Real-Time Vehicle Taillight Recognition Based on Siamese Recurrent Neural Network

Ming Liu*, Bingyan Liao, Chenye Wang, Yayun Wang and Yaonong Wang
ZheJiang Dahua Technology CO.,LTD, Hangzhou, China.
* Corresponding Author Email: liu_ming@dahuatech.com

Abstract. Vehicle taillight recognition has been well investigated recent years. However, few methods can be applied in practical because of the time consumption or less recognition accuracy. In this paper, we design a light-weight framework for vehicle taillight recognition in real-time, which is divided into detection and recognition stage. To achieve this purpose, Finer Detection Block and dense anchor regression are utilized to improve the robust of detection stage. In addition, a Siamese CNN-GRU network is proposed, which not only captures the short-time difference but also learns the long-time flash features. We conduct ablation experiments to verify our proposed method, which obtains a mean recall of 94.34% with 103 FPS on 1080Ti.

1. Introduction
Recently years, researchers have developed many successful technologies towards intelligent traffic system. This system aims to analyse vehicles automatically and give a warning report to the dangerous or illegal actions of driving. Nonetheless, the vehicle action recognition is still an open question. In particular, the taillight status is one of the most critical features to define the vehicle actions and remains lots of problems to be solved.

Traditional recognition methods are usually developed based on color thresholds [1][6]. These methods are easily influenced by ambient light. Recently, some deep learning methods are proposed which can be divided into taillight detection [10][11] stage and taillight recognition [15] stage. The recognition stage processes the taillight images which come from detection stage and outputs the final taillight status. Though numerous methods have been proposed to recognize vehicle taillight signals, these methods are hard to be practically deployed because of the time consuming or less recognition accuracy.

In this work, we propose a light-weight two stage method to recognize the taillight statues in real-time with high accuracy. In detection stage, a network based on ResCat is proposed to combine the advantages of ResNet [12] and DenseNet [13]. Besides, the Dense Anchor Regression is proposed to improve the positive samples, which further improves the performance of detection stages. In recognition stage, we design a Siamese CNN-GRU network which receives taillight sequence extracted from continuous frames and output the light status. Siamese network is utilized to extract the difference between two adjacent taillight images, and GRU is used to connect all the difference in a period of time to judge if the lights are continuously flashing. In the inference process, we extract GRU [14] status of each frame in turn and save the status for next frame. The processing time is spread to each frame by this means and the Frames-Per-Second (FPS) can be greatly improved. The experiments show our proposed method can achieve a real-time recognition while keeping a high accuracy.
2. Related Works
There are several taillight recognition methods proposed, which are mainly based on artificial features and deep convolution neural networks features.

Frohlich et al. [2] detect light spots and extract light features using FFT over time in vehicle bounding boxes, and an Adaboost classifier is followed to predict the status of the turn signal. Chen et al. [1] use color thresholds to train AdaBoost classifier for locating taillights, and then utilize the reflectance contrast to tell the directions. These manually engineered features can be obtained in real time, but it’s not robust to the difficult sense such as the different viewpoints of vehicles.

Considering the robust features extracted by deep convolution neural networks, Wang et al. [8] train a Convolutional Neural Network (CNN) model from vehicle rear regions to tell the state of the taillights. Based on mono-camera, Yoneda K et al. [5] use CNN to extract features, and then utilize Fast Fourier Transform to calculate the flashing frequency and recognize the taillight status. In addition, Zhong et al. [9] identify the taillight regions by a FCN network [7], and recognition stage is realized by using a linear support vector machine (SVM) classifier to classify the extracted features within the region. In order to take temporal dependencies into account, Hsu et al. [4] propose a CNN-LSTM structure to learn eight statuses of taillights, where the networks extract taillight features by CNN and recognize the taillights status by LSTM. Based on the CNN-LSTM structure, an attention-based Recurrent Neural Network (RNN) is proposed by K. Lee et al. [3]. They integrate spatial and temporal attention model to emphasize the important region of the images and the key frame of the input sequence respectively. It further improves the robust of temporal features and achieves a state-of-the-art performance. These methods usually have high accuracy but slow inference process because of the time consumption of CNN.

