IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 8, 2020, accepted June 17, 2020, date of publication June 22, 2020, date of current version July 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004178

Multilevel Traffic State Detection in Traffic Surveillance System Using a Deep Residual Squeeze-and-Excitation Network and an Improved Triplet Loss

FENG TANG, XINSHA FU, MINGMAO CAI, YUE LU, YANJIE ZENG, SHIYU ZHONG, YAN HUANG, AND CHONGZHEN LU

School of Civil Engineering and Transportation, South China University of Technology, Guangzhou 510640, China

Corresponding author: Xinsha Fu (fuxinsha_scut@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51978283, and in part by the National Natural Science Foundation of China under Grant 51778242.

ABSTRACT Although a substantial number of traffic videos have been accumulated via daily monitoring, deep learning is seldom utilized to process these data for multilevel traffic state detection. The application of deep learning is limited for two reasons: (1) the multilevel traffic state based on traffic images has not been defined. (2) The high noise information in traffic images and extremely similar features of adjacent traffic states hinder accurate detection. Based on this situation, A new definition of the image-based multilevel traffic state is proposed using the ratio of the vehicle areas to the road areas in a traffic image, and a standard image dataset, including various illuminations and vast scenes, are established. A deep residual network named TrafficNet, which is embedded with Squeeze-and-Excitation blocks and is learned by the improved triplet loss, is proposed for multilevel traffic state detection. The Squeeze-and-Excitation block effectively reduces the model's attention to noise information and focuses on road areas that are associated with traffic features in an image. The improved triplet loss maps the learned features to a metric space where the distance between features of inter-class is larger than that within the same class, which improves the discrimination of features between adjacent traffic states. Relevant experiments prove that the performance of TrafficNet, whose accuracy (*Acc*) in classifying 10 traffic states reaches 94.27% with the testing dataset, is much better than that of traditional deep classification models, which do not include Squeeze-and-Excitation blocks or the improved triplet loss.

INDEX TERMS Multilevel traffic state, deep residual network, squeeze-and-excitation blocks, improved triplet loss.

I. INTRODUCTION

With population growth and urban development, the number of vehicles on roadways has increased rapidly, which has created serious traffic congestion problems [1], [2]. Traffic congestion not only occupies a substantial amount of public time and resources but also increases the risk of traffic accidents. To improve the road operation efficiency, real-time and accurate traffic state detection must be performed.

The detection equipments that are utilized for traffic parameters could be divided into fixed detectors and mobile detectors. Fixed detectors, including magnetic frequency devices and wave frequency devices and videos [3]–[6], capture the behaviors of moving vehicles via installations at fixed road locations. Mobile detectors installed in vehicles, including the Global Positioning System (GPS) and electronic tagging devices, can obtain traffic parameters by monitoring fixed road markers. Magnetic frequency devices and wave frequency devices exhibit problems of difficult installation, frequent maintenance, and limited detection ranges. Mobile equipment is extensively employed due to its advantages of portability and wide detection range, while lower detection accuracy greatly limits its use. Traffic videos have the merits of small location spacing, wide monitoring range, real-time

The associate editor coordinating the review of this manuscript and approving it for publication was Sabah Mohammed.

acquisition and strong data continuity, which provide implementation possibilities for automatic traffic state detection technology from the perspective of image semantic interpretation. But the transmission and calculation of continuous video recordings require a considerable amount of equipment and power resources. Because the process of changing traffic state is time consuming, detection of the traffic state of each frame of video is not necessary. If video detection is converted to image detection with fewer frames, a considerable amount of resources will be conserved [10], [21].

Some studies have simplified traffic state detection to image classification problems based on traffic videos [13]–[16]. Numerous methods indirectly represented the traffic state by extracting the speeds of moving objects on images. For example, work [13] extracted moving objects by a background subtraction algorithm [14] and then applied the optical flow [15] to estimate the speed of the moving object. These methods assumed that more congested scenes have more moving objects with low speeds or stopped objects. However, they relied on preprocessing algorithms such as background subtraction and tracking, which limits their detection speed and accuracy. With a substantial breakthrough of deep learning technology in the field of vision tasks, some studies [9], [10], [20], [21] used the advantage of the powerful fitting ability of a convolutional neural network (CNN) to automatically mine traffic features from images. These works automatically and efficiently extracted congestion features and did not require preprocessing of images. Relevant experiments [9], [21] also verified that the accuracy of traffic state detection using the deep learning method is substantially higher than the traditional preprocessing algorithm.

However, traffic state detection based on deep learning [20], [21] only approximately identifies congestion or non-congestion via subjectively dividing image data, whose achievement cannot solve an actual problem because changes in the traffic state are continuous processes. The traffic state should be divided into enough levels to monitor the changes of the multilevel traffic state more smoothly, which is beneficial to managers who take effective traffic diversion measures before congestion occurs. Compared with the detection of the two-traffic state, the detection of the multilevel traffic state has more difficulties: (1) large-scale marked multilevel traffic state datasets that contain different scenes, such as different light conditions and road conditions, are lacking, because they do not have a clear definition of the image-based multilevel traffic state, and (2) traditional deep classification models [7], [8] have been suggested to be applicable to most public datasets, such as ImageNet, MNIST and CIFAR-10, while the applicability of traditional deep classification models to multilevel traffic datasets has not been verified due to the two differences between traffic datasets and public datasets. First, different objects in the public dataset represent different classes, while all objects in our traffic data are roads and vehicles. This difference hinders detection accuracy because the inter-class feature gap in the traffic dataset is fuzzier than that in the public dataset. Second, objects in the public dataset are located in the center and occupy most pixels of an image, while the effective part of traffic images is relatively small. A large amount of noise information, such as sidewalks, central dividers, and surrounding buildings in traffic images, interferes with the discrimination ability of the deep models.

If a deep model equally processes all features in traffic images, the features of the noise information would interfere with the model's judgment. We have determined that visual attention mechanisms are capable of addressing this issue [11], [23]–[25]. The role of visual attention mechanisms is to actively disregard invalid information and focus on effective areas in the images. The latest research [23]–[25] also verified that combining the visual attention mechanisms and a CNN can achieve excellent results in image classification tasks. In addition, the visual attention mechanism of squeeze-and-excitation block [11], which employed a mask that identifies the key features in an image by another layer of new weight, has achieved excellent results, due to its small number of parameters and fast calculation.

Excessive similarity of adjacent traffic states result in feature aggregation in the mapping space, which hinders the accuracy of classification. We have discovered that deep metric learning is capable of addressing this issue. Deep metric learning attempts to seek an appropriate metric space to acquire features with strong expressive ability by designing a unique loss function to adjust the locations between different samples in metric space. Deep metric learning has also achieved many successful applications in the field of computer vision, such as face recognition [26], [47], face verification [27], [28] and image retrieval [29]. The most classic case of deep metric learning is work [47], in which a triplet loss function was designed to train a CNN for face re-recognition. Triplet loss considers a group with three samples, namely, anchor, positive sample (belongs to the same class as anchor) and negative sample (belongs to different classes with anchor), as an analysis unit. The purpose of the triplet loss function is to shorten the distance between the anchor and the positive sample and push the distance between the anchor and the negative sample by taking the Euclidean distance as the measurement index. When training large-scale datasets, triplet loss has the problem of high calculation consumption and slow calculation time. Considering the same attribute of the face dataset and multilevel traffic state dataset, the training idea of triplet loss can be employed for reference to improve the accuracy of traffic state detection.

In conclusion, This paper promotes 3 contributions for the multilevel traffic state detection: (1) we proposed a definition of image-based multilevel traffic state, which utilizes the areas ratio between vehicles and roads in an image to quantify the levels of the traffic state. The advantage of the new definition is that we can accurately divide the traffic state into any level by quantitative indicators instead of subjectively and approximately dividing the traffic state into two levels. (2) The standard dataset, which contains 10 traffic

