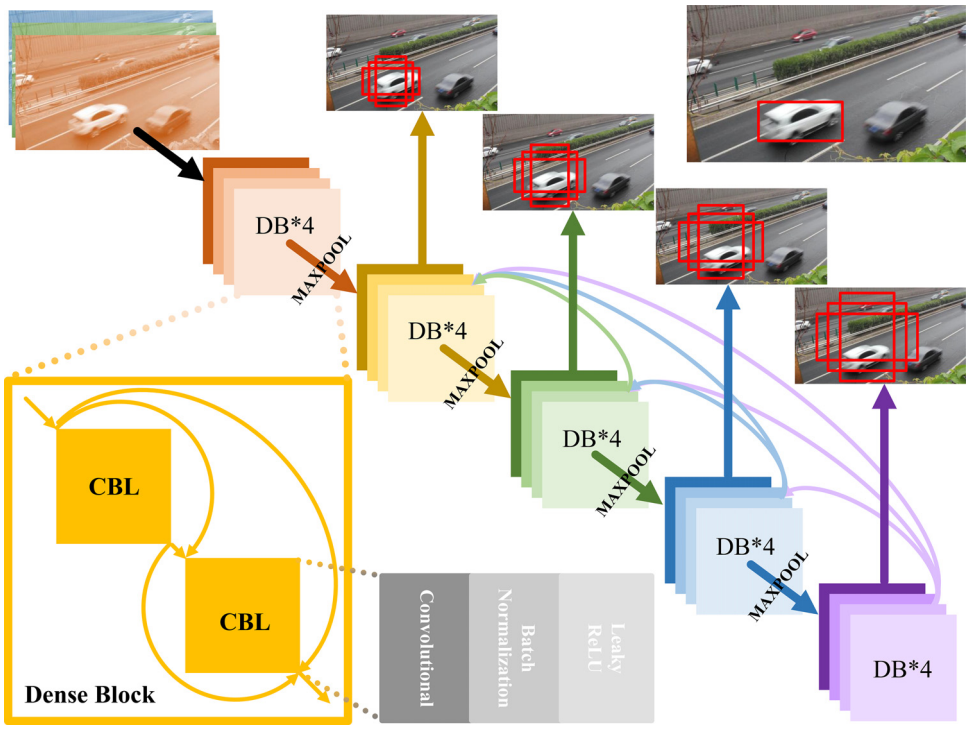


Fig. 3. Deep learning for video analysis at the edge.

mize the loss function, which can not only improve the network's ability to predict object categories, but also optimize the position of the target frame. The optimized loss function is shown in formula 11:

$$
\begin{aligned}
loss = {} & \lambda_{coord} \sum_{m=0}^{S^2} \sum_{n=0}^{B} \prod_{mn}^{obj} [(x_m - \hat{x}_m)^2 + (y_m - \hat{y}_m)^2] \\
& + \lambda_{coord} \sum_{m=0}^{S^2} \sum_{n=0}^{B} \prod_{mn}^{obj} \left[\left(\frac{w_m - \widehat{w_m}}{\widehat{w_m}} \right)^2 + \left(\frac{h_m - \hat{h_m}}{\hat{h_m}} \right)^2 \right] \\
& + \sum_{m=0}^{S^2} \sum_{n=0}^{B} \prod_{mn}^{obj} \left(C_m - \widehat{C}_m \right)^2 + \lambda_{koobj} \sum_{m=0}^{S^2} \sum_{n=0}^{B} \prod_{mn}^{koobj} \left(C_m - \widehat{C}_m \right)^2 \\
& + \sum_{m=0}^{S^2} \prod_{m}^{obj} \sum_{c \in classes} (p_m(c) - \hat{p}_m(c))^2 \qquad (11)
\end{aligned}
$$

where $\prod_{mn}^{obj}$ denotes if the target becomes up in cell m and $\prod_{mn}^{obj}$ represents which the nth bounding box indicator in cell m is "responsible" for the speculation. Each bounding box contains five predictable values: x, y, w, h and the confidence. (x, y) denotes the predicted coordinate location of the target center; (w, h) denotes the width and the height of the bounding box; and the confidence denotes the confidence for that bounding box to embrace that target; while $(\hat{x}, \hat{y}, \hat{w}, \hat{h})$ denotes the labeled center coordinates of the object. $(C_m, \widehat{C_m})$ respectively represents the calculated confidence and the set confidence. $P_m(c)$ is the predicted probability that the target is a part of the class c while $\widehat{P_m(c)}$ is the true probability. The parameter λ_{coord} is to enhance the importance of the bounding box in the loss calculation. λ_{noobj} can decrease the affect of the non-object region on the confidence computation of the object area.

