Abstract—A self-driving car is a hot research topic in the field of the intelligent transportation system, which can greatly alleviate traffic jams and improve travel efficiency. Scene classification is one of the key technologies of self-driving cars, which can provide the basis for decision-making in self-driving cars. In recent years, deep learning-based solutions have achieved good results in the problem of scene classification. However, some problems should be further studied in the scene classification methods, such as how to deal with the similarities among different categories and the differences among the same category. To deal with these problems, an improved deep network-based scene classification method is proposed in this article. In the proposed method, an improved Faster region with convolutional neural network features (RCNN) network is used to extract the features of representative objects in the scene to obtain local features, where a new residual attention block is added to the Faster RCNN network to highlight local semantics related to driving scenarios. In addition, an improved Inception module is used to extract global features, where a mixed Leaky ReLU and ELU function is presented, to reduce the possible redundancy of the convolution kernel and enhance the robustness. Then, the local features and the global features are fused to realize the scene classification. Finally, a private dataset is built from the public datasets for the specialized application of scene classification in the self-driving field, and the proposed method is tested on the proposed dataset. The experimental results show that the accuracy of the proposed method can reach 94.76%, which is higher than the state-of-the-art methods.

Index Terms—Deep network, faster region with convolutional neural network features (RCNN), feature fusion, scene classification, self-driving car.

I. INTRODUCTION

WITH the acceleration of urbanization and the rapid development of the social economy, the number of cars continues to increase, and the transportation situation becomes more and more complex. To solve urban traffic congestion, frequent traffic accidents, and other problems, the intelligent transportation system comes into being [1]–[3], which combines various advanced information technologies with the whole ground traffic management system to achieve efficient, convenient, and safe traffic control [4], [5].

The self-driving car is a vehicle that perceives the environment and runs with little or no manual input [6], [7], which is an indispensable part of the intelligent transportation system. Because the driverless system relies on the results of environment perception to make driving behavior decisions, environment perception has become the research hot spot in the self-driving car field [8]. The specific tasks of environment perception in the field of self-driving include scene classification, obstacle detection, lane recognition, and so on [9], [10]. Scene classification is one of the most important and challenging tasks in the self-driving car field because the environment of the traffic is complicated and volatile, and the categories are various [11].

The scene classification of the self-driving car refers to the information of the road and its surroundings is obtained by the onboard camera, radar, or other sensors, and then, the state of the current position is recognized by the corresponding processing methods [12], [13]. To achieve a higher level of intelligent driving, the self-driving car needs to understand the high-level semantic information of its location to make the decision of driving strategy and path planning. For example, the car should slow down near the school, pay attention to the use of anti-skid mode/function in rainy and snowy weather, keep driving at high speed on the highway, and so on [14], [15].

At the beginning of the research on scene classification, most of the existed methods were based on the underlying visual features. For example, Vailaya et al. [16] used the low-level visual features to generate a series of semantic tags to train the binary Bayesian classifiers, which is effective to solve the problem of content-based image scene recognition. However, a single low-level visual feature is difficult to represent complex scene visual content, and the accuracy of scene classification is low. Latte et al. [17] presented a methodology by fusing color features and texture features to recognize certain crop field images. The method of low-level visual feature fusion can improve the scene classification accuracy, but it is difficult to recognize the images outside the training set accurately. Quelhas et al. [18] applied the bag-of-words to scene recognition based on local invariant features.
and probabilistic latent space models, which can improve the generalization ability of the scene recognition method. However, the influence of synonyms and polysemous has not been considered in the bag-of-words, so this method cannot satisfy the requirement of self-driving cars.

Recently, deep learning methods, especially convolutional neural network (CNN)-based methods, have achieved good results in many fields, including image processing and speech recognition [19]–[22]. More and more attention has been paid to scene classification based on deep learning technology. For example, Chen et al. [23] proposed a road scene recognition method based on a multilabel neural network. This network architecture integrates different classification patterns into the training cost function. Zheng and Naji [24] proposed a deep learning neural network for road scene recognition, which can improve the accuracy of the semantic perception of the road scenes in three dimensions, namely, time period, weather, and road type. Tang et al. [25] used the GoogLeNet model for scene recognition, which is divided into three layers from bottom to top. The output features of each layer are fused to generate the final decision of scene recognition. Wang et al. [26] presented a multiresolution CNN network model, including the coarse resolution CNN and the fine resolution CNN, which are used to capture the visual structure at large scale and relatively small scale, respectively.

The methods introduced above provide a good foundation for the scene classification of self-driving cars. However, two main difficulties need to be further studied in the scene recognition of self-driving cars. The first one is that scenes of the same category differ greatly. The second one is that there are visual similarities among different categories of scenes. The main reason for these two difficulties is that there are a variety of objects in the scene, which will influence the recognition of the scene. For example, the same scene may have different forms of expression if the objects in the scene have obvious rotation or shadows. Some examples of these scenes difficult to recognize are shown in Fig. 1.

Fig. 1. Some examples of the scenes in self-driving cars difficult to recognize: (a) street and (b) crosswalk that are different scenes with similar objects; (c) and (d) parking lots that are under different situations.

To deal with these problems of the scene classification for self-driving cars, an improved deep learning-based method is proposed. As we know, the scenes of self-driving cars have a strong correlation with the objects in these scenes, where the objects mainly refer to those representative objects of the usual traffic scenes. For example, the pedestrians and zebra crossings are often contained in the crosswalk scenes. Multiple refueling tanks often exist in gas stations. However, the scene category cannot be decided simply by the representative objects existing in the scene. Based on this idea, in the proposed method, the local features of the representative objects in the scene and the global features of the whole scene image are extracted and fused to realize scene classification accurately.