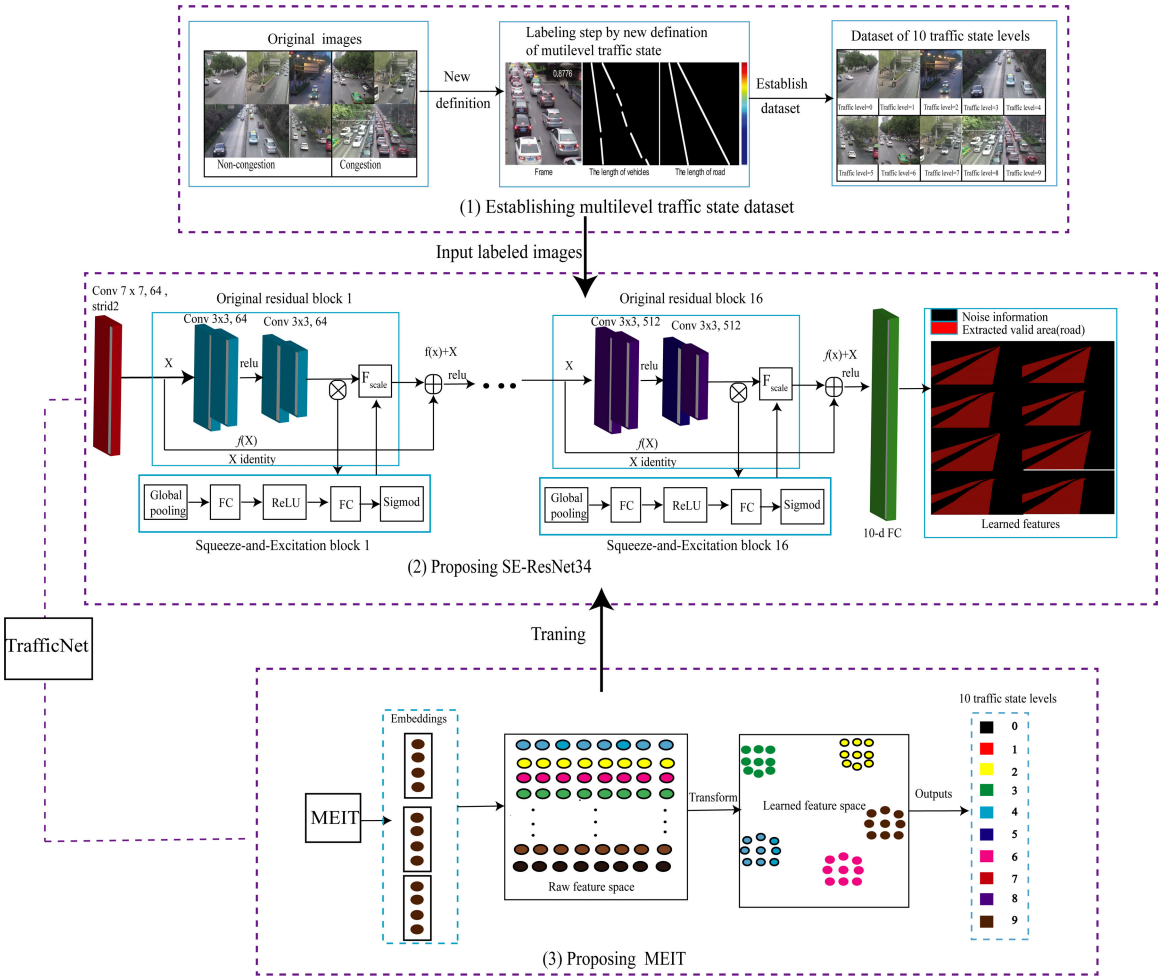


FIGURE 1. The pipeline of the proposed method. Step (1) represent establishing multilevel traffic state dataset. Step (2) describes the process of embedding the SE block into ResNet34. Step (3) illustrates the proposed MEIT, which is used to training SE-RsNet34 for multilevel traffic state classification.

states, were established to solve the problem of the lack of image-based traffic data [9], [10]. (3) We also proposed a new framework named TrafficNet that combines the visual attention mechanism of Squeeze-and-Excitation (referred to as SE) [11] and deep metric learning based on the improved triplet loss [12] (referred to as MEIT) for multilevel traffic state classification. The SE block, which attached to ResNet34, is divided into two steps: compressing a feature map into a feature description vector by global average pooling for each channel of feature map and establishing a relationship between each channel feature by two fully connected layers. The purpose of the SE block is to independently learn the weight of each channel features to improve the weight related to the traffic state and reduce the weight of noise information. The MEIT approach establishes a loss function, which contains an anchor, a farthest positive sample and a nearest negative samples in a batch-sample, to map features to a metric space. In this space, the distance between different classes is maximized and the inner classes are closely spaced. In relevant experiments, the accuracy (*Acc*) of TrafficNet to classify 10 traffic state reaches 94.27% for testing data. And the performance of our model is much better than that of traditional deep classification models which do not include SE blocks or the improved triplet loss, which suggested that our methods are suitable for multilevel traffic state detection. The pipeline of the proposed method is shown in Fig. 1.

II. RELATED WORK

A. TRAFFIC STATE DETECTION BASED ON VIDEOS OR IMAGES

Currently, research of traffic state detection can be divided into two categories, where the key points and moving areas of images are analyzed in the first category and image features are directly extracted in the second category. The first approach assumes that a larger number of moving objects represents a higher degree of traffic congestion in the scenes. Sreekumar *et al.* [13] proposed a congestion classification algorithm, which is based on the segmentation of moving vehicles. First, the background subtraction method [14] was employed to segment moving objects, and second, the optical

flow [15] was utilized to calculate the speeds of the moving points. Last, fuzzy logic was adopted to make decisions regarding the traffic state. Sobral *et al.* [6] proposed the congestion identification method, which combines key points and moving pixel features. The researchers estimated the density of traffic scene images by a background subtraction method and then calculated the speed using the Kanade-Lucas-Tomasi (KLT) algorithm [16]. These methods heavily rely on the preprocess of the background subtraction method and object tracking, which limits the detection accuracy because of the associated uncertainty. The second approach directly identifies related congestion features to achieve automatic detection. Derpanis and Wildes [17] proposed spatio-temporal analysis to distinguish features using visual variation rules in traffic scenes. Riaz and Khan [18] tagged features of moving vehicles by analyzing the statistical information of the motion vector. Dallalzadeh *et al.* [19] suggested symbolic representation to combine physiognomic information and mobile information of traffic state. Yuan *et al.* [10] extracted congestion features without supervision by the local smoothing density estimation method. These methods do not rely on image preprocessing algorithms and work well in specific traffic scenes. However, designating suitable features for different traffic scenes and the multilevel traffic state remains a challenging task.

B. VISUAL ATTENTION MECHANISMS

Visual attention mechanisms can be interpreted as methods that bias the allocation of existing computing resources toward the most abundant and effective information [22]–[25]. Existing visual attention models can be divided into soft attention models and hard attention models. Soft attention models [26], [27] predict the attention region in a deterministic way and can adopt end-to-end training by the back-propagation method. Hard attention models [29]–[31] predict the attention points of images, which are stochastic and often trained by reinforcement learning [32] or maximizing the approximate variational lower bound. In general, the soft attention model is more effective than the hard attention model because of its end-to-end training method. Visual attention mechanisms have also demonstrated their significant effects in the areas of sequence learning [33], [34], image localization and understanding [35], and image capture [36], [37]. In these applications, the visual attention mechanisms corrects one or more features exported from the convolution layer, which renders the features more sensitive to the effective information. From the perspective of attention modeling, relevant scholars have carried out studies on the comprehensive utilization of spatial and channel attention [38], [39]. Newell *et al.* [40] designed the trunk-mask attention mechanism based on hourglass modules, which are inserted in the middle layer of the deep residual network to improve the expression ability of features. Conversely, Jie *et al.* [11] designed a lightweight gating mechanism named Squeeze-and-Excitation to improve the expression ability of the whole network, which established the relationship between channels by an efficient full-connection layer. For traffic images, to disregard invalid information in images more efficiently, the selection of a suitable attention mechanism in the CNN network is important.

C. DISTANCE METRIC LEARNING

The purpose of distance metric learning is to map features to a space where the measurement distance between similar samples is small and the measurement distance between different classes is large. The mapping function can be linear transformation [41]–[44] or a deep neural network [45]–[47]. Currently, distance metric learning has become one of the most active research topics in computer vision and pattern recognition. The most influential methods are the triplet loss and an associated series of improvement methods. Weinberger *et al.* [41] considered searching for the metric space by proposing an optimization method of large-margin-nearest-neighbor loss, which could effectively decrease the distance between similar samples and increase the distance between dissimilar samples. Inspired by the method of the large-margin-nearest-neighbor loss, FaceNet [47] proposed a classic structure, which is referred to as the triplet loss and established anchor samples, positive samples (similar to the anchor samples) and negative samples (different from the anchor samples) to learn a new embedded space. In this structure, the distance between the anchor and the negative sample is larger than the distance between the anchor and the positive sample, and the minimum distance threshold is the *margin*. The advantage of traditional triplet loss is that although all features of the same class will eventually form a single cluster in space, these features will only be as close as possible without over learning and collapsing into a point. However, traditional triplet loss has a major drawback [48], since sampling all triples existing in a batch-sample, with an increasing number of datasets, the number of triplets may increase to the third power, which makes the time-consuming training process unrealistic. Therefore, numerous variant constructs based on triplet loss have emerged to avoid defects, and this paper also uses an improved triplet loss. Extensive evaluation suggests that triplet loss and its improved method has a strong adaptive ability in face recognition, clustering and image retrieval. An examination of how to combine triplet loss and classification methods to solve the problem caused by the particularity of traffic data is warranted.

III. METHOD OF ESTABLISHING MULTILEVEL TRAFFIC STATE DATA

Traffic state indexes are mainly employed to reflect the traffic characteristics, including traffic flow, speed, density, queue length, occupancy, headway spacing, etc. The main idea of traffic state detection is that, given the threshold values of different traffic states, the current traffic state can be determined by comparing the actual traffic parameters to the thresholds. However, obtaining the traffic state indexes by image data has not been studied. Considering the strong spatial expression ability of images, the traffic state index is represented from

the perspective of the space occupancy reflected by an image. Concretely, we intuitively apply the area ratio of vehicles to roads as the traffic sate index, which is calculated as

$$C_T = \frac{\sum_{(x,y)} f(x,y)}{W \times L} \tag{1}$$

where $C_T \in [0, 1]$ is the traffic state index; the higher is its value, the more congested is the traffic. L and W represent the length of roads and the width of roads, respectively. (x, y) is the coordinate of a point on the road. $f(x, y)$ is a logical function, whose value is 0 or 1, which explains whether the point (x, y) is occupied by vehicles. The definition of $f(x, y)$ is expressed as follows:

$$f(x) = \begin{cases} 0, & \text{not occpupied} \\ 1, & \text{occpupied} \end{cases} \tag{2}$$

The new definition C_T has the following advantages:

(1) The definition is accurate and quantitative. Compared with the traditional method, which subjectively and crudely classifies the traffic state into two levels, the proposed traffic index can be used to precisely classify the traffic state into any level.