4. Experiments

In this part, a numeric assessment of the proposed methods and comparison with the existing methods are presented. SumMe dataset and Hollywood2 dataset are used to assess the validity of the proposed algorithms for the super frame segmentation and the key frames extraction. The SumMe dataset is composed of 25 videos and the Hollywoo2 data set is composed of 3669 samples, which are classified into 12 movement kinds and 10 prospects, and all of them are clipped from the 69 Hollywood movies.

4.1. Evaluating the redundant video processing

To verify the effectiveness of detection pairs based on Spatio-temporal interesting points, we also conducted comparative experiments. As illustrated in Fig. 4, this experimental video extracted from the Hollywood2 dataset contains 23,276 video frames. After the motion magnitude is detected, the total of video frames is reduced to 800 frames, and the entire video is splitted into 13 segments of interest. As exhibited in Fig. 5, another experimental video which is extracted from the same dataset depicts a picture of the outdoor street scene, and it contains 7373 video frames. After the motion magnitude is detected, the number of video frames decreases to 1,650, and the whole video is allocated to 28 interesting segments while removing duplicate video frames has not change the description of the video details. Additionally, after using super frame segmentation on the remaining video frames, the quantity of interest segments is decreased from 11 to 8. As the important frame and the video segment of interest after the super frame segmentation are in one-to-one correspondence, subsequent key frame selection is reduced.

4.2. Evaluating super-frame segmentation

As depicted in the Fig. 6, the video displays driving activities in a complex outdoor situation taken from the first perspective. The algorithm in this work is used to split the video into super frames, and the video with 7000 frames is divided into 6 video segments. These 6 video clips contain the main scenes in the entire long video. The video key frames extracted from these 6 video