The main contributions of this article are given as follows.
1) An improved deep network model is proposed for the scene classification of self-driving cars. The network combines local features and global features of the whole scene to improve the accuracy of classification.
2) The model extracts one or more objects with discriminative features in the image by using the pretrained faster region with CNN features (RCNN) network. In the proposed model, the Faster RCNN network is improved by adding a residual connection module based on spatial attention to help the network pay attention to more details and retain more discriminative features.
3) The model uses Inception_V1 to extract the global features of the whole image, where the activation function of the Inception_V1 is replaced by a mixed function with ELU and Leaky ReLU functions to improve the convergence and accuracy of the network. In addition, a special dataset for the scene classification of self-driving cars is set up based on the public datasets of image classification, and various experiments are conducted to test the performance of the proposed method.

This article is organized as follows. Section II describes the proposed method and gives out the structure of the proposed deep network. Experiments for the scene classification of self-driving cars in various situations are conducted in Section III. Section IV discusses the performance of the proposed method and the effectiveness of the presented dataset with some additional comparison experiments. Finally, the conclusion is given in Section V.

II. PROPOSED DEEP NETWORK

In this article, a new deep network is proposed to deal with the problem of the scene classification for self-driving cars, which is shown in Fig. 2. The proposed deep network includes four main parts: the improved Faster RCNN, the improved Inception_V1 module, the feature fusion module, and the classification network. In this study, the improvement of the Faster RCNN is that a residual connection module based on spatial attention is added into the structure of the deep network. The improvement of the Inception_V1 module is that a mixed function with ELU and Leaky ReLU functions is used as the activation function.

As shown in Fig. 2, the first two units are used to extract local features and global features, respectively. The improved Faster RCNN is pretrained, and its output result is the features of the representative objects contained in the image. These representative objects are defined in advance according to
Fig. 2. Total structure of the proposed deep network for the scene classification of self-driving cars.

Fig. 3. Representative objects defined for the scene classification of self-driving cars. (a) Zebra crossings and pedestrians in the crosswalk. (b) Gas tanks in the gas station. (c) Parking cars and lines in the parking lot. (d) Houses in street. (e) Isolation belts in the highway.

common sense and are used as labels for network training. In this study, a total of seven representative objects of the usual traffic scenes are defined, which are shown in Fig. 3. The representative objects are zebra crossing, pedestrian, gas tank, parking car, parking line, house, and isolation belt. The scenes defined in this study are crosswalk, gas station, parking lot, street, and highway. In the proposed method, the predefined representative objects are automatically detected by the improved Faster RCNN. There is no need to artificially decide what objects should be detected during the scene classification process. The output of the improved Inception_V1 network is the global features of the whole image. The feature fusion module fuses local features and global features. These network structures are described in detail as follows.

A. Improved Faster RCNN Network for Local Feature Extraction

1) Structure of the Improved Faster RCNN Network: The local feature extraction is based on the Faster RCNN [27]. The main reason for using the Faster RCNN is that the performances of the Faster RCNN series are significantly better than other networks (see [28] for details). In this study, the structure of the improved Faster RCNN is shown in Fig. 4, where the VGG16 Net is used as the underlying framework to get the feature map of the whole image, which consists of 13 convolution layers and four pooling layers, and the activation function is ReLU. The residual attention module combines the top–down attention map with the input bottom–up convolution features, to obtain the feature map and
output to the next layer. The details about the residual attention module will be introduced in Section II-A2.

In the proposed deep network, the region proposal network (RPN) [29] is used to generate region proposals, where one branch is used to judge whether the anchor belongs to the foreground or background, and the other branch is used to form the bounding box coordinates. RPN generates nine anchors for each pixel in the feature map and sorts the anchors from large to small according to the score of the input foreground. Then, the first 12,000 anchors are selected. Finally, only 2000 anchors are reserved based on the none-maximum suppression (NMS) algorithm (see [30] for details), which are input into the next layer.

The region-of-interest (ROI) pooling network [31] maps the input ROIs to the last layer of the VGG16 network to get a proposal feature map with a fixed size (that is, 300 * 7 * 7 * 512 in this study). Finally, this fixed size feature map is fully connected by the predication network, and the Softmax is used to classify the specific categories. At the same time, fully connected by the predication network, and the Softmax of the RPN network regression loss of the predication network. The loss function location regression loss, the classification loss, and the location classification loss and location regression loss, which is an insensitive parameter and set as 1 in this study [33], [34].

\[ L_{\text{RPN}}(p, t) = \frac{1}{N_{\text{cls}}} \sum_i L_{\text{cls}}(p_i, t_i) + \lambda \frac{1}{N_{\text{reg}}} \sum_i p_i^* L_{\text{reg}}(t_i, t_i^*) \quad (1) \]

where \( N_{\text{cls}} \) is the number of the anchors in the minibatch; \( N_{\text{reg}} \) is the number of the anchor locations; \( t_i = \{t_x, t_y, t_w, t_h\} \) is the predicted coordinates of bounding box for the \( i \)th anchor; \( t_i^* = \{t_x^*, t_y^*, t_w^*, t_h^*\} \) is the real coordinates of the bounding box of the objects; and \( \lambda \) is a coefficient to balance the classification loss and location regression loss, which is an insensitive parameter and set as 1 in this study [33], [34].

\[ p_i \text{ represents the probability that the } i \text{th anchor is predicted as the target, and } p_i^* \text{ is the ground truth, namely,} \]

\[ p_i^* = \begin{cases} 0, & \text{Negative label} \\ 1, & \text{Positive label} \end{cases} \quad (2) \]

The RPN classification loss \( L_{\text{cls}} \) is a cross-entropy, which is denoted by

\[ L_{\text{cls}}(p_i, p_i^*) = -\log[p_i^* p_i + (1 - p_i^*)(1 - p_i)]. \quad (3) \]

The RPN location regression loss \( L_{\text{reg}} \) is denoted by

\[ L_{\text{reg}} = R(t_i - t_i^*) \quad (4) \]

where \( R(\cdot) \) is the smooth L1 loss function, which is defined as

\[ R(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (5) \]

The loss function of the predication network is the same as that of the RPN network, where the classification loss function uses the cross-entropy and the regression loss function uses the smooth L1 loss function.