(2) The definition is universal. The definition is applicable to all traffic images, which enables different cameras to cooperatively work to compare the traffic state between different scenes. This merit renders the detection results more practical.

(3) The definition takes into account the spatial information of the traffic state, which fully utilizes the outstanding properties of images.

Because the traffic index C_T is calculated at the pixel level, marking the data is time consuming. To improve the marking efficiency, we assume that the width of a vehicle and a lane are equal, which is consistent with most practical situations. Thus, this finding does not affect the accuracy of the traffic index calculation. The simplified calculation equation of the traffic state index C_T is expressed as follows:

$$C_T = \frac{\sum_{y} f(y)}{L} \tag{3}$$

In the simplified definition, the length of a lane is denoted by the length of a straight line in an image, and the total lengths of vehicles can be expressed by intermittent straight lines. Due to perspective transformation, vehicles far from the camera occupy a small pixel area in an image. Therefore, different pixel points should be given corresponding weights to eliminate the errors caused by perspective transformation. We obtained the weight by the lane width, which is also affected by perspective transformation. The width of the same lane is fixed, while the lane width far from the camera in an image is less than that close to the camera in an image. Therefore, the weight can be calculated by the ratio of the lane width in the image to the standard width in practice. An illustrated drawing of multilevel traffic state annotation is shown in Fig. 2 and typical labeled images of 2 scenes are shown in Fig. 3. We can accurately divide the traffic state into several levels according to the new labeling method.

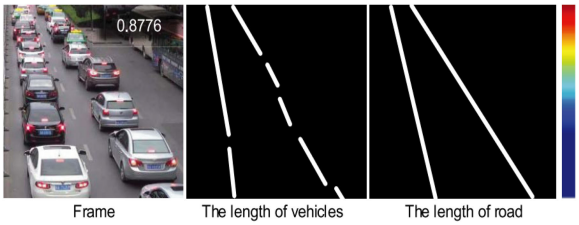


FIGURE 2. Schematic of the fine annotations on the dataset according to traffic state index C_T. The left figure is a typical traffic image, and the upper right label indicates the traffic sate index C_T. The middle image is the corresponding marked image, in which the white lines represent the lengths of vehicles. The white lines in the right image represent the lengths of roads. The color bar represents the weight of the pixels in the line, where red represents the largest weight.

FIGURE 3. The visualization of labeling method. In this figure, two different scenes is used for examples. The numbers on the binary image indicate the congestion level.

IV. METHODOLOGY

TrafficNet is composed of two parts: SE-ResNet34 and MEIT. SE-ResNet34, which is stacked with SE-original residual blocks, is used to extract the features of the effective region in the image, while the MEIT is used to map features to a metric space that is most appropriate to the multilevel traffic state. In this section, first, we introduce the SE block. Second, we describe the structure of ResNet34 and SE-ResNet34. Last, we introduce MEIT, which employs training models.

A. BUILDING SE BLOCK

As shown in Fig. 4, the SE block [11] is a kind of computing unit that can be built on the transformation F_{tr}, which maps the input $X \in R^{C' \times H' \times W'}$ to feature maps $U \in R^{C \times H \times W}$. The form of F_{tr}, which is described in section C in this paper,

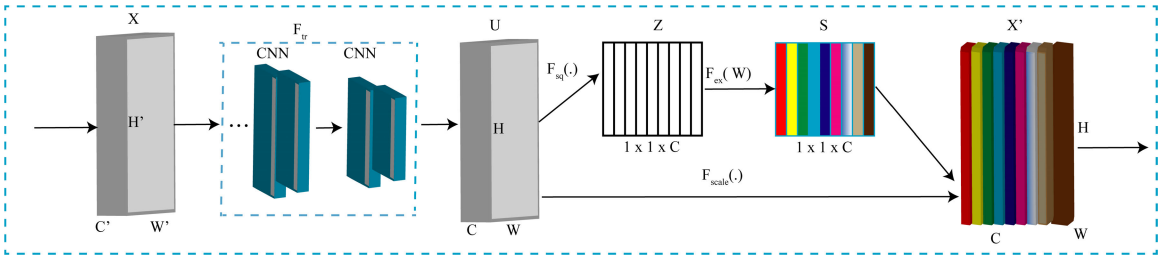


FIGURE 4. Framework of a SE block.

is related to the CNN embedded by SE blocks. As a conventional CNN, the feature maps U continue to undergo other transformations until the final feature maps are generated. As we know, each channel of a feature map reflects local features of the input image. However, the transformation F_{tr} treats all channels of X equally, which leads to the result that the noise region of U, except the road region is assigned a large weight, which is likely to affect the expression ability of multilevel traffic state features. The SE block, which is divided into two parts, namely, *squeeze* and *excitation*, reconstructs the channel's weight of U by modeling the relationships between channels. Thus, the features that are related to the road region are more significant.

In particular, *squeeze* generates channel-level information using global average pooling $F_{sq}(.)$, which is aimed at compressing global spatial information into the channel descriptor vector $Z \in R^C$. The channel descriptor vector Z is considered as collections of local features, where each element represents the global features of each channel of U. Formally, we denote $Z = [z_1, z_2, \ldots\ldots z_c]$, which is generated by compressing the feature maps $U = [u_1, u_2, \ldots\ldots u_c]$. Using the spatial dimensions $H \times W$ of U, the c-th element of Z is calculated as follows:

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j) \tag{4}$$

After compressing the information, *excitation* is implemented to fully capture the relationship among the channels of U. Each element of the descriptor vector Z represents the global feature of the corresponding channel of U. Therefore, two fully connected layers, which are regarded as the mapping function $F_{ex}(.)$, is established to parameterize the nonlinear relation of each element of Z. Parameters are then activated by the sigmoid activation function to obtain the channel weight at the pixel level of U. The *excitation* equation is expressed as

$$S = F_{ex}(Z, W) = \sigma(g(Z, V)) = \sigma(V_2 \delta(V_1 Z)) \tag{5}$$

where σ is the sigmoid function; δ is the rectified linear unit (ReLU) activation function; $V_1 \in R^{C/R \times C}$ and $V_2 \in R^{C/R \times C}$ represent the weight matrices of the full-connectivity layer; and C/R is the reducing dimension gravity of the full-connectivity layer, for which the recommended value is 16 [11]. Thus, each element of $S \in R^C$, whose values fall between 0 and 1, represents the model's attention to each channel of the feature maps U.

The final outputs of the SE block are obtained by rescaling U with the activation S as

$$x'_c = F_{scale}(u_c, s_c) = u_c s_c \tag{6}$$

where $X' = [x'_1, x'_2, .., x'_C]$ and $F_{scale}(u_c, s_c)$ refers to channel-wise multiplication between the scalar s_c and the feature map $u_c \in R^{H \times W}$. Obviously, the output X' of the SE block is the product of readjusting the channel weight on U. In the process of task learning, the weight of the channel related to the traffic state is increased, which improves the expression ability of the features.

B. BUILDING RESIDUAL LEARNING BLOCK

ResNet [11] is a deep neural network stacked with residual blocks, which realized shortcut connections by identity mapping. As shown in Fig. 5, a residual block is defined as

$$H(x) = F(x, \{W_i\}) + x \tag{7}$$

where x is the input of the residual block, and $H(x)$ is the output. $F(x, W_i)$ is the residual mapping that needs to be learned, which can be composed of any stack of convolution layers. The formulation of $F(x, W_i) + x$ can be achieved by shortcut connections, which can skip one or more layers by identity mapping.

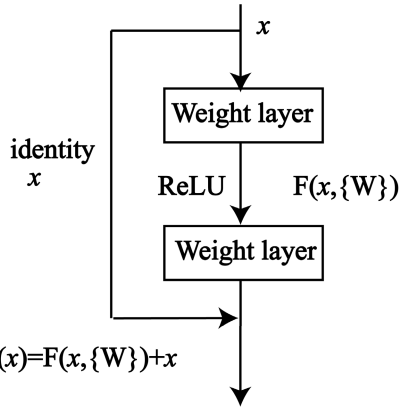


FIGURE 5. Residual learning: a residual block.

The learning process of a residual network can also be derived from the residual block described in equation (7). We assume that x_l is the input to the l-th residual block and y_l is the output; the residual block performs the following computation:

$$y_l = h(x_l) + F(x_l, \{W_l\}) \tag{8}$$
$$x_{l+1} = \delta(y_l) \tag{9}$$

where F denotes the residual function, W_l is a set of weights and biases that are correlative with the l-th residual block, and δ is the ReLU activation function. In the calculation process, the function $h(x_l)$ and function $\delta(y_l)$ are regarded as identical mappings, that is, $h(x_l) = x_l$ and $\delta(y_l) = y_l$. The following equation can be obtained from equations (8) and (9):

$$x_{l+1} = x_l + F(x_l, \{W_l\}) \tag{10}$$
$$x_{l+2} = x_{l+1} + F(x_{l+1}, \{W_{l+1}\}) \tag{11}$$

Thus, we can obtain the following formulation from equations (10) and (11):

$$x_{l+2} = x_l + F(x_l, \{W_l\}) + F(x_{l+1}, \{W_{l+1}\}) \tag{12}$$

After the recursive process, we can obtain

$$x_L = x_l + \sum_{i=1}^{L-1} F(x_i, \{W_i\}) \tag{13}$$

where x_L represents any deeper block L, and x_l represents any shallower block l.