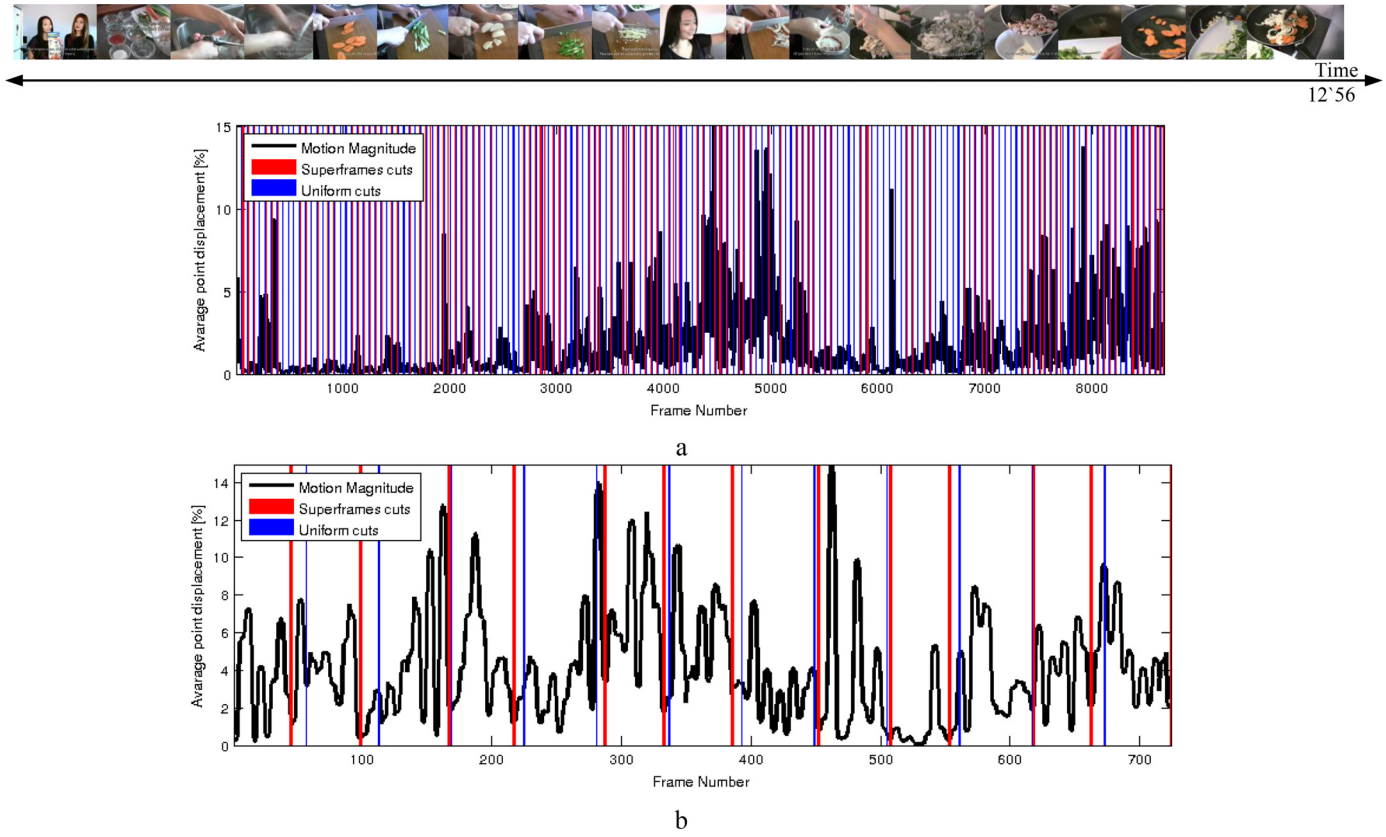


Fig. 4. Remove the redundant video frames using the STIP.

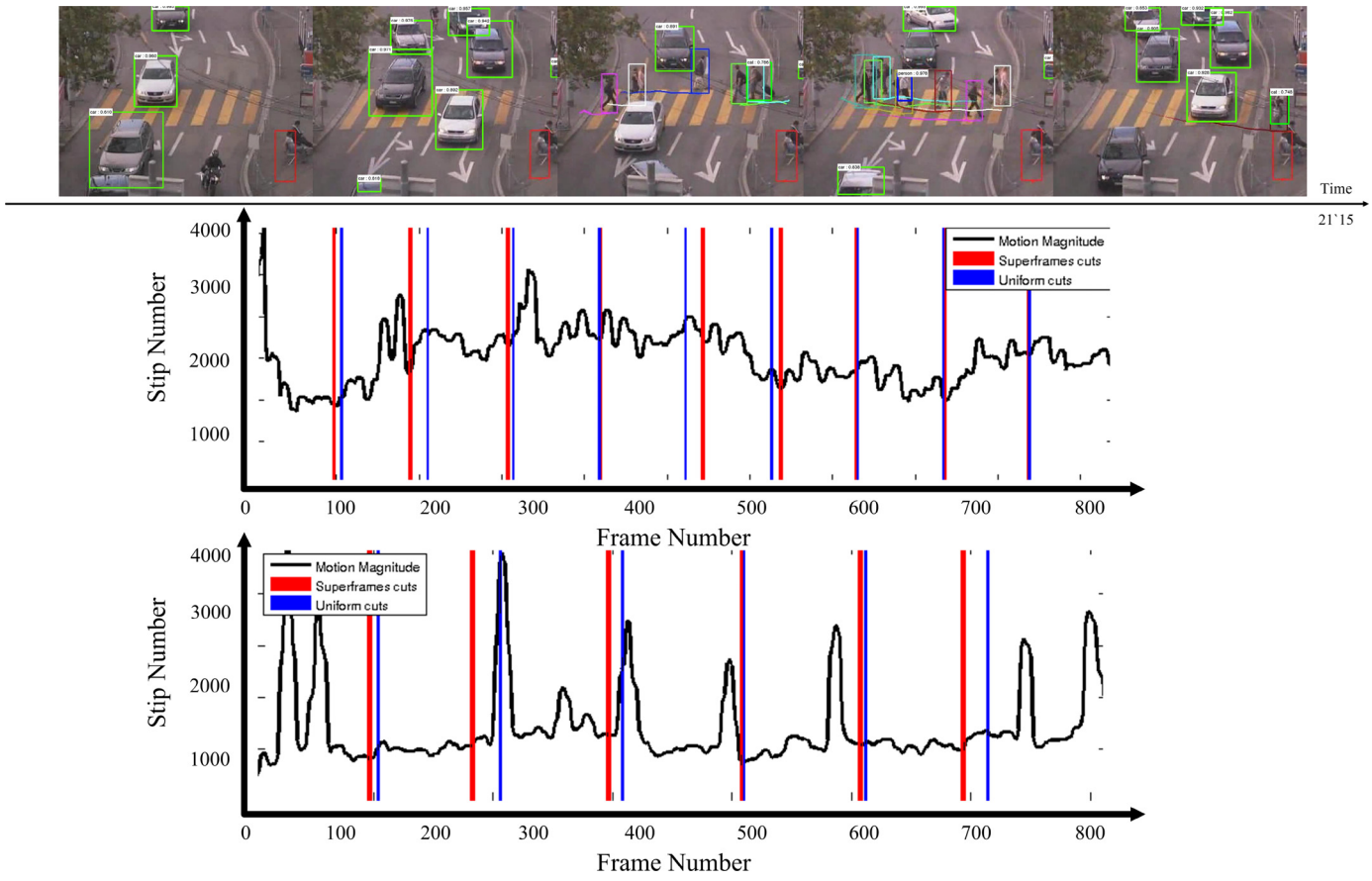


Fig. 5. Recognition the redundancy of the long video on street scene.