2) Residual Attention Module: The existing methods of image scene classification mainly focus on the multilayer CNNs, but the large amount of redundant information contained in images is not conducive to scene classification. These CNN-based methods do not clearly distinguish between key information and redundant information; thus, the efficiency and accuracy of image scene classification are affected, and the ability to extract features is limited. The spatial attention mechanism is widely used in visual tasks [35], which can adaptively learn to focus on more prominent regional feature maps in the scene and input these feature maps into the subsequent bottom–up feature extraction process. However, the spatial attention mechanism will lose the previous feature maps, so the bottom–up process is interrupted by the attention model.

To deal with the problem introduced above, a single-layer spatial attention model with residual connection is proposed...
to integrate the attention map and convolution feature map. In the proposed residual attention module (see Fig. 5), the input feature map is first normalized by batch normalization and operated by a single $1 \times 1$ convolution layer. The $1 \times 1$ convolution layer can be used in general to change the filter space dimensionality (see [36] and [37] for details). The purpose of the $1 \times 1$ convolution layer used here is to reduce the number of channels and improve the calculation performance. After the convolution layer, based on the spatial attention module, the attention mask is generated, and different weights are given to different regions of the feature map to get a new feature map. Then, the feature value of any point $(i, j)$ in the feature map processed by the spatial attention mechanism is

$$F_{output}(i, j) = F_{input}(i, j) \odot a_{ij} \quad (6)$$

where $F_{output}$ and $F_{input}$ are the output and input feature values of the point $(i, j)$ through the spatial attention module, respectively; $a_{ij}$ is the attention weight of the point $(i, j)$; and $\odot$ represents the dot product operation.

To avoid the disappearance of feature value before the attention module, a residual connection is introduced; then, (6) is modified as

$$F_{output}(i, j) = F_{input}(i, j) \odot a_{ij} + F_{input}(i, j). \quad (7)$$

**Remark 1:** Based on the proposed residual attention module, the feature map with spatial attention and convolution is combined with the input feature map. Thus, the bottom–up feature extraction process will not be interrupted, and the top information of the image will also be taken into account.

3) **Local Feature Extraction:** The detailed process of the local feature extraction based on the proposed Faster RCNN is introduced as follows. First, based on the pretrained local network, 300 region proposals are generated, and a further NMS algorithm is applied to them (the threshold for the NMS is 0.3 in this study). Then, the target bounding box with confidence greater than 0.5 is detected by the crop layer, which is converted to the position coordinates $[y_1, x_1, y_2, x_2]$ in the original feature map (the feature map after the attention module). The parts in the bounding box are extracted from the original feature map and resized uniformly by the following pooling layer to obtain $N$ local feature maps with the same size (it is $7 \times 7 \times 512$ in this article). Finally, the $N$ local feature maps are fused by an elementwise addition operation (see [38] and [39] for details), which is denoted as follows:

$$Z_{add} = \sum_{i=1}^{N} X_i \quad (8)$$

where $Z_{add}$ represents the fusion feature tensor of single channels and $X_i$ represents the single-channel feature tensor of the $i$th target region (the total number of channels is 512 in this study).

Finally, the local fusion feature vector $Z$ is operated by a flat layer and two fully connected layers, namely,

$$X_p = \varphi_p(X)$$
$$X_f = \varphi_f(X_p) \quad (9)$$

where $\varphi_p$ represents the flat operation; $X_p$ denotes the output of the flat layer, which is a flat tensor; $\varphi_f$ represents a two-layer fully connected operation; and $X_f$ is the final local feature, with the size of $1 \times 5$ in this study.

The pseudocode of the local feature extraction process based on the proposed network is shown in Fig. 6.

**B. Improved Inception_V1 Network for Global Feature Extraction**

In this article, a global feature extraction network is proposed based on the Inception network [40]. The main reason for using the Inception network is that it can extract more information from input images at different scales by using a global average pooling layer instead of a fully connected layer, which can greatly reduce the number of parameters while increasing the depth of the network, and has a good performance in the classification problem. In this study, the Inception_V1 network is improved for the global feature extraction, and its structure is shown in Fig. 7.

As shown in Fig. 7, the Inception_V1 network has nine Inception blocks in total, and each Inception block has four branches. The first branch performs a $1 \times 1$ convolution on the input, which can proceed cross-channel feature transformation to improve the expression ability of the network; the second branch first uses a $1 \times 1$ convolution and then performs a $3 \times 3$ convolution; the third branch uses a $1 \times 1$ convolution and then performs a $5 \times 5$ convolution; and the fourth branch uses a $1 \times 1$ convolution after a $3 \times 3$ max-pooling directly. Each Inception block uses an aggregation operation to combine these four branches. Besides the Inception blocks, there are three convolution layers and two max-pooling layers after the input layer. In addition, there are an average pooling layer and a fully connected layer before the output layer in the Inception_V1 network.