C. BUILDING SE-RESIDUAL LEARNING BLOCK AND SE-Resnet34

In this paper, SE-ResNet34 is established by embedding the SE blocks into convolution units of ResNet34. The process of embedding a SE block into a deep model is to define the mapping function F_{tr} (as shown in Fig. 4). According to the work [11], ResNet34 is stacked with 16 groups of residual blocks, which is referred to as the original residual block. Each group of original residual block consists of two 3×3 convolution layers (as shown in Fig. 6). Regarding the discussion of the embedding method with SE blocks, the transformation F_{tr} is considered the non-identity branch of an original residual block, so that formed SE-original residual block (as shown in Fig. 6), which is the basic unit of SE-ResNet34.

In particular, we denote $W_1 = [w_{11}, w_{12}, \ldots\ldots w_{1d}]$ and $W_2 = [w_{21}, w_{22}, \ldots\ldots w_{2c}]$ as the first filter kernel and second filter kernel, respectively, where w_{1d} refers to the parameters of the d-th filter in the first convolution and w_{2c} refers to the parameters of the c-th filter in the second convolution. We can then write the outputs as $U = [u_1, u_2, \ldots\ldots u_c]$, where:

$$\begin{aligned} u_c &= F_{tr}(X, \{W_1, W_2\}) = w_{2c} * (\delta(w_{1d} * X)) \\ &= \sum_{r=1}^{r=d} w_{2c}^r * (\delta \sum_{s=1}^{c'} w_{1d}^s * x^s) \end{aligned} \tag{14}$$

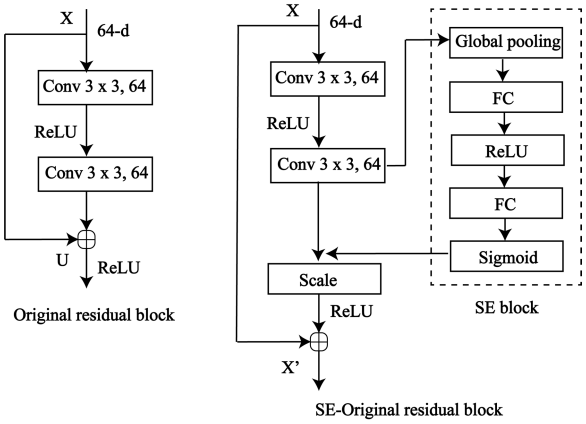


FIGURE 6. The schema of an original residual block and a SE-original residual block.

Here, $*$ denotes the convolution, $w_{1d} = [w_{1d}^1, w_{1d}^2, \ldots, w_{1d}^c]$, $w_{2c} = [w_{2c}^1, w_{2c}^2, \ldots, w_{2c}^d]$, $X = [x^1, x^2, \ldots, x^C]$ and $u_c \in R^{H \times W}$. w_{1d}^c is a 3×3 kernel that represents a single channel of w_{1d}, which acts on the corresponding channel of X. w_{2c}^r is a 3×3 kernel that represents a single channel of w_{2c}, which acts on the corresponding channel of output with the first convolution. By substituting equation (14) into equation (4)-(6), the SE block output X' in this paper can be obtained. In combination with equations (7) and (13), the equation of an SE-original residual block can be obtained as follows:

$$X' = F_{tr}(X, \{W_1, W_2\}) + X \tag{15}$$

We can infer the learning process of the SE-residual network as equation (16), by referring to the learning process of the residual network with equation (8) to equation (13).

$$X_L' = X_l' + \sum_{i=1}^{L-1} F(X_i', \{W_i\}) \tag{16}$$

where X_L' represents any deeper SE-original residual block L, and X_l' represent any shallower SE-original residual block l. As a result, the structures of ResNet34 and SE-ResNet34 are established as shown in Table 1.

D. BUILDING MEIT

MEIT is executed in a batch-sample by randomly sampling P traffic state classes and K samples of each class. For the i-th anchor x_a^i in a batch-sample, we selected the positive sample x_p^i, which belongs to the same class as anchor x_a^i and is the farthest from anchor x_a^i by the Euclidean distance. Additionally, we selected the negative sample x_n^i, which belongs to classes that differ from those of anchor x_a^i and is the nearest to anchor x_a^i by the Euclidean distance. Once all the triplets in a batch-sample have been identified, the output features of SE-ResNet34 are mapped to a metric space by the MEIT, where the distance of samples in the same class is minimized and the distance of the samples between two different classes is

TABLE 1. Parameter settings and convolution operations of Resnet-34 (Left) and SE-Resnet-34 (Right). The shapes and operations with specific parameter settings of an original residual block and a SE-original residual block are listed inside the brackets, and the number of stacked blocks in a stage is presented outside. The inner brackets following by fc indicates the output dimension of the two fully connected layers in an SE block.

Output size	ResNet-34		SE-ResNet-34	
112×112	conv,7 × 7,64,stride 2			
56×56	max pool,3 × 3,stride 2			
56×56	[conv,3 × 3,64] [conv,3 × 3,64]	×3	[conv,3 × 3,64] [conv,3 × 3,64] [fc,[4,64]]	×3
28×28	[conv,3 × 3,128] [conv,3 × 3,128]	×4	[conv,3 × 3,128] [conv,3 × 3,128] [fc,[8,128]]	×4
14×14	[conv,3 × 3,256] [conv,3 × 3,256]	×6	[conv,3 × 3,256] [conv,3 × 3,256] [fc,[16,256]]	×6
7×7	[conv,3 × 3,512] [conv,3 × 3,512]	×3	[conv,3 × 3,512] [conv,3 × 3,512] [fc,[32,512]]	×3
1×1	global average pool, 10-*d* fc			

maximized. The specific equations of the MEIT are presented as follows:

$$L(\theta; X) = \sum_{i=1}^{P}\sum_{a=1}^{K}[margin + \max D(f_\theta(x_a^i), f_\theta(x_p^i)) - \min D(f_\theta(x_a^i), f_\theta(x_n^i))]_+ \quad (17)$$

$$D(f_\theta(x_i), f_\theta(x_j)) = \|f_\theta(x_i) - f_\theta(x_j)\|_2^2 \quad (18)$$

$$[f_\theta(.)]_+ = \begin{cases} 0, & f_\theta(.) < 0 \\ f_\theta(.), & f_\theta(.) \geq 0 \end{cases} \quad (19)$$

$D(.)$ is the Euclidean distance function between two points in the embedded space, as shown in equation (18). $f(.)$ is the nonlinear mapping function of CNN, where is SE-ResNet34 in this paper. $[.]_+$ represents the *hinge* function, as shown in equation (19), which ensures that the output value is greater than or equal to 0. The *margin* represents the distance thresholds of $\max D(f(x_a^i), f(x_p^i))$ and $\min D(f(x_a^i), f(x_n^i))$, where a *softmargin* is another option for model training. The *hinge* function aims to avoid the effect of "*corrected triplet samples*" but training "*corrected triplet samples*" is helpful to obtain more obvious gaps between two classes [48]. Therefore, the *softplus* function $ln(1 + \exp(.))$, which has a similar behavior to that of the *hinge* function, can replace the *hinge* function by performing a smooth approximation. The *softplus* function decays exponentially rather than in a certain numerical manner; thus, the *softplus* function is referred to as the *softmargin* method.

In this paper, softmax loss is introduced for joint training to complete the classification task. The final equation of the MEIT is expressed as follows, where λ is the weight of the improved triplet loss:

$$L_{MEIT} = \lambda L(\theta; X) + L_{softmax} \quad (20)$$

Owing to equation (16), equation (20) and the chain rule of backpropagation, $\frac{\partial L_{MEIT}}{\partial X_l'}$ can be calculated as

$$\frac{\partial L_{MEIT}}{\partial X_l'} = \frac{\partial L_{MEIT}}{\partial X_L'}(1 + \frac{\partial}{X_l'}\sum_{i=1}^{L-1} F(X_i', \{W_i\})) \quad (21)$$

V. EXPERIMENTS

A computer with an Intel i7 CPU @ 4.2 GHz and 1080 Ti GPU is utilized to train our dataset. Pytorch 0.4.0 with Python 3.7 is employed to realize the proposed TrafficNet and baseline for multilevel traffic state detection.

A. ALLOCATION OF DATASET

We annotated 30,000 traffic images that contain 25 different scenes from Xian, Shaanxi, China with the new definition of multilevel traffic state and transformed the labeled traffic data into 10 levels marked as [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], with the value span of 0.1 in each class, as shown in Table 2. The resolutions of traffic images in this dadaset vary from 352 × 288 to 1920 × 1080. We selected typical images from each scene, which contains different weather and light conditions, as shown in Fig. 7. The labeled dataset were divided into training dataset, validation dataset and test dataset with the ratio of 7:1.5:1.5 for each traffic state level. The training dataset, which contains 21,000 images, is applied to learn all the parameters of TrafficNet and the baseline. The validation data, which contains 4500 images, is used to fine-tune the parameters of TrafficNet and determine the optimal hyper-parameters of the EMIT. The test dataset, which contains 4500 images, was employed to evaluate the performances of the models in this paper. In addition, we also labeled two 10-minute traffic videos, including 2 scenes, for video traffic state detection.