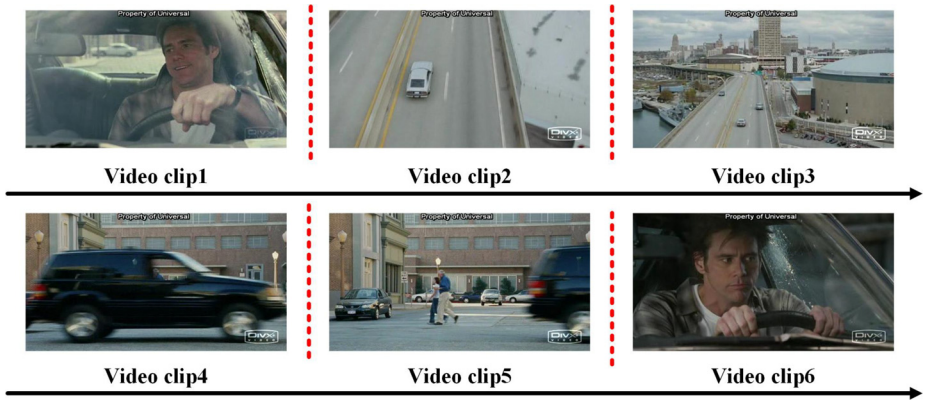


Fig. 6. Long video segmentation.

Table 1
Feature evaluation on the SumMe data set.

Feature	Mean rank	Top-1	Top-2	Top-3
Contrast	0.560	2	3	5
FaceImpact	0.302	1	2	3
Clarity	0.527	2	3	3
Saturation	0.394	1	1	4
Attention	0.380	1	2	2
Boundray	0.330	1	1	2

Table 2
Performance evaluation for key frame selection .

Algorithms	Precision(%)	Recall	F-measure
OV	0.49	0.64	0.56
DT	0.67	0.54	0.60
STIMO	0.73	0.57	0.64
VSUMMI	0.76	0.59	0.66
Our proposed	**0.78**	**0.61**	**0.68**

clips represent the main content of the long video. Table 1 demonstrates that the six different features are impactful on the video segmentation problem. In the super frame segmentation, the feature influence analysis is generally dependent on the average value of each feature. The number of top 1, top 2, and top 3 of the six features in the video illustrates different meanings. The top 1 indicates that contrast and clarity make a vital contribution to video segmentation. The top three represents which all features take a constructive part in large frame segmentation. It can be observed from the experimental results which all the features have an important effect on super frame segmentation. The whole function of each feature is very similar, though it has the lowest scores for contrast and facial impacts. The facial features are the most important consideration in video key frame detection. However, they are greatly affected by video clearness and snapshot angle in the real detection environment, making it an uncertain factor.

4.3. Evaluating the important frame selection

The selection of important video frames is a very subjective operation. Besides, different video types and lengths will also influence the choice of key frames. The evaluation of performance metrics for video key frames:

- Representativeness: the selected video key frame can represent the main content of the video clip;
- Continuity: the selected video key frame maintains the continuity of the content;
- Repeatability: the selected video key frame has less redundancy.

The video data set used in this section comes from the Key frame-Sydney (KFSYD) data set [32], containing 10 videos. To demonstrate the effectiveness of our presented algorithm, four important frame extraction methods of OV, DT, STIMO, and VSUMM [33] are compared on multiple video sequences. To assess the key frame extraction, we adopt the F-measure as a metric. The F-measure is denoted as:

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

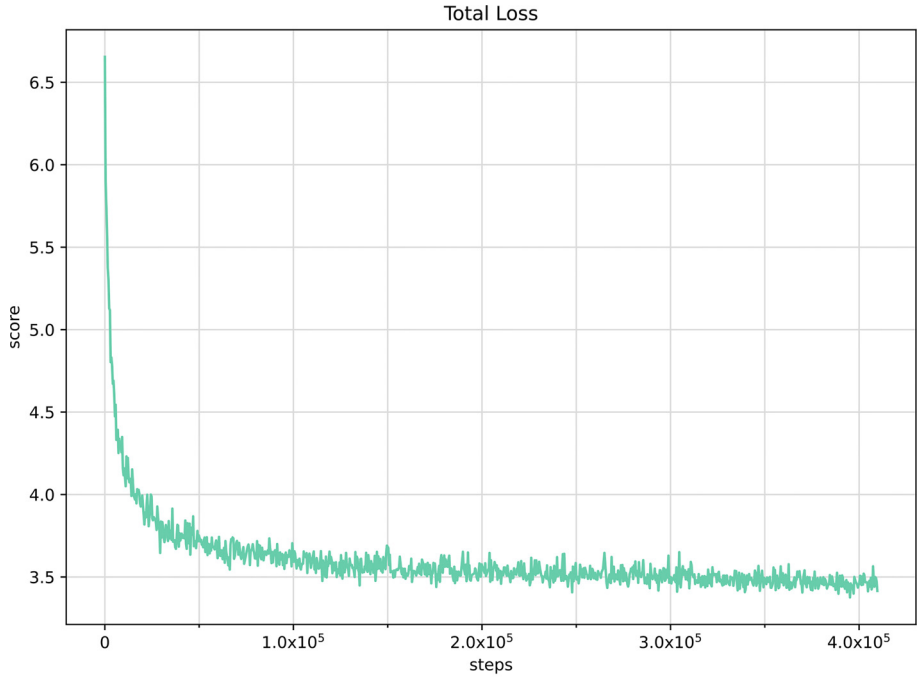
The Precision measure of the important frame extraction is denoted as the ratio of the number of the significant frames which are extracted by the proposed algorithm and match the ones selected manually to the total of important frames selected by the proposed method; and the Recall measure is denoted as the ratio of the important frames that are chosen by the proposed algorithm and match the ones selected manually to the total number of frames. It can be revealed from the Table 2 that the smaller the total number of significant frames selected by the approach, the higher the matching rate of key frames, and the lower the corresponding recall rate. If the algorithm extracts too many key frames, the recall rate will increase while the accuracy will decrease. F-value is a compromise of precision and recall, adopted to balance them.

4.4. Evaluating YOLOv3 for vehicle identification

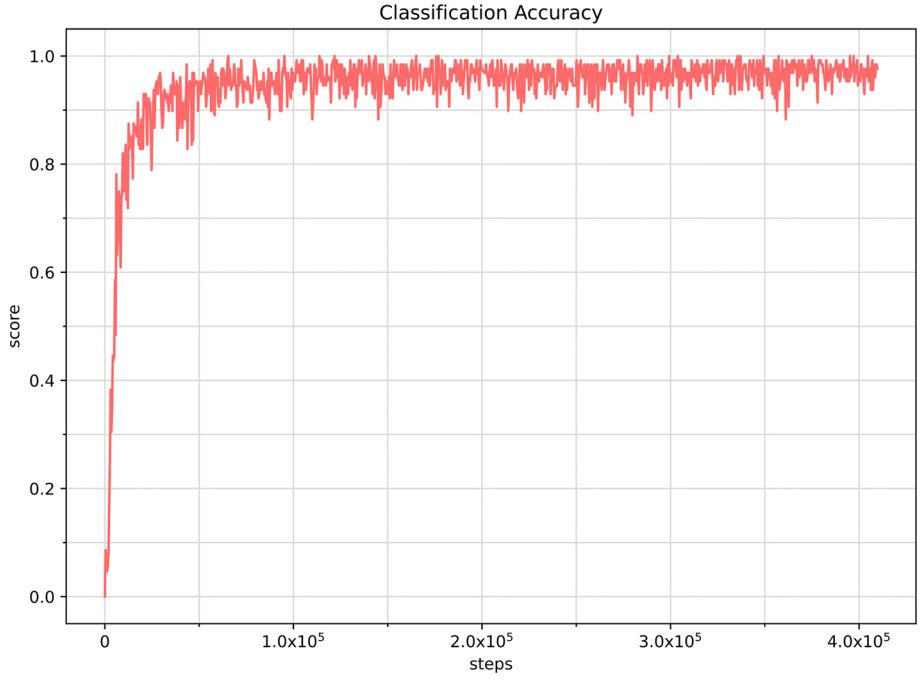
4.4.1. Experimental settings

We evaluate vehicle identification and tracking training experiments on UA-DETRAC [34] dataset, which was mainly taken in the crossroad in Beijing city. The dataset consists of about 8000 handcrafted marked vehicles and 1.20 million target video frames. Our experiments in this work are performed on the Ubuntu 18.04 system, and the YOLOv3 algorithm is run under the Darknet framework. To achieve the real time response, we offload the vehicle detection task to the edge nodes Jetson TX2 [35] with an optimized dense block structure.

The making of vehicle recognition data sets first uses Visual Object Tagging Tool (VOTT) software to tag the vehicle, and generates the corresponding xml file with the labeled information. The xml file contains the image name, path, target tag name and target location coordinates. The xml file containing the annotation information cannot be directly used to train the YOLOv3 network. The xml file needs to be converted into a txt file supported by YOLOv3. After marking all the pictures, the original pictures and



(a) YOLOv3 training loss curve



(b) Classification accuracy curve

Fig. 7. YOLOv3 train results.

Table 3
Performance comparison for the vehicle detection .