As we know, the activation function is a key part of the deep network, which is used to realize the nonlinear mapping...
Fig. 6. Pseudocode of the local feature extraction process based on the proposed network.

```
Initialize: R = \{r_0, \ldots, r_{N-1}\}, D = \{d_0, \ldots, d_{L-1}\}, \text{batch\_size};
\% R is the list of initial detection boxes, D contains corresponding detection scores, D is the set of final detection.
\% N is the NMS threshold; D_l is the detection threshold; L_s is the number of labels; I_s is the size of input data.
\% \text{batch\_size} is the size of samples selected for once extraction.
For each batch in training image:
    \{ 
        m = arg\max(R);
        M = r_m;
        D = D + M;
        B = B - M;
        For (i, in D) do 
            If \text{IoU}(M, D_i) > \theta, then \% \text{IoU}() is a function to calculate the intersection over union.
                S = S - D_i; \% Remove the redundant detection box.
        EndFor
    \}
EndWhile
Return D, S;
For (i, in D) do 
    If \text{Det}(x_i) > \theta, then \% \text{Det}() is an operation to detect bounding boxes whose confidence > 0.5.
            Save the position of x_i.
        EndIf
EndFor
Call \text{Bounding\_boxes}(x_1, x_2, \ldots, x_k). \% Obtain the position of bounding boxes.
EndFor
Call \text{crop\_and\_resize}(). \% Extract the corresponding feature maps from the last layer of pre-trained network.
Call \text{Max\_pool2D}(). \% Uniform the size of feature maps as 7*7*512.
Call \text{Add}(X); \% Add() is the feature maps fusion operation by element-wise addition.
\% X represents the single-channel feature tensor of the l-th target region.
\% \text{pool}() is the flat operation, where \text{pool}() represents the fusion feature tensor.
\% \text{pool}() is a two-layer fully connected operation, and the final feature map is 1*5.
Output X;
```

Fig. 6. Pseudocode of the local feature extraction process based on the proposed network.

for feature extraction. In the general Inception block, the ReLU function is used as the activation function, which has some defects, such as information loss and neuron death. To deal with the problem of the loss of feature information and improve the rate of convergence, a mixed activation function method is presented in this article.

The mixed activation function method is that the Leaky ReLU and ELU functions are used alternately as the activation function of the convolution layer for the Inception networks. In this study, the Leaky ReLU is used as the activation function of the initial convolution layer of the network first, and then, the ELU function is used alternately. The main reasons for using the Leaky ReLU and ELU functions are that Leaky ReLU uses a small slope instead of the negative axis to reduce the loss of information and alleviate the zero gradient problem, and ELU can relieve gradient disappears and is more stable to the input change. The average output of ELU is close to zero, so the convergence speed is faster.

The expression of Leaky ReLU is given as follows:

\[
y_i = \begin{cases} 
  x_i, & \text{if } x_i \geq 0 \\
  \alpha x_i, & \text{if } \alpha x_i < 0 
\end{cases}
\]  

(10)

where \(\alpha\) is a fixed parameter, which is set as 0.2 in this article.

The expression of ELU is given as follows:

\[
y_i = \begin{cases} 
  x_i, & \text{if } x_i \geq 0 \\
  \exp(x_i) - 1, & \text{if } x_i < 0 
\end{cases}
\]  

(11)

After the convolution and Inception network, the average pooling and fully connected operation proceed, which outputs the final global feature map with the size of 1*5 in this study.

Remark 2: Based on the proposed mixed activation function method of Leaky ReLU and ELU, the problem of information loss and slow convergence can be solved efficiently.

C. Feature Fusion and Classification Network

The last part of the proposed deep network is the feature fusion and the classification network. The fusion method in this article is given as follows.

Suppose that the two input feature vectors are \(X = [a_1, a_2, \ldots, a_N]\) and \(Y = [\beta_1, \beta_2, \ldots, \beta_N]\). When the batch size is 1, the fused feature vector \(Z_{cat}\) is obtained by appending the two input feature vectors \(X\) and \(Y\), namely,

\[
Z_{cat} = [a_1, a_2, \ldots, a_N, \beta_1, \beta_2, \ldots, \beta_N]
\]  

(12)

where the size of the \(Z_{cat}\) is 1*2N, namely, 1*10 in this study. Then, the fused feature vector is sent into the fully connected layer to train the deep network for scene classification.

After the fully connected layer, the size of the feature vector is changed to 1*k, where \(k\) is the number of scene classifications in the experiment (which is 5 in this study). The classification training is carried out through Softmax, and the loss function used in this article is a cross-entropy loss function [41]. The loss function in this article is commonly used in image classification to ensure the maximum probability of positive prediction, which is given as follows:

\[
\text{Loss}_{cls} = \frac{1}{M} \sum_i L_i = \frac{1}{M} \sum_i \left( - \sum_{c=1}^{k} y_{ic} \log(p_{ic}) \right)
\]  

(13)

where \(k\) is the number of categories, namely, the number of scene classification; \(M\) is the number of the samples; \(p_{ic}\) is the probability that the \(i\)-th observation sample belongs to the \(c\)-th category; and \(y_{ic}\) denotes the indicator variable of the \(i\)-th observation sample and the \(c\)-th category, which is defined as follows:

\[
y_{ic} = \begin{cases} 
  1, & \text{if the } i\text{-th sample is the } c\text{-th category} \\
  0, & \text{otherwise}
\end{cases}
\]  

(14)

The whole workflow of the proposed method is summarized as follows.
Fig. 8. Some images with different scenes in the special dataset built for scene classification of self-driving cars in this article.