B. IMPLEMENTATION

To obtain the initial models, TrafficNet and the baseline in this paper were pre-trained using the ImageNet 2012 dataset in reference [11], including ResNet34 with softmax loss, ResNet34 with triplet loss, ResNet34 with MEIT, SE-ResNet34 with softmax loss, SE-ResNet34 with triplet loss and TrafficNet(SE-ResNet34 with MEIT). Next, we retrained these models with the multilevel traffic state dataset. The input image sizes of all models were uniformly resized to 224×224. Additionally, each input image was normalized by mean RGB-channel subtraction. TrafficNet performed data augmentation (abbreviated as aug) with random cropping and random horizontal flipping. TrafficNet and each baseline was trained with identical optimization schemes, which was the Adam optimizer [49] with $\epsilon = 10^{-3}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. To determine the ideal optimization scheme, the parameter attenuation strategy reported in research [50] was adopted in this paper, as shown in equation (22), where $\epsilon_0 = 10^{-3}$, $t_0 = 15000$, and $t_1 = 25000$. Normally, the training speed changed to $\beta_1 = 0.5$ when the number of training iterations reached t_0, and training stopped when the

TABLE 2. Description of multilevel traffic state dataset.

Traffic state index C_T	$[0.0, 0.1)$	$[0.1, 0.2)$	$[0.2, 0.3)$	$[0.3, 0.4)$	$[0.4, 0.5)$	$[0.5, 0.6)$	$[0.6, 0.7)$	$[0.7, 0.8)$	$[0.8, 0.9)$	$[0.9, 1.0]$
Traffic state levels	0	1	2	3	4	5	6	7	8	9
Number of samples	2550	2907	2323	3298	2850	2888	3378	3398	2987	3421

FIGURE 7. Labeled images of the traffic state dataset, which contains different weather, illumination and road condition. Each image represents a traffic scene for a total of 25 scenes. There are sunny and rainy weathers in this dataset and 2 scenes are recorded at night. Different scenes contain different lane numbers including 2 3 4 5 and complicated crossroads.

number of training iterations reached t_1. In addition, accuracy (referred to as *Acc*) is adopted as the evaluation index of model performance.

$$\epsilon(t) = \begin{cases} \epsilon_0, & t < t_0 \\ \epsilon_t 0.001 \dfrac{t - t_0}{t_1 - t_0}, & t_0 \le t \le t_1 \end{cases} \tag{22}$$

C. OPTIMAL SELECTION OF HYPERPARAMETERS

MEIT requires sufficient samples to ensure that the farthest positive and the nearest negative samples are representative in a bath-sample. On the other hand, the model performance is not distinct when the batch size reaches a certain value, and too many samples will increase the training time. To determine the suitable size of a batch-sample (referred to as *Batchsize*), we designed the number of classes P as 8 and the samples of each class $K \in [4, 8, 16, 32, 64, 128, 256]$ to assess the performance of TrafficNet in validation dataset and test dataset by fixing *margin* as *softmargin* and λ as 1, since the values of *margin* and λ are not related to *Batchsize*. The result is shown in Fig. 8.

As shown in Fig. 8, when *Batchsize* was 32, *Acc* was low. With the increase in *Batchsize*, the performance was significantly improved. *Acc* was nearly saturated when *Batchsize* reached 256. After this stage, an increase in *Batchsize* had minimal impact on the performance. In addition, the choice of *Batchsize* also affects the convergence rates of models. Choosing a larger *Batchsize* would reduce the number of iterations and increase the time of each iteration [41], [47], which synthetically reduced the convergence rate of the models. However, the size of *Batchsize* had inapparent influences on the training time in experiments of this paper and the convergence rate of all models was in an acceptable range. Therefore, the influence of *Batchsize* on model's accuracy is mainly discussed. Based on the previously mentioned considerations, the best *Batchsize* for the training dataset was 256 in this study.

In addition, MEIT can be decomposed into two parts for interpretation according to equation (20). The first part, which is $\lambda L(\theta; X)$, involves batch hard mining of the triplet samples, which improves the features discrimination of the multilevel traffic state. The second part is the *softmax* loss of $L_{softmax}$, which aims to realize multi-classification. The value

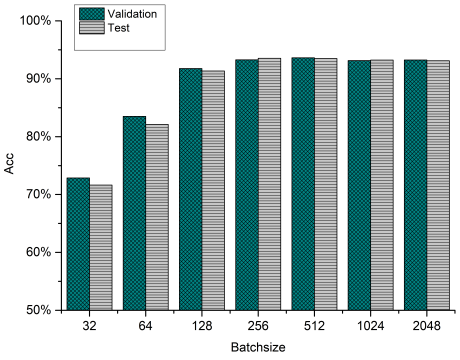


FIGURE 8. Influence of Batchsize on TrafficNet. The horizontal axis is Batchsize, and the vertical axis is *Acc*.

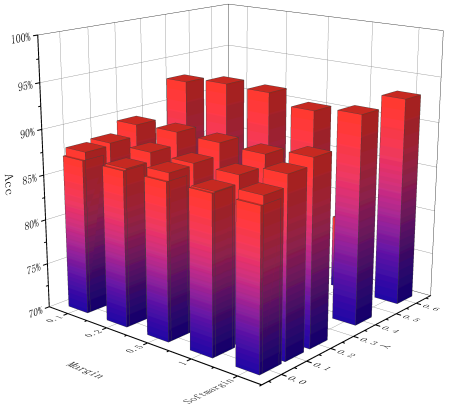


FIGURE 9. The 3 dimensional histogram of *Acc* values with the *margin* and λ. The x-axis and y-axis represent the *margin* and the λ, respectively. The Z-axis represents the *Acc*.

of the *margin* in the function $L(\theta; X)$ directly determines the quality of extracted features, and the value of λ represents the weight between the two parts of MEIT. Determining the values of two hyper-parameters, which jointly promote the selection and classification of multilevel traffic state features, is crucial. Orthogonal experiments with *margin* ∈ [0.1, 0.2, 0.5, 1,*softmargin*] and λ ∈ [0, 0.01, 0.1, 0.2, 0.4, 0.6, 1] were conducted using the validation dataset to search the most appropriate hyper-parameters values. The results are listed in Table 3.

TABLE 3. *Acc* of TrafficNet in validation dataset using different *margin* and λ.

Acc / λ \ $margin$	0.1	0.2	0.5	1	$softmargin$
0	87.07%	87.07%	87.07%	87.07%	87.07%
0.01	87.91%	87.23%	87.90%	87.16%	87.97%
0.1	88.19%	88.21%	88.12%	88.04%	89.25%
0.2	89.46%	89.61%	89.48%	89.31%	90.05%
0.4	92.96%	93.54%	93.42%	92.34%	92.87%
0.6	78.32%	79.31%	79.22%	78.21%	93.11%
1	18.86%	10.00%	10.25%	9.25%	**94.89%**

As indicated by the results in Table 3, the best hyper-parameters combination was [*margin* = *softmargin*, *lambda* = 1], whose *Acc* reached 94.89%. The hyper-parameters combination, that *margin* adopts a hard-boundary value and λ = 1, as shown in the red part of table 3, was almost invalid. The improved triplet loss function $L(\theta; X)$ includes a *hinge* function that trains classification models by a hard-partition method of scoring each class, when a hard-boundary value of the *margin* is adopted. Conversely, the softmax loss function $L_{softmax}$ is employed for classification by calculating the probability of belonging to each class, which is a soft training method due to its fuzzy division. Therefore, when the weights of the *hinge* function and softmax loss function are equal, that is, λ = 1, the learning bias of MEIT is eliminated, which causes model failure.

The 3 dimensional histogram of *Acc* values with the *margin* and λ were drawn, as shown in Fig. 9. First, the influence of λ on TrafficNet performance was analyzed. When the *margin* was fixed as a hard-boundary value, *Acc* increased with an increase in λ until the latter reached 0.4. Thereafter, increasing λ inhibited the model efficiency. This phenomenon proved that a λ value of 0.4 balances the competition between the soft partition $L_{softmax}$ and the hard partition $L(\theta; X)$, which maximizes *Acc*. When *softmargin* is used to train TrafficNet, the whole equation of MEIT is unified to soft categorization, thus improving *Acc* with increasing values of λ. Next, the influence of the *margin* was analyzed. Even when different values of λ were employed, *Acc* values showed similar trends with the changes in *margin*. The order of the model performances was (*margin* = *softmargin*) > (*margin* = 0.2) > (*margin* = 0.5) > (*margin* = 0.1) > (*margin* = 1). Thus, the soft margin method is more suitable for the multilevel traffic dataset compared with the hard classification methods. And TrafficNet employed the hyper-parameters of *margin* = *softmargin* and λ = 1 in this paper.

D. COMPARISON WITH STATE-OF-THE-ART METHODS

To demonstrate the advantages of TrafficNet, we compared our methods with traditional classification methods. Details of softmax loss and triplet loss can be referred to works [7] and [47]. To ensure fairness, all schemes were conducted using the test dataset of multilevel traffic state and employed the optimal parameter setting. The results are summarized in Table 4.

1) DISCUSSION OF MODEL PERFORMANCE

First, we focused on the effects of SE blocks by comparing model 1 to model 4, model 2 to model 5, and model 3 to model 6. We discover that the network combined with the SE block showed better results than the corresponding model without the SE block. This finding demonstrates that the embedding of the SE block enhanced features discrimination of the multilevel traffic state. Second, we analyzed the effect of the MEIT by comparing models 1 2 3 and models 4 5 6, respectively. We observed that the models using softmax loss had the worst performance. The models that uses traditional

TABLE 4. Performance of TrafficNet and the baseline.