Algorithms	Precision(%)	Recall	FPS
Faster RCNN	0.89	0.84	35
YOLOv3	0.92	0.89	43
Optimized YOLOv3	**0.95**	**0.91**	**52**

all the generated files are stored according to the VOC data file structure for use in training the vehicle recognition model. In order to ensure that YOLOv3 learns the common features from the samples during training, the sample images come from the different situations, videos from different angles in the same scene, and the sample images contain hundreds of targets of the same color and shape in each classification. To enhance the generalization of the method, we randomly select 10,000 images from the whole dataset for training, 80% of which is used to produce the training set, and the remaining is assigned as the verification set and test set in a 1:1 ratio.

4.4.2. Model training

The YOLOv3 network is pre-trained first, so that a well-converged classification model can be quickly obtained, which can save a lot of time. Then we utilize the vehicle recognition data set to fine-tune the model to achieve the best recognition effect. We exploit the convergence of stochastic gradient descent (SGD) method to train the YOLOv3 50,000 times. Generally, at the beginning of the iteration, a larger learning rate is used, and always set to 0.001, so that the model parameters can be varied quickly to change to the characteristics of the data set. After a certain number of iterations, the learning rate should be decreased, otherwise it is effortless to be oscillation of the error at the extreme point, which is not able to further reduce model training errors. When the learning rate is decreased to close to 0, the model error hardly changes. When the YOLOv3 training error oscillates around a certain value, it means that the model learning has been ended. The learning rate is chosen to be 0.001 between 0 and 25,000 iterations. In 25000 35000 iterations, 0.1 times the current learning rate, and in 35,000 50,000 iterations, the learning rate is multiplied by 0.01 times. The adjustment of the learning rate reduces training loss. As shown in Fig. 7, it can be figured out that the loss value is converged after 50,000 iterations training, and the subsequent values wholly remain unchanged. Likewise, the classification precision curve shows a smooth tendency at the 50,000 iterations point after a sharp increase during the training. Therefore, we achieve a comparable classification precision value, about 0.95.

4.4.3. Experimental analysis and results

Compared to the typical vehicle recognition effect, it can be concluded that the optimized YOLOv3 has greatly improved vehicle detection performance. For the purpose of the further manifestaion of the effectiveness of the optimized YOLOv3, Precision, Recall and Frames per second (FPS) are used as performance metrics and FPS is measured the detection speed. The algorithms Faster R-CNN, YOLOv3 as well as the optimized YOLOv3 have been performed on the same data sets. As seen in the Table 3, the experimental results demonstrate that the accuracy and recall rate of our proposed algorithm are bigger than that of the others, while it has better performance in target recognition. The precision of the Faster R-CNN algorithm is equivalent to that of the YOLOv3 while the recall rate is slightly smaller than that of the YOLOv3. The main reason is that the YOLOv3 is not effective in the small objects detection, but the recognition speed of both YOLOv3 and our improvement is much more fast, which may satisfy the request of the low latency vehicle detection. The optimized YOLOv3 has improved precision and recall rate, which can improve the missed detection when the YOLOv3 identifies small targets, and to a certain extent ensure the accuracy of the traffic statistics method based on the YOLVv3.

5. Conclusions

Smart cameras are generating large volumes of videos which are required to be handled in short time, by training deep learning models in Internet of Vehicles. This research explores edge computing to migrate video analysis tasks to edge devices to achieve low latency and high quality of experience. To this end, we explore the improvement of the STIPs detection method for removing redundant video frame by combining surround suppression with local and time constraints, the multi-modal linear combination video segmentation algorithm based on high-level semantic feature modeling, and key frames extraction with the construction of the region of interests. Furthermore, we propose an optimized vehicle identification method based on YOLOv3 algorithm trained with a large amount of traffic videos. The model is pruned to guarantee its validly at the edge and edge servers. Finally, our evaluation results show that our proposed algorithms could produce more reasonable video segments and accurate key frames, indicating that they have high robustness. At the same time, the detection accuracy and speed of the optimized YOLOv3 have better performance than the other models on the UA-DETRAC dataset. To fulfill the low-latency goal of IoV video surveillance, computational complexity is also an important measure to be considered in the model except the accuracy. The excessively complicated model makes the convergence and the train speed extremely slow and the delay increase. Therefore, our future work will design a lightweight DL structure deployed at the edge, where fewer convolution kernels create a small number of feature maps.

Declaration of Competing Interest

We declare that we have no competing interests.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 62172438), the fundamental research funds for the central universities (31412111303, 31512111310) and by the open project from the State Key Laboratory for Novel Software Technology, Nanjing University, under Grant No. KFKT2019B17.