1) The scene image to be classified is input into the proposed deep network.
2) Generate 300 region proposals for the image by the pretrained improved Faster RCNN. Then, the local features of the image are extracted by performing some subsequent operations on these 300 region proposals, such as NMS, flat, and two-layer fully connection [see (8) and (9)].
3) Meanwhile, the global features of the image are extracted by the improved Inception_V1 network.
4) Then, the local features and global features are fused by (12) to obtain the fused feature vector.
5) Send the fused feature vector to a fully connected layer, and the size of the feature vector is changed to 1 * 5.
6) Conduct scene classification through Softmax classifier, and finally, output the scene category that the image belongs to.

III. EXPERIMENTS

A. Datasets

There are many public datasets widely used for image classification training and testing, such as KITTI and UMC [42], [43]. However, these datasets are not set up for scene classification of self-driving cars especially. Thus, the accuracy and efficiency will be low if these public datasets are used directly for scene classification in self-driving cars. To deal with this problem, a special dataset is established to train and verify the performance of the deep network in the scene classification of self-driving cars. There are five categories of the scene in the proposed dataset, namely, crosswalk, gas station, parking lot, highway, and street. Each category has 15,000 pictures that are selected from the public datasets KITTI [42] and Place365 [26].

The proposed dataset contains various traffic scenes at different locations, time periods, and various light and weather conditions, such as day and night, and cloudy and sunny days. Because the scenes are to be classified by a self-driving car, the images should be obtained by the cameras mounted on the car. Thus, when selecting the images, their availability should be fully considered, including the shooting angle, shooting distance, and representative objects in the images. The workload to establish this special dataset is enormous, which is to ensure the performance of the deep network based on this dataset. Some images with different scenes of the self-driving cars are shown in Fig. 8.

Remark 3: In this study, the special dataset is set up by selecting pictures with clear and distinct representative objects of scene images from the public dataset because the self-driving car will only be in one scene at the same time in most cases. Other complex situations will be further studied in the future.
The structure parameters of the proposed network

Fig. 9. Summary of the structure for the proposed deep network.

**TABLE I**

| Parameters of the Proposed Deep Network and the Experimental Environment |
|---------------------------------------------------------------|
| **Network** | **Learning rate** | 0.01 |
|              | **Momentum**      | 0.9  |
|              | **Dropout keep probability** | 0.8 |
|              | **Batch size**    | 16   |
|              | **Max. iters**    | 20000|
|              | **Validation interval** | 500 |
| **Environment** | **Programming software** | Python3.6 |
|                | **Deep learning framework** | TensorFlow |
|                | **Computer System** | Windows 10 |
|                | **Cpu**           | 2.10GHz |
|                | **RAM**           | 32.0GB |
|                | **Gpu**           | GeForce RTX 2080 |

### B. Experimental Results of the Proposed Method

The experiments are conducted on a computer with Windows 10 System, and the deep network is realized on the TensorFlow deep learning framework with Python 3.6 [44]. A simple summary of the structure for the proposed deep network is shown in Fig. 9, which is introduced in detail in Section II. The parameters of the proposed deep network and the experimental environment are listed in Table I.

The proportion of training and test sets in the whole dataset is 6:4, and the proportion of the train and validation set in the training set is 1:1. The size of the input images to the deep network is $224 \times 224$. The curve of the loss function and the accuracy of the proposed deep network after 20000 iterations on the training set and the validation set are shown in Figs. 10 and 11, respectively.

The results in Fig. 11 show that the accuracy of the validation set reaches the highest value to 95.04% when the number of training iterations is 19000 and the accuracy of the training set reaches 100% (see Fig. 10). The confusion matrix of the proposed deep network on the test set is shown in Fig. 12, which shows the scene classification accuracy for each category of the proposed deep network. The results in Fig. 12 show that the highway has the highest accuracy because the features of the highway are most easy to extract. The street has the lowest accuracy because the streets are the most complex scenes in self-driving cars. However, based on the proposed method, the local features and the global features are fused, so the accuracy of the street can reach 93.66%.
C. Comparison Experiments

To show the efficiency of the proposed method, some comparison experiments are conducted, where some other state-of-the-art deep learning-based methods are tested on the same dataset used in Section III-B. These state-of-the-art deep learning-based methods include MobileNet (with 14 Conv, 13 DW, one Pool, one FC, and one Softmax layers) [45], ResNet101 (with 101 layers) [46], AlexNet (with eight layers) [47], EfficientNet (with 16 MBConv, two Conv, one Pool, and one FC layers) [48], and Inception_V1 (with 22 layers; see Fig. 7 for details) [49]. The main reason that these deep learning methods are selected for comparative experiments is that these methods are classic deep learning models in scene classification with good performance. To test these methods under different situations, the dataset is divided into three parts, namely, sunny day, rainy day, and night. The comparison results are listed in Table II. Some scene classification results based on these methods are shown in Fig. 13.

The results in Table II show that the total accuracy of the proposed method can reach 94.76% and increases 4.67% (relative value) compared with the general Inception_V1, which obtains the second-best result in this scene classification experiment. In the experiment, all the state-of-the-art deep learning-based methods can have a relatively high accuracy of the scene classification on a sunny day. From the results in Table II, we can also see that the classification accuracies of all these methods decrease obviously on a rainy day and at night because the feature extraction is difficult for the images on a rainy day and night. However, the proposed deep network can have higher accuracy on a sunny day, the rainy day, and at night because the proposed deep network uses both the local features and global features to realize the scene classification (see the scene classification results in Fig. 13 for details). The standard deviation of the accuracies on different situations based on the proposed method is the smallest of these methods, which also shows that the proposed deep network has good performance in various situations.