3. Methods
We propose a two stage method to recognize the status of the taillight in real-time. The first stage is the detection stage, and the second is recognition stage. The whole structure is shown in figure 1. The input is the vehicle sequence where every image contains only one car. The detection stage outputs the taillights locations, which are utilized to crop the taillight images and generate the taillight sequence. The recognition stage combines the Siamese CNN and GRU to predict the status of taillight. The details of the two stages are explained in the Section 3.1 and Section 3.2.

![Figure 1. The whole structure of our proposed method.](image)

3.1. Taillight Detection
The robust taillight detection results can give a significant contribution to the final recognition results. To improve the recognition performance, it is important to recall the taillight as much as possible while reduce the false negatives.

Finer detection block. We design a light-weight network to make full use of the information in low and high level feature, the single block is called ResCat. This block combines the connecting formats of ResNet and DenseNet. The structure comparison with other blocks is shown in figure 2. figure 2 (a) uses shortcut connecting to make it easier to optimize the residual mapping. figure 2(b) shows the DenseNet block which utilizes the concatenate connecting to directly connect any layer to all subsequent layers and further improve the information flow between layers. These dense connections can both contribute to the feature extracting process in low and high level. To combine the advantages
of ResNet and DenseNet, the ResCat block is proposed with the architecture of figure 2 (c). The features are first dense connected based on DenseNet block. After the features are downsampled to the same by an $1 \times 1$ convolution, the residual learning can be introduced in. Connecting features of different stages by ResCat block encourages feature reuse throughout the network, and leads to residual learning. The experiments show that ResCat block can further improve the feature extraction ability and contributes to the detection performance.

![ResNet block](image1.png) ![DenseNet block](image2.png) ![ResCat block](image3.png)

**Figure 2.** The architectures in different frameworks. $\odot$ is the concatenate operation and $\oplus$ is the shortcut connecting.

Dense anchor regression. The detection frame is based on yolov3 [17]. Different sizes of feature maps are utilized to detect objects. The anchor regression strategy is that one grid cell only regresses one best anchor. This strategy does work when objects are normally distributed in the image, while it may not work well to our taillights detection task because of the special distributions. Taillights are sparse distribution in a car image because of the separated location and the certain number. What’s more, as is shown in figure 3, the normalized minimum side of every ground truth bounding box indicate that the taillight region is relatively small compared with the whole car.

![Figure 3](image4.png)

**Figure 3.** The statistical bounding box distribution of training dataset. The side is normalized to $[0, 1]$ and ratio comes from the width divided by height.

This sparse distribution of small object leads to few positive anchors to regress in the training stage because of the anchor regression strategy of yolov3. Based on the analysis above, we propose a dense regression method to improve the positive samples which outperforms the original regression strategy. In training process, all the anchors with the IoU $> \alpha$ are regressed, where the IoU is the Intersection over Union between proposed anchor and ground truth. By using the dense regress strategy, the number of sample objects is about 3 times larger than original methods when $\alpha = 0.2$.

**3.2. Taillight Recognition**

Construction of the taillight sequence. Usually, one vehicle has just one left lamp and one right lamp. However, we cannot guarantee that the detection results always satisfy this condition. When multiple lamps are detected, we need to pick the correct one from the wrong results. The whole process is described in figure 4. The wrong detected lamp boxes can be divided into three classes: (1) Single taillight of other vehicles; (2) Not a taillight; (3) Taillight pair of other vehicles. Considering that the taillight are always paired and have similar vertical coordinate, we pair the left and right lamp firstly to filter the boxes described in (1) and (2). Specifically, we build collections for left and right taillights separately. For each operation, we traverse every taillight in the two collections to searching left lamp
and right lamp which have the closest vertical coordinate. These two lamps compose a taillight pair. In order to ensure the accuracy of matching, the taillights are subject to

$$\begin{align*}
&|y_l - y_r| < \tau \\
&x_l < x_r
\end{align*}$$

(1)

Where \((x_l, y_l)\) and \((x_r, y_r)\) are the center coordinate of left and right taillight, \(\tau\) is the threshold set by experiments. After that, removing these two lamps from respective collection. Repeating above process until one collection becomes empty or no lamp pair satisfies the condition. If there is no selected lamp pair in the last, we directly use the pair with the closest vertical coordinate as final result.