References

[1] M. Ali, A. Anjum, O. Rana, A.R. Zamani, D. Balouek-Thomert, M. Parashar, Res: real-time video stream analytics using edge enhanced clouds, IEEE Trans. Cloud Comput. (2020), doi:10.1109/TCC.2020.2991748. 1–1
[2] N. Chen, Y. Chen, S. Song, C. Huang, X. Ye, Poster abstract: smart urban surveillance using fog computing, in: 2016 IEEE/ACM Symposium on Edge Computing (SEC), 2016, pp. 95–96, doi:10.1109/SEC.2016.25.
[3] T. Fuse, K. Kamiya, Statistical anomaly detection in human dynamics monitoring using a hierarchical dirichlet process hidden markov model, IEEE Trans. Intell. Transp. Syst. 18 (11) (2017) 3083–3092, doi:10.1109/TITS.2017.2674684.
[4] J. Jia, Q. Ruan, Y. Jin, G. An, S. Ge, View-specific subspace learning and re-ranking for semi-supervised person re-identification, Pattern Recognit. 108 (2020) 107568.
[5] C. Fan, Y. Wang, C. Huang, Heterogeneous information fusion and visualization for a large-scale intelligent video surveillance system, IEEE Trans. Syst. Man Cybernet.: Systems 47 (4) (2017) 593–604, doi:10.1109/TSMC.2016.2531671.
[6] A. Li, Z. Miao, Y. Cen, X.-P. Zhang, L. Zhang, S. Chen, Abnormal event detection in surveillance videos based on low-rank and compact coefficient dictionary learning, Pattern Recognit. 108 (2020) 107355.
[7] M. Satyanarayanan, The emergence of edge computing, Computer (Long Beach Calif.) 50 (1) (2017) 30–39, doi:10.1109/MC.2017.9.
[8] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, S. Sinha, Real-time video analytics: the killer app for edge computing, Computer (Long Beach Calif.) 50 (10) (2017) 58–67, doi:10.1109/MC.2017.3641638.