IV. DISCUSSION

The total performance of the proposed method has been proven on the special dataset for self-driving cars by some experiments in Section III. In this section, some additional comparison experiments are conducted to discuss the performance of the key parts of the proposed network, including the local feature network based on Faster RCNN and the global feature network based on Inception_V1. Based on these experiments, not only the reasons why these deep networks...
are used in the proposed model but also the ablation analyses (including the attention module in the general Faster RCNN and the mixed activation function with ELU and Leaky ReLU in the Inception_V1) are given out. In addition, the effectiveness of the presented special dataset for the scene classification of self-driving cars and the generalization performance of the proposed model are discussed by the comparison experiments conducted on a public dataset and some real-world traffic videos.

**A. About the Local Feature Extraction Network**

First, the performance of the improved Faster RCNN for local feature extraction is discussed by a comparison experiment with the general Faster RCNN and YOLOv5 (the latest version of the YOLO object detection algorithm [50]). In the proposed deep network, the task of the Faster RCNN is to detect the representative objects of the scene images, including zebra crossings and pedestrians in crosswalks, and fuel tanks of gas stations (see Fig. 3). Thus, before the experiment, each image is manually marked on the dataset. Then, the improved Faster RCNN and the other two networks are used to detect all these representative objects of the input images. The comparison experiment results are listed in Table III, and some experimental results are shown in Fig. 14.

![Fig. 14. Some representative object detection results.](image)

| Network               | mAP    | Standard deviation | On sunny day | On rainy day | At night |
|-----------------------|--------|--------------------|--------------|--------------|----------|
| YOLOv5 [50]           | 72.01% | 7.21%              | 71.79%       | 76.31%       | 62.16%   |
| Faster RCNN [27]      | 90.78% | 2.64%              | 93.59%       | 88.35%       | 90.40%   |
| Improved Faster RCNN  | 91.97% | 1.92%              | 94.17%       | 90.63%       | 91.10%   |

* mAP means the mean average precision [11].
representative objects efficiently in various situations. For example, the improved Faster RCNN can detect pedestrians in the distance with a small scale (see the images of the first line in Fig. 14) and more gas tanks at the night (see the images of the second line in Fig. 14). In the images of the third line in Fig. 14, the improved Faster RCNN can detect both the parking car and lines, and the detection score is higher than the general Faster RCNN. The results in Table III show that the improved Faster RCNN has higher accuracy in all the situations than the other two methods, and the standard deviation of the improved Faster RCNN is less than the general Faster RCNN and YOLOv5, which shows that the stability of the improved Faster RCNN is good.

**B. About the Global Feature Extraction Network**

Another important part of the proposed deep network is the global feature network based on Inception_V1. To show the performance of the improved Inception_V1 in the scene classification of self-driving cars, some comparison experiments are conducted. In these experiments, various combinations of the global feature extraction networks and the improved Faster RCNN-based local feature extraction network are tested on the special dataset. In these combinations, the total structure is the same as the proposed deep network (see Fig. 2), except that the global networks are different. The global feature extraction networks used in these experiments include Inception_V1 [49], Inception_V3 [51], MobileNet [45], ResNet101 [46], and the improved Inception_V1 presented in Section II-B. The comparison experiment results are listed in Table IV.

The results in Table IV show that the combination of the Inception_V1 achieves the second-best performance of all the previous methods except for the improved method in this article. It is the main reason why the Inception_V1 is used for global feature extraction in the proposed deep network. The results of these comparison experiments also show that the improved Inception_V1 has the best performance compared with other methods. Compared with the method using the general Inception_V1 for global feature extraction, our method based on the improved Inception_V1 increases 2.49% in the accuracy of the scene classification (relative value), which means that the improvements of Faster RCNN and Inception_V1 in the proposed method are both important for the scene classification.

To further show the efficiency of the mixed activation function in the Inception_V1 network proposed in this study, some additional comparison experiments are conducted on the validation set, and the result of our method in Section III-B is used as a reference (see Fig. 11). In these experiments, all the settings and structures are the same as the proposed deep network, except that the activation function used in the Inception_V1 network is different. Here, other common activation functions (including Leaky ReLU, ELU, and ReLU) are compared with the mixed activation function used in the improved Inception_V1 network. The accuracies of these experiments based on the proposed deep network with the Inception_V1 using different activation functions are shown in Fig. 15. The results in Fig. 15 show that the proposed mixed activation function can obtain the highest accuracy at a relatively high speed.

**C. About the Special Dataset**

In this study, a special dataset is set up by selecting pictures with clear and distinct representative objects of scene images from some public datasets to further improve the efficiency of the deep learning-based method in self-driving cars. To verify the superiority of the presented dataset, a comparative experiment is carried out with the public dataset BDD100k, which is a driving-related dataset for heterogeneous multitask learning [52]. In addition, to verify the performance of the proposed method in practical application and test the generalization of the proposed deep network model trained in the special dataset, some traffic videos obtained from vehicle data recorder are also used in this experiment. Some images in these traffic videos are shown in Fig. 16. The experimental results are listed in Table V, and the results of Section III-C are used as a reference (see Table II).