Serial number	Models	*Acc*
1	Resnet34 + $softmax$ loss	89.45%
2	Resnet34 + triplet loss	90.27%
3	Resnet34 + MEIT	92.05%
4	SE-Resnet34 + $softmax$ loss	91.82%
5	SE-Resnet34 + triplet loss	92.64%
6	**TrafficNet**	**93.05%**
7	**TrafficNet+aug**	**94.27%**

triplet loss produced an average performance, and the models that applies MEIT achieved the best performance. Features trained by softmax loss simply divide all hyperspace samples into classes, which is suitable for classification when notable feature gaps among the classes are observed. However, this method is not suitable for the fine classification of the multilevel traffic state, which exhibits extremely small gaps among the classes. Although the traditional triplet loss can effectively gather samples that belong to the same class and widen the inter-class gap among samples belong to different classes, which yields better performance than softmax loss, the randomness of triplet samples mining produces invalid triplets that reduces the speed and accuracy of training. MEIT improved the triplet quality by mining hard positive and hard negative samples in a batch-sample. Thus, MEIT achieved the highest *Acc*. Using data enhancement (aug), model 7 reached an *Acc* of 94.27%. Compared with the original data (model 6), *Acc* increased by 1.31%, which indicates that the deep model needs more data support.

2) DISCUSSION OF COMPUTATIONAL COMPLEXITY

For the proposed TrafficNet to be of practical use, it must provide a good balance between improved performance and increased computation complexity. To illustrate the computational burden of the our method, we take the comparison between ResNet34 and SE-ResNet34 as an example. In a single forward pass for an input image of 224 × 224 pixels, ResNet34 requires 2.62 GFLOPs. And SE-ResNet34 requires 2.64 GFLOPs, corresponding to a 0.76% relative increase over the original ResNet34. In exchange for this slight additional computational burden, the accuracy of the SE-Resnet34 greatly exceeds that of the ResNet34. In practical terms, a single forward and backward pass through ResNet34 takes 130 ms, while SE-Resnet34 takes 143 ms to train a minibatch of 256 images. We believe that the small additional computational costs incurred by SE blocks are justified due to their contribution to model performance. On the other hand, we used the MEIT training SE-ResNet34, which reduced the speed of model convergence. But tracking the model training process can be found that the iteration times of convergence for all models, whose values were between 20000-23000, were very nearly (as shown in Fig. 10 and Fig. 11). Overall, TrafficNet achieved greater accuracy based on the same level of computational complexity and calculation time.

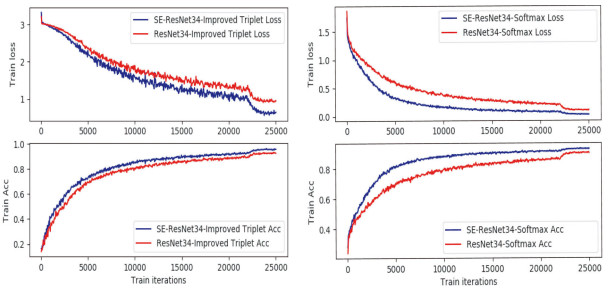


FIGURE 10. Loss and *Acc* changes in SE-ResNet34 and ResNet34.

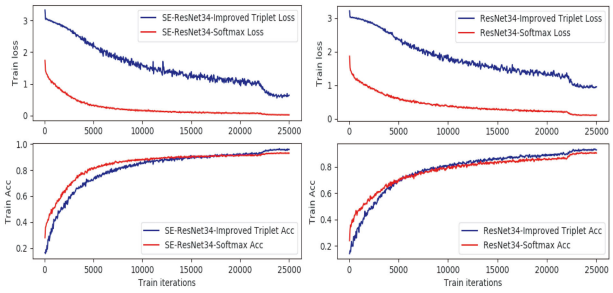


FIGURE 11. Loss and *Acc* changes in the improved triplet loss (MEIT) and softmax loss.

E. MODEL TRAINING TRACKING AND VISUALIZATION

To more intuitively examine the advantages of TrafficNet, this paper also visualized the change rules of the loss function and *Acc* in TrafficNet and traditional models (as shown in Fig. 10 and Fig. 11).

As shown in Fig. 10, regardless of whether MEIT or softmax loss was employed, SE-ResNet34 was lower than ResNet34 in terms of loss value and was higher than ResNet34 in terms of *Acc* at the same point of iteration progress. This phenomenon also proves that the introduction of SE blocks reduces the interference of the noise information and improves the speed and accuracy of training. As shown in Fig. 11, regardless of whether the SE block was introduced, the values of MEIT was always higher than the values of softmax loss. The reason is that MEIT involves batch hard mining of the triplet loss and softmax loss, which yields a large cardinal value. According to the declining trend of the loss curve, softmax loss decreased more rapidly and required fewer iterations to achieve convergence, which shows that the MEIT training is relatively slow. In addition, in the initial stage of training, the *Acc* of softmax loss was higher than that of the MEIT. With increasing training iterations, however, the *Acc* of the EMIT gradually catch up and surpass until reached convergence. In conclusion, TrafficNet ensures the best detection accuracy while achieving a high level of computational efficiency.

To intuitively verify the advantages of the learned features of MEIT, we also mapped the final outputs of SE-ResNet34 to 2-dimensional space in the test dataset by different loss functions and visualized the positions of all samples in the mapped space. The visualization diagram is shown in Fig. 12.

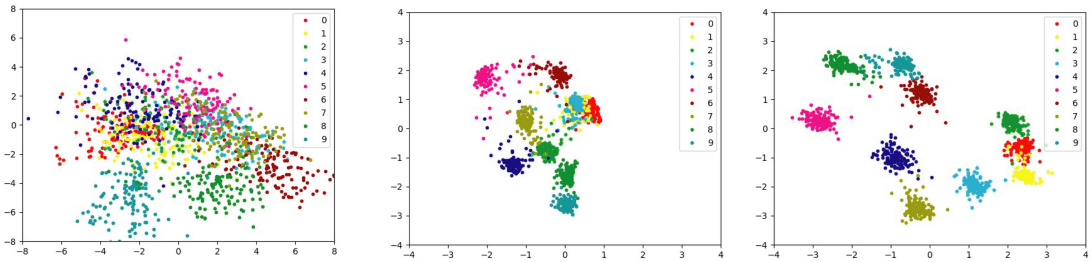


FIGURE 12. The distribution of features learning by different loss function in 2-dimension space. The dots with different colors represent the correctly predicted samples in the different classes. Fig (a) shows the features distribution of softmax loss, Fig (b) shows the features distribution of the traditional triplet loss, and Fig (c) is the features distribution of MEIT.

TABLE 5. Prediction probability of TrafficNet and baseline for typical images in test dataset.

prediction probability / models / Traffic state levels	Resnet34 + $softmax$ loss	Resnet34 + triplet loss	Resnet34+ MEIT	SE-Resnet34 + $softmax$ loss	SE-Resnet34+ triplet loss	TrafficNet+ aug
Typical image 1, Ground truth=0						
0	**80.54%**	**83.65%**	**87.38%**	**85.34%**	**91.25%**	**95.11%**
1	14.32%	12.43%	10.47%	12.26%	6.56%	4.89%
2	2.44%	1.68%	1.15%	1.22%	2.19%	0%
3	1.98%	1.15%	1%	1.18%	0%	0%
4	0.72%	1.09%	0%	0%	0%	0%
Typical image 2, Ground truth=9						
5	0.37%	0.47%	0%	0%	0%	0%
6	1.06%	1.07%	0.90%	1.39%	0%	0%
7	3.17%	2.44%	2.22%	2.69%	2.73%	1.56%
8	14.11%	12.61%	10.54%	10.86%	7.23%	5.04%
9	**81.29%**	**83.41%**	**86.34%**	**85.06%**	**90.04%**	**93.40%**

As shown in Fig. 12, softmax loss can simply distinguish the classes of each sample, that is, to determine the colors of the different points in the visual image. However, the distribution positions of the samples were relatively scattered, and the distance among different classes was small, which produced indistinct classes. With traditional triplet loss training, the overall samples were more concentrated with a distinct distance among the classes. The inter-class distance of the features trained by MEIT was much larger than that of the features trained by the traditional triplet loss, which made the features more representative.

In addition, we extracted two representative traffic images from the test dataset, where the traffic state levels were 0 and 9. We visualized the prediction probability of the traffic state with TrafficNet and each baseline (as shown in Table 5). For the image with a traffic state level of 0, TrafficNet predicted only two possible levels-0 and 1-and the prediction probability of the 0-th level reached 95.11%, which was the highest among all models. TrafficNet improved the accuracy by 14.57% compared with ResNet34 + softmax loss. For the image with a traffic state level of 9, the correct prediction probability of TrafficNet was 93.40%, which was the highest among all models. Compared with ResNet34 + softmax loss, TrafficNet improved the accuracy by 12.11%. These results verified that TrafficNet has great advantages in multilevel traffic state identification.

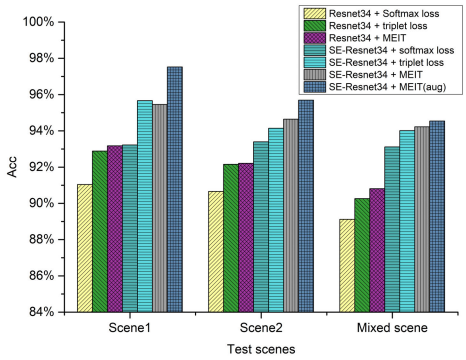


FIGURE 13. *Acc* of each deep model in different traffic scenes. The horizontal coordinate represents either two scenes or one mixed scene.