[9] X. Wang, Y. Han, V.C.M. Leung, D. Niyato, X. Yan, X. Chen, Convergence of edge computing and deep learning: acomprehensive survey, IEEE Commun. Surv. Tutor. 22 (2) (2020) 869–904, doi:10.1109/COMST.2020.2970550.
[10] Y. Zhou, L. Liu, L. Shao, M. Mellor, Fast automatic vehicle annotation for urban traffic surveillance, IEEE Trans. Intell. Transp. Syst. 19 (6) (2018) 1973–1984, doi:10.1109/TITS.2017.2740303.
[11] Y. Zhang, Y. Jin, J. Chen, S. Kan, Y. Cen, Q. Cao, Pgan: part-based nondirect coupling embedded gan for person reidentification, IEEE Multimedia 27 (3) (2020a) 23–33.
[12] X. Zhang, S. Yang, J. Zhang, W. Zhang, Video anomaly detection and localization using motion-field shape description and homogeneity testing, Pattern Recognit. (2020b) 107394.
[13] S. Ding, S. Qu, Y. Xi, S. Wan, A long video caption generation algorithm for big video data retrieval, Future Generat. Comput. Syst. 93 (2019) 583–595.
[14] S. Wan, X. Xu, T. Wang, Z. Gu, An intelligent video analysis method for abnormal event detection in intelligent transportation systems, IEEE Trans. Intell. Transp. Syst. (2020) 1–9, doi:10.1109/TITS.2020.3017505.
[15] J. Chen, X. Ran, Deep learning with edge computing: a review, Proc. IEEE 107 (8) (2019) 1655–1674, doi:10.1109/JPROC.2019.2921977.
[16] Y. Fang, X. Zhang, F. Yuan, N. Imamoglu, H. Liu, Video saliency detection by gestalt theory, Pattern Recognit. 96 (2019) 106987.
[17] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646, doi:10.1109/JIOT.2016.2579198.
[18] S. Yi, Z. Hao, Q. Zhang, Q. Zhang, W. Shi, Q. Li, Lavea: latency-aware video analytics on edge computing platform, in: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017, pp. 2573–2574, doi:10.1109/ICDCS.2017.182.
[19] H. Zhang, G. Ananthanarayanan, P. Bodik, M. Philipose, P. Bahl, M.J. Freedman, Live video analytics at scale with approximation and delay-tolerance, in: NSDI'17 Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation, 2017, pp. 377–392.
[20] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Siddhartha, I. Stoica, Chameleon: scalable adaptation of video analytics, in: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, 2018, pp. 253–266.
[21] M. Song, K. Zhong, J. Zhang, Y. Hu, D. Liu, W. Zhang, J. Wang, T. Li, In-situ AI: towards autonomous and incremental deep learning for iot systems, in: 2018 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2018, pp. 92–103, doi:10.1109/HPCA.2018.00018.
[22] S. Zhang, Z. Du, L. Zhang, H. Lan, S. Liu, L. Li, Q. Guo, T. Chen, Y. Chen, Cambricon-x: an accelerator for sparse neural networks, in: 2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2016, pp. 1–12, doi:10.1109/MICRO.2016.7783723.
[23] H. Sun, W. Shi, X. Liang, Y. Yu, Vu: edge computing-enabled video usefulness detection and its application in large-scale video surveillance systems, IEEE Internet Things J. 7 (2) (2020) 800–817, doi:10.1109/JIOT.2019.2936504.
[24] T.Y.-H. Chen, H. Balakrishnan, L. Ravindranath, P. Bahl, Glimpse: continuous, real-time object recognition on mobile devices, GetMobile: Mobile Computing and Communications 20 (1) (2016) 26–29.
[25] T. Zhang, A. Chowdhery, P.V. Bahl, K. Jamieson, S. Banerjee, The design and implementation of a wireless video surveillance system, in: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, 2015, pp. 426–438.
[26] C. Hung, G. Ananthanarayanan, P. Bodik, L. Golubchik, M. Yu, P. Bahl, M. Philipose, Videoedge: processing camera streams using hierarchical clusters, in: 2018 IEEE/ACM Symposium on Edge Computing (SEC), 2018, pp. 115–131, doi:10.1109/SEC.2018.00016.
[27] C. Chen, B. Liu, S. Wan, P. Qiao, Q. Pei, An edge traffic flow detection scheme based on deep learning in an intelligent transportation system, IEEE Trans. Intell. Transp. Syst. (2020a) 1–13, doi:10.1109/TITS.2020.3025687.
[28] C. Chen, Z. Liu, S. Wan, J. Luan, Q. Pei, Traffic flow prediction based on deep learning in internet of vehicles, IEEE Trans. Intell. Transp. Syst. (2020b) 1–14, doi:10.1109/TITS.2020.3025856.
[29] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, K. Ouni, Car detection using unmanned aerial vehicles: comparison between faster r-CNN and YOLOv3, in: 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), 2019, pp. 1–6, doi:10.1109/UVS.2019.8658300.
[30] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[31] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA,2017, pp. 4700–4708.
[32] G. Guan, Z. Wang, S. Lu, J.D. Deng, D.D. Feng, Keypoint-based keyframe selection, IEEE Trans. Circuits Syst. Video Technol. 23 (4) (2013) 729–734, doi:10.1109/TCSVT.2012.2214871.
[33] S.E.F. de Avila, A.P.B. Lopes, A. da Luz, A. de Albuquerque Araújo, Vsumm: a mechanism designed to produce static video summaries and a novel evaluation method, Pattern Recognit. Lett. 32 (1) (2011) 56–68, doi:10.1016/j.patrec.2010.08.004.
[34] S. Lyu, M. Chang, D. Du, W. Li, Y. Wei, M. Del Coco, P. Carcagnì, A. Schumann, B. Munjal, D.-H. Choi, et al., Ua-detrac 2018: report of avss2018 iwt4s challenge on advanced traffic monitoring, in: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2018, pp. 1–6, doi:10.1109/AVSS.2018.8639089.
[35] T. Amert, N. Otterness, M. Yang, J.H. Anderson, F.D. Smith, Gpu scheduling on the nvidia tx2: hidden details revealed, in: 2017 IEEE Real-Time Systems Symposium (RTSS), 2017, pp. 104–115, doi:10.1109/RTSS.2017.00017.

Shaohua Wan is currently an associate professor with the School of Information and Safety Engineering, Zhongnan University of Economics and Law. His main research interests include deep learning for Internet of Things and edge computing. He is an author of over 130 peer-reviewed research papers and books, including over 30 IEEE/ACM Transactions papers such as TII, TITS, TOIT, TMM, TOMM, PR, etc. and many top conference papers in the fields of Multimedia. He is a senior member of IEEE. School of Information and Safety Engineering, Zhongnan University of Economics and Law.

Songtao Ding received his Ph.D degree in automatic control from Northwestern Polytechnical University, Xi'an, China, in 2019. He is currently a lecturer in the Department of Computer Science and Technology at Xian University of Posts and Telecommunications. His main research interests include computer vision, object detection, medical image segmentation. Xi'an University of Posts & Telecommunications.

Chen Chen is currently a Professor with the Department of telecommunication and a Member of the State Key Laboratory of Integrated Service Networks, Xidian University, where he is also the Director of Xian Key Laboratory of Mobile Edge Computing and Security and the Intelligent Transportation Research Laboratory. He serves as the general chair, a PC chair, the workshop chair, or a TPC member of a number of conferences. He has authored/coauthored two books, over 100 scientific papers in international journals and conference proceedings. Xidian University.