The results in Table V show that the accuracies of all the tested deep networks decrease obviously on the public dataset BDD100k. The main reason is that the public dataset contains a large number of scenes unrelated to the classification task of

| Network structure          | Accuracy |
|----------------------------|----------|
| Improved Faster RCNN+Inception_V3 | 55.24%   |
| Improved Faster RCNN+MobileNet | 73.49%   |
| Improved Faster RCNN+ResNet101  | 74.82%   |
| Improved Faster RCNN+Inception_V1 | 92.46%   |
| Ours (Improved Faster RCNN+Improved Inception_V1) | 94.76%   |

Fig. 15. Curve of the accuracy on the validation set based on the proposed deep network with the Inception_V1 using different activation functions.
Fig. 16. Some images in the traffic videos obtained from an automobile data recorder.

| Network          | Total accuracy on dataset BDD100K | Total accuracy on traffic videos |
|------------------|-----------------------------------|---------------------------------|
| MobileNet [45]   | 47.56%                            | 50.98%                          |
| ResNet101 [46]   | 50.15%                            | 49.68%                          |
| AlexNet [47]     | 58.46%                            | 61.12%                          |
| EfficientNet [48]| 72.00%                            | 72.78%                          |
| Inception_V1 [49]| 72.10%                            | 73.14%                          |
| Ours             | 75.99%                            | 83.46%                          |

The proposed deep learning-based method is based on the features of the discriminating objects in the scene, which are almost the same in different scenes for self-driving cars, such as the pedestrians and zebra crossings for crosswalk scenes. In addition, in the proposed method, the global features of the whole image are fused with the local features of the discriminating objects. Thus, the problem of generalization can be solved to some extent. However, lots of works should be done to further improve the generalization ability of the scene classification model in the future.

V. CONCLUSION

The scene classification for self-driving cars based on the deep network is studied, and an improved integrated deep network is presented in this article. In the proposed deep network, the Inception network and Faster RCNN network are used to extract global features and local features, respectively, which are two of the main CNN models for visual computing with excellent performance. In the proposed deep network, these two networks are improved to increase accuracy and computing efficiency. To further improve the efficiency of the deep learning-based method in self-driving cars, a special dataset is set up based on some public datasets. In addition, various comparison experiments are conducted, and the results show that the proposed deep network has a better performance than the state-of-the-art deep networks in the scene classification task for self-driving cars. However, there are some limitations of the proposed method, including the division problem of the scene categories and the classification problem for some heterogeneous scenes, such as the roadside parking lots and gas stations along the street. These problems should be further studied.

In future work, the special dataset for self-driving should be further studied, including the situations with heterogeneous road agents, to make it more suitable for the deep learning-based methods of scene classification in the self-driving car field. On the other hand, how to perform a more nuanced division of the scene categories for self-driving cars is a subject worthy of study. In addition, other deep
network models (such as VGG and AlexNet) will be further studied to check their performance for different tasks in the self-driving car field, including lane recognition and obstacle detection.

REFERENCES

[1] M. Huang, X. Yan, Z. Bui, H. Zhang, and Z. Xu, “Key technologies of intelligent transportation based on image recognition and optimization control,” Int. J. Pattern Recognit. Artif. Intell., vol. 34, no. 10, Sep. 2020, Art. no. 2054024.

[2] A. Mammerei, T. Zuo, and A. Bokkerche, “Extending the detection range of vision-based vehicular instrumentation,” IEEE Trans. Instrum. Meas., vol. 65, no. 4, pp. 856–873, Apr. 2016.

[3] M. Karaduman and H. Eren, “Smart driving in smart city,” in Proc. 5th Int. Istanbul Smart Grid Cities Cong. (ICSGC), Istanbul, Turkey, Apr. 2017, pp. 115–119.

[4] F. Duarte, “Self-driving cars: A city perspective,” Sci. Robot., vol. 4, no. 28, Mar. 2019, Art. no. eaa9843, doi: 10.1126/scirobotics.a9843.8.

[5] N. A. Greenblatt, “Self-driving cars and the law,” IEEE Spectr., vol. 53, no. 2, pp. 46–51, Feb. 2016.

[6] J. Ni, Y. Chen, Y. Chen, J. Zhu, D. Ali, and W. Cao, “A survey on theoretical and applications for self-driving cars based on deep learning methods,” Appl. Sci., vol. 10, no. 8, p. 2749, Apr. 2020.

[7] R. Hussain and S. Zeadally, “Autonomous cars: Research results, issues, and future challenges,” IEEE Commun. Surv. Tuts., vol. 21, no. 2, pp. 1275–1313, 2nd Quart., 2019.

[8] L. Jones, “Driverless when and cars: Where?” Eng. Technol., vol. 12, no. 22, pp. 36–40, 2017.

[9] Q. Xu, M. Wang, Z. Du, and Y. Zhang, “A positioning algorithm of autonomous car based on map-matching and environmental perception,” in Proc. 33rd Chin. Control Conf. (CCC), Nanjing, China, Jul. 2014, pp. 707–712.

[10] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, “Robust lane detection from continuous driving scenes using deep neural networks,” IEEE Trans. Veh. Technol., vol. 69, no. 1, pp. 41–54, Jan. 2020.

[11] Y. Parmar, S. Natarajan, and G. Sobha, “DeepRange: Deep-learning-based object detection and ranging in autonomous driving,” IET Intell. Transp. Syst., vol. 13, no. 8, pp. 1256–1264, Aug. 2019.

[12] J.-R. Xue, J.-W. Fang, and P. Zhang, “A survey of scene understanding by event reasoning in autonomous driving,” Int. J. Autom. Comput., vol. 15, no. 3, pp. 249–266, 2018.