F. TRAFFIC STATE DETECTION BASED ON VIDEOS

Two 10-minute traffic videos of 2 scenes were collected to test the performance of TrafficNet in video data. We compared the detection performance of TrafficNet and baseline in this paper for a single scene and multiple scenes, as shown in Fig. 13. Multiple scenes data were randomly selected, with 5-minute videos of scene 1 and 5-minute videos of scene 2, respectively. In addition, we visualized continuous detection results of TrafficNet with ground truth in scene 1 and scene 2, respectively (as shown in Fig. 14) and extracted the detection results of typical sequence images in the video of scene 1 (as shown in Fig. 15).

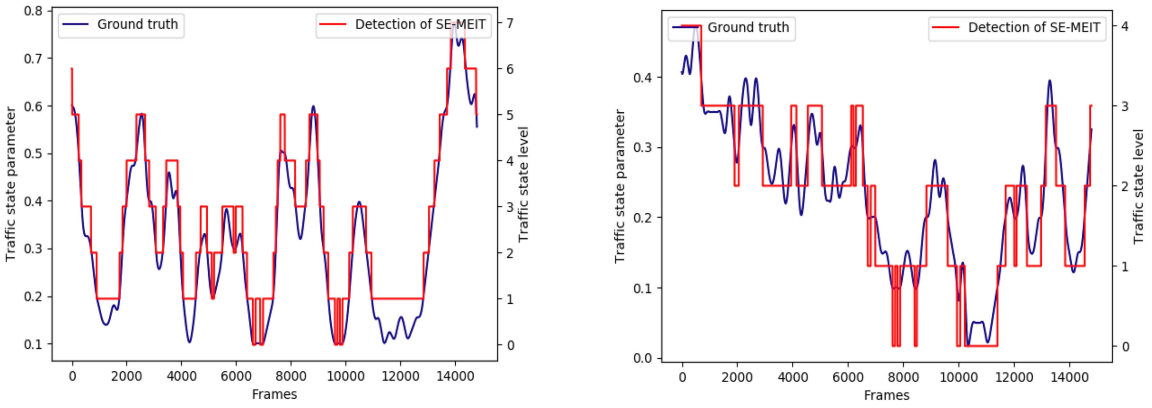


FIGURE 14. Visualization diagram of actual video detection. Fig (a) represents the detection of scene 1, and Fig (b) represents the detection of scene 2.

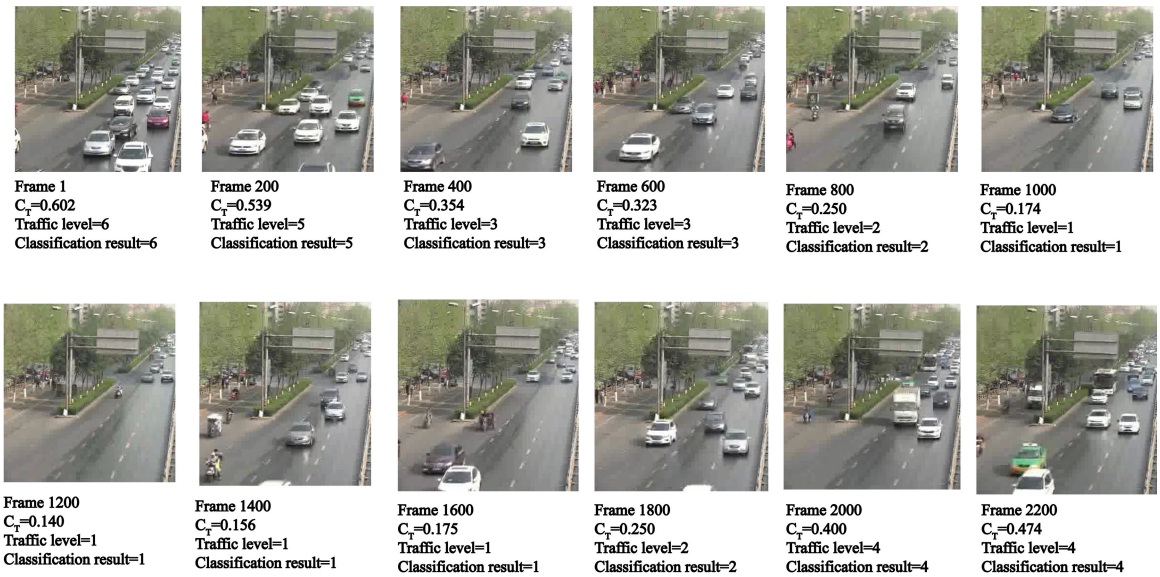


FIGURE 15. The detection results of TrafficNet in typical images, which is extracted at an interval of 200 frames in a video data of scene1. This figure proved that TrafficNet is also applicable to video detection and we can observe the continuous change process of traffic state in videos.

From the results in Fig. 13, we can obtain the following conclusions. First, the detection performance of each deep model for a single scene was better than that for the mixed scenes, which proves that the scene changes suppress the recognition ability of the models. Second, the performance of the models for scene 1 was slightly better than that for scene 2. Scene 2 was a monitoring video of a toll booth, which occupied a small proportion of all scenes in dataset. Thus, TrafficNet had a low bias toward toll station recognition, and the expression ability of its features was slightly weak. Last, TrafficNet remained the most effective detection model for a single scene.

Fig. 14 and Fig. 15 shows that the multilevel traffic state detected by TrafficNet coincided with the ground truth, which demonstrates that TrafficNet is suitable for video detection in practice. Moreover, the recognition results do not exhibit the jumping level phenomenon. These results proved that multilevel traffic state detection is more convenient for smoothly observing the changes in traffic conditions, which is beneficial for managers who implement traffic dredging measures in advance.

VI. CONCLUSION

In this paper, faced with the problem of no definition of image-based multilevel traffic state, we used the idea of applying the vehicle and road area proportions in an image to quantify the levels of traffic state and established an accurate and unified multilevel traffic state dataset. Then, a new model named TrafficNet was proposed, which embeds a visual attention mechanism of SE blocks into ResNet34 and using a deep metric learning method of MEIT as training loss function to solve the challenges due to the noise information in traffic images and the extreme similarities between adjacent classes. Based on the multilevel traffic state dataset,

experiments were conducted to determine suitable parameters for TrafficNet and to verify that the performance of TrafficNet is superior to those of traditional classification models without SE blocks or MEIT. In addition, we tracked the training processes of all models and visualized those features spatial distribution in two-dimension space, where further verified TrafficNet's strong capability of feature extraction and classification for multilevel traffic state. Finally, video testing using actual scenes showed that TrafficNet has high practical value, which enables management with real-time, accurate and useful traffic state information.

REFERENCES

[1] Y. Yuan, W. Dong, and Q. Wang, "Anomaly detection in traffic scenes via spatial-aware motion reconstruction," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1198–1209, Feb. 2016.

[2] Y. Yuan, J. Fang, and Q. Wang, "Online anomaly detection in crowd scenes via structure analysis," *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 562–575, Mar. 2015.

[3] S. Bahrami and M. J. Roorda, "Optimal traffic management policies for mixed human and automated traffic flows," *Transp. Res. A, Policy Pract.*, vol. 135, pp. 130–143, May 2020, doi: 10.1016/j.tra.2020.03.007.

[4] S.-Y. Cho, T. W. S. Chow, and C.-T. Leung, "A neural-based crowd estimation by hybrid global learning algorithm," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 29, no. 4, pp. 535–541, Aug. 1999.

[5] G.-J. Kim, K.-Y. Eom, M.-H. Kim, J.-Y. Jung, and T.-K. Ahn, "Automated measurement of crowd density based on edge detection and optical flow," in *Proc. 2nd Int. Conf. Ind. Mechatronics Autom.*, Wuhan, China, May 2010, pp. 553–556.

[6] A. Sobral, L. Oliveira, L. Schnitman, and F. D. Souza, "Highway traffic congestion classification using holistic properties," in *Proc. Comput. Graph. Imag. Signal Process.*, Shanghai, China, 2013, pp. 374–516.

[7] X. Liang, X. Wang, L. Zhen, S. Liao, and S. Z. Li, "Soft-margin softmax for deep classification," in *Proc. ICONIP* Guangzhou, China, 2017, pp. 413–421.

[8] W. Shi, Y. Gong, X. Tao, D. Cheng, and N. Zheng, "Fine-grained image classification using modified DCNNs trained by cascaded softmax and generalized large-margin losses," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 683–694, Mar. 2019.

[9] Y. Yuan, J. Wan, and Q. Wang, "Congested scene classification via efficient unsupervised feature learning and density estimation," *Pattern Recognit.*, vol. 56, pp. 159–169, Aug. 2016.

[10] Q. Wang, J. Wan, and Y. Yuan, "Locality constraint distance metric learning for traffic congestion detection," *Pattern Recognit.*, vol. 75, pp. 272–281, Mar. 2018.

[11] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 29, 2019, doi: 10.1109/TPAMI.2019.2913372.

[12] Y. J. H. Y. Yan; Hao Xu and B. L. Xu, "Image clustering via deep embedded dimensionality reduction and probability-based triplet loss," *IEEE Trans. Image Process.*, vol. 29, pp. 5652–5661, 2020, doi: 10.1109/TIP.2020.2984360.

[13] U. K. Sreekumar, R. Devaraj, Q. Li, and K. Liu, "TPCAM: Real-time traffic pattern collection and analysis model based on deep learning," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, San Francisco, CA, USA, Aug. 2017m pp. 1–4.

[14] C. Zhan, X. Duan, S. Xu, Z. Song, and M. Luo, "An improved moving object detection algorithm based on frame difference and edge detection," in *Proc. 4th Int. Conf. Image Graph.*, Chengdu, Sichuan, China, 2007, pp. 519–523.

[15] A. Bainbridge-Smith and R. G. Lane, "Determining optical flow using a differential method," *Image Vis. Comput.*, vol. 15, no. 1, pp. 11–22, Jan. 1997.

[16] K. Yang, A. Pan, Y. Yang, S. Zhang, S. Ong, and H. Tang, "Remote sensing image registration using multiple image features," *Remote Sens.*, vol. 9, no. 6, p. 581, Jun. 2017.

[17] K. G. Derpanis and R. P. Wildes, "Classification of traffic video based on a spatiotemporal orientation analysis," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Kona, HI, USA, Jan. 2011, pp. 606–613.

[18] A. Riaz and S. A. Khan, "Traffic congestion classification using motion vector statistical features," in *Proc. 6th Int. Conf. Mach. Vis. (ICMV)*, London, U.K., Dec. 2013, pp. 277–786.

[19] E. Dallalzadeh, D. S. Guru, and B. S. Harish, "Symbolic classification of traffic video shots," in *Advances in Computational Science, Engineering and Information Technology*. Berlin, Germany: Springer, 2013, pp. 125–164.

[20] P. Wang, W. Hao, Z. Sun, S. Wang, E. Tan, L. Li, and Y. Jin, "Regional detection of traffic congestion using in a large-scale surveillance system via deep residual TrafficNet," *IEEE Access*, vol. 6, pp. 68910–68919, 2018.

[21] X. Ke, L. Shi, W. Guo, and D. Chen, "Multi-dimensional traffic congestion detection based on fusion of visual features and convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2157–2170, Jun. 2019.

[22] D. Zanca, M. Gori, and A. Rufa, "A unified computational framework for visual attention dynamics," in *Proc. 2nd Conf. Math. Modeling Motor Neurosci.*, Pavia, Italy, 2018, pp. 183–188.

[23] M. Ranzato and G. E. Hinton, "Modeling pixel means and covariances using factorized third-order Boltzmann machines," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 2551–2558.

[24] L. Lin, H. Luo, R. J. Huang, and M. Ye, "Recurrent models of visual co-attention for person re-identification," *IEEE Access*, vol. 7 pp. 8865–8875, 2019.

[25] T. Klein and M. Nabi, "Attention is (not) all you need for commonsense reasoning," 2019, *arXiv:1905.13497*. [Online]. Available: http://arxiv.org/abs/1905.13497

[26] H. Y. Li, J. Chen, R. M. Hu, and M. Yu, "Action recognition using visual attention with reinforcement learning," in *Proc. 25th Int. Conf. MultiMedia Modeling*, Thessaloniki, Greece, 2019, pp. 365–376.

[27] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," 2015, *arXiv:1506.02025*. [Online]. Available: http://arxiv.org/abs/1506.02025

[28] P. Sermanet, A. Frome, and E. Real, "Attention for fine-grained categorization," 2014, *arXiv:1412.7054*. [Online]. Available: http://arxiv.org/abs/1412.7054

[29] X. Liu, T. Xia, J. Wang, Y. Yang, F. Zhou, and Y. Lin, "Fully convolutional attention networks for fine-grained recognition," 2016, *arXiv:1603.06765*. [Online]. Available: http://arxiv.org/abs/1603.06765

[30] R. Ozawa and M. Minami, "Cognitive resource allocation optimization for real-time multiple object recognition," in *Proc. SICE Annu. Conf.*, Chofu, Japan, Aug. 2008, pp. 1848–1853.

[31] S. Qu, Y. Xi, and S. Ding, "Visual attention based on long-short term memory model for image caption generation," in *Proc. 29th Chin. Control Decision Conf. (CCDC)*, Chongqing, China, May 2017, pp. 4789–4794.

[32] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, May 1992.

[33] T. Bluche, "Joint Line segmentation and transcription for end-to-end handwritten paragraph recognition," in *Proc. NIPS* Barcelona, Spain, 2016, pp. 838–846.

[34] A. Miech, I. Laptev, and J. Sivic, "Learnable pooling with context gating for video classification," 2017, *arXiv:1706.06905*. [Online]. Available: http://arxiv.org/abs/1706.06905

[35] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, D. Ramanan, and T. S. Huang, "Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 2956–2964.

[36] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6298–6306.

[37] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip reading sentences in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 3444–3450.

[38] W. Fei, "Residual attention network for image classification," in *Proc. CVPR* Honolulu, HI, USA, 2017, pp. 3156–3164.

[39] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. ECCV*, Munich, Germany, 2018, pp. 3–19. [Online]. Available: https://link.springer.com/conference/eccv

[40] A. Newell, K. Yang, and D. Jia, "Stacked hourglass networks for human pose estimation," in *Proc. ECCV* Amsterdam, The Netherlands, 2016, pp. 483–499.
[41] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.
[42] H. Dong, L. Ping, Z. Shan, C. Liu, J. Yi, and S. J. N. Gong, "Person re-identification by enhanced local maximal occurrence representation and generalized similarity metric learning," *Neurocomputing*, vol. 307, pp. 25–37, Sep. 2018.
[43] Z. Li, X. Tao, and S. Gong, "CBAM: Learning a Discriminative Null Space for Person Re-identification," in *Proc. CVPR*, Seattle, WA, USA, 2016, pp. 1239–1248.
[44] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Learning to rank in person re-identification with metric ensembles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1846–1855.
[45] S. Ding, L. Lin, G. Wang, and H. Chao, "Deep feature learning with relative distance comparison for person re-identification," *Pattern Recognit.*, vol. 48, no. 10, pp. 2993–3003, Oct. 2015.
[46] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based CNN with improved triplet loss function," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2016, pp. 1335–1344.
[47] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 815–823.
[48] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2016, pp. 4004–4012.
[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[50] D. Wang, X. Y. Zhu, and Y. Liu, "Multi-layer channel normalization for frequency-dynamic feature extraction," *Softw. Qual. J.*, vol. 14, no. 9, pp. 1523–1529, Sep. 2003.

FENG TANG was born in Huaihua, China, in 1992. He received the B.S. degree in civil engineering from the Changsha University of Science and Technology, Changsha, China, in 2015. He is currently pursuing the Ph.D. degree with the School of Civil and Transportation Engineering, South China University of Technology, Guangzhou, China. His research interests include traffic safety analysis and intelligent transportation system (ITS).

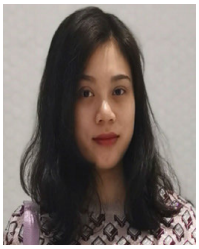
XINSHA FU was born in Changsha, China, in 1955. He received the B.S. degree in highway and bridge engineering from the Changsha University of Science and Technology, Changsha, in 1981, and the M.S. degree in highway engineering and the Ph.D. degree from Chang'an University, Xian, China, in 1987 and 2008, respectively. From 1981 to 2000, he was a Professor with the Changsha University of Science and Technology. Since 2000, he has been a Professor with the South China University of Technology, Guangzhou, China. His research interests include theoretical research on highway alignment design, research on highway alignment and computer-aided design, road safety audit, and ITS (intelligent transportation system).

Dr. Fu served as China highway society youth expert committee, China's institutions of higher learning civil engineering professional guidance committee, the Director of the institute of computer application, road of China highway society academic committee member, China faces the transport in the 21st century version of the institutions of higher learning materials (highway) editorial committee, the computer aided engineering staff, and a central south highway project staff.

MINGMAO CAI was born in Fuzhou, China, in 1997. He received the B.S. degree in traffic engineering from the Qingdao University of Technology, Qingdao, China, in 2019. He is currently pursuing the M.S. degree with the School of Civil and Transportation Engineering, South China University of Technology, Guangzhou, China. His research interests include traffic safety analysis and intelligent transportation system (ITS).

YUE LU was born in Shijiazhuang, China, in 1992. He received the B.S. degree in civil engineering from the Changsha University of Science and Technology, Changsha, China, and the M.S. degree in highway and railway engineering from the South China University of Technology, Guangzhou, China, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the School of Civil and Transportation Engineering. His research interests include traffic safety analysis and highway route design theory.

YANJIE ZENG received the B.S. degree in civil engineering from Chongqing Jiaotong University, Chongqing, China, in 2013. He is currently pursuing the Ph.D. degree with the School of Civil and Transportation Engineering, South China University of Technology, Guangzhou, China. His research interests include object detection and visual tracking.



SHIYU ZHONG was born in Ganzhou, China, in 1992. She received the B.S. degree from the Southwest China University of Science and Technology, Mianyang, China, in 2013. She is currently pursuing the Ph.D. degree in civil engineering and transportation with the South China University of Technology, Guangzhou, China. Her research interests include transportation data quality issues, intelligent transportation systems, and transportation safety.

YAN HUANG was born in Zhuzhou, China, in 1988. He received the B.S. degree in mechanical engineering from Second Artillery Engineering University, Xian, China, in 2011, and the M.S. degree in traffic engineering from the Changsha University of Science and Technology, Changsha, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Civil and Transportation Engineering, South China University of Technology, Guangzhou, China. His research interests include picture processing and point cloud data processing.

CHONGZHEN LU was born in Shangrao, China, in 1997. He received the B.S. degree in civil engineering from Nanchang Hangkong University, Nanchang, China, in 2013. He is currently pursuing the M.S. degree with the School of Civil and Transportation Engineering, South China University of Technology, Guangzhou, China. His research interests include traffic safety analysis and intelligent transportation system (ITS).