According to the shooting angle by camera, the most likely case for multiple pairs is that the taillight of the front vehicle falls into the bounding box of the vehicle behind as the vehicle shown in figure 4. So, the taillight pair closest to the bottom of the image is selected as the final result. For another case all lights detected have the same position property due to obstruction and wrong detection, we also choose the taillight closest to the bottom of the image as final result.

After determining the bounding box of the vehicle's taillights in each frame, the taillights in continuous frames are cropped and form the taillight sequences for turn signal recognition. In order to maintain the integrity of the lights, the cropped area will expand \(n\) times based on the detected taillight boxes.

Figure 4. The whole process of taillight filtering. Figure 5. The structure of the proposed recognition network.

Turn signal recognition. Turn lamp and emergency light are timing signals, what we focus on should be the change in brightness of taillight rather than the light itself. They contain both short term difference and long term regularity. For the former, the brightness of the tail light in adjacent frames is obvious different. However, it is not enough to distinguish turn light and emergency light from others lights, such as the moment the brake light on. Over a longer period of time, this difference should be appeared periodically. In this paper, we construct a RNN to recognition the signal which could capture the flash between adjacent frames and long term regularity simultaneously. As is shown in figure 5, the proposed network is composed with Siamese network, CNN and GRU three parts.

The input \(F\) is the chunks of the continuous cropped tail light images \(f_1, f_2, ..., f_i, ..., f_n\), which are detected in first stage and can be formulated as:

$$F = \{f_1, f_2, ..., f_i, ..., f_n\}$$

(2)

In every time-step, the adjacent two taillight images \(f_i\) and \(f_{i+1}\) are fed into Siamese network to extract deep features. Siamese network is composed with two fully-convolutional networks which share the same architecture and weights. It has shown excellent performance in calculation similarity of two inputs, such as image matching task, tracking task and person re-identification task. Immediately following the Siamese network is a CNN, which receives the concatenated feature of taillight image \(f_i\) and \(f_{i+1}\) and filters most of the unused information for only preserving the changes. The two networks work together to capture the changes of taillight.

Long term regular flash is the most important characteristic of taillight signal. Here, we construct a GRU network to judge if the flash regularity exists in the input images. GRU is one of the variant of LSTM which has only update gate and reset gate, while LSTM has three gates. Due to fewer parameters, GRU becomes easier to train. At the last of the network, two fully connection layers are
utilized to output the taillight status. According to the status of left and right taillight, the following four taillight signals can be recognized as table 1.

| Left status | Right status | signal          |
|-------------|--------------|-----------------|
| flash       | flash        | Emergency       |
| Not flash   | Not flash    | Turn left       |
| Not flash   | Flash        | Turn right      |
| Not flash   | Not flash    | Off             |

**Table 1.** Taillight signals according to the taillight status.

4. Experiment

In this section, we first introduce the dataset used to train and test. Then, the training details and hyper parameters setting of the proposed method are explained. After that, we evaluate our network on test dataset and conduct the ablation experiments to verify the improvements.

4.1. Datasets Construction

Since there is no public dataset, we collect traffic videos from multiple scenes and crop vehicle images from them. To avoid data redundancy, we extract vehicle images from videos every 25 frames and filter the repeat vehicles. There are a total of 100000 vehicle images for detection training and 10000 images for detection test. All vehicle images are marked with locations of taillights include ‘left taillight’ and ‘right taillight’. Meanwhile, we collect 3000 vehicle sequences for recognition training and 500 sequences for test. Each taillight has a ground truth bounding box and the sequence is marked with the properties of ‘turn right’, ‘turn left’, ‘off’ and ‘emergency’.

4.2. Implement Details

In detection stage, the input size is set to 112 × 112. The input images are first processed by some regular data arguments, such as random blur, random color jitter and random scale jitter, etc. The training loss is defined as [11], while the regression loss