[13] Y. Yang, F. Chen, F. Wu, D. Zeng, Y.-M. Ji, and X.-Y. Jing, “Multi-view semantic learning network for point cloud based 3D object detection,” Neurocomputing, vol. 397, pp. 477–485, Jul. 2020.

[14] H. Xu and G. Srivastava, “Automatic recognition algorithm of traffic signs based on convolution neural network,” Multimedia Tools Appl., vol. 79, nos. 17–18, pp. 11551–11565, May 2020.

[15] Y. Parmar, S. Natarajan, and G. Sobha, “DeepRange: Deep-learning-based object detection and ranging in autonomous driving,” IET Intell. Transp. Syst., vol. 13, no. 8, pp. 1256–1264, Aug. 2019.

[16] J.-R. Xue, J.-W. Fang, and P. Zhang, “A survey of scene understanding by event reasoning in autonomous driving,” Int. J. Autom. Comput., vol. 15, no. 3, pp. 249–266, 2018.

[17] Y. Yang, F. Chen, F. Wu, D. Zeng, Y.-M. Ji, and X.-Y. Jing, “Multi-view semantic learning network for point cloud based 3D object detection,” Neurocomputing, vol. 397, pp. 477–485, Jul. 2020.

[18] H. Xu and G. Srivastava, “Automatic recognition algorithm of traffic signs based on convolution neural network,” Multimedia Tools Appl., vol. 79, nos. 17–18, pp. 11551–11565, May 2020.

[19] Y. Parmar, S. Natarajan, and G. Sobha, “DeepRange: Deep-learning-based object detection and ranging in autonomous driving,” IET Intell. Transp. Syst., vol. 13, no. 8, pp. 1256–1264, Aug. 2019.

[20] J.-R. Xue, J.-W. Fang, and P. Zhang, “A survey of scene understanding by event reasoning in autonomous driving,” Int. J. Autom. Comput., vol. 15, no. 3, pp. 249–266, 2018.

[21] Y. Yang, F. Chen, F. Wu, D. Zeng, Y.-M. Ji, and X.-Y. Jing, “Multi-view semantic learning network for point cloud based 3D object detection,” Neurocomputing, vol. 397, pp. 477–485, Jul. 2020.

[22] H. Xu and G. Srivastava, “Automatic recognition algorithm of traffic signs based on convolution neural network,” Multimedia Tools Appl., vol. 79, nos. 17–18, pp. 11551–11565, May 2020.
H. Alhichri, A. S. Alsowayed, Y. Bazi, N. Ammour, and N. A. Alajlan, “Classification of remote sensing images using EfficientNet-B3 CNN model with attention,” IEEE Access, vol. 9, pp. 14078–14094, 2021.

R. K. Mohapatra, K. Shaswat, and S. Kedia, “Offline handwritten signature verification using CNN inspired by inception V1 architecture,” in Proc. 5th Int. Conf. Image Inf. Process. (ICIIP), Solan, India, Nov. 2019, pp. 263–267.

M. Kasper-Eulaers, N. Hahn, S. Berger, T. Sebulonsen, O. Myrland, and P. E. Kummervold, “Short communication: Detecting heavy goods vehicles in rest areas in winter conditions using YOLOv5,” Algorithms, vol. 14, no. 4, p. 114, Mar. 2021.

Jianjun Ni (Senior Member, IEEE) received the Ph.D. degree from the School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, China, in 2005. He was a Visiting Professor with the Advanced Robotics and Intelligent Systems (ARIS) Laboratory, University of Guelph, Guelph, ON, Canada, from November 2009 to October 2010. He is currently a Professor with the College of Internet of Things Engineering, Hohai University, Changzhou, China. He has published over 100 papers in related international conferences and journals. His research interests include control systems, neural networks, robotics, machine intelligence, and multiagent systems. Dr. Ni also serves as an associate editor and a reviewer for a number of international journals.

Kang Shen received the B.S. degree from Hohai University, Changzhou, China, in 2020, where she is currently pursuing the M.S. degree with the Department of Detection Technology and Automatic Equipment, College of Internet of Things Engineering. Her research interests include self-driving, robot control, and machine learning.

Yinan Chen received the B.S. degree from Hohai University, Changzhou, China, in 2018, where she is currently pursuing the M.S. degree in communication and information systems with the College of Internet of Things Engineering. Her research interests include deep learning and image processing.

Weidong Cao (Member, IEEE) received the Ph.D. degree in mechanical engineering from Chongqing University, Chongqing, China, in 2018. He is currently a Lecturer with the College of Internet of Things Engineering, Hohai University, Changzhou, China. His research interests include swarm intelligence optimization algorithms and data-driven modeling.

Simon X. Yang (Senior Member, IEEE) received the B.Sc. degree in engineering physics from Beijing University, Beijing, China in 1987, the M.Sc. degree in biophysics from the Chinese Academy of Sciences, Beijing, in 1990, the M.Sc. degree in electrical engineering from the University of Houston, Houston, TX, USA, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Alberta, Edmonton, AB, Canada, in 1999. He is currently a Professor and the Head of the Advanced Robotics and Intelligent Systems Laboratory, University of Guelph, Guelph, ON, Canada. His research interests include robotics, intelligent systems, sensors and multisensor fusion, wireless sensor networks, control systems, transportation, and computational neuroscience. Dr. Yang has been very active in professional activities. He was the General Chair of the 2011 IEEE International Conference on Logistics and Automation and the Program Chair of the 2015 IEEE International Conference on Information and Automation. He also serves as the Editor-in-Chief of the International Journal of Robotics and Automation and an Associate Editor for the IEEE TRANSACTIONS ON CYBERNETICS and several other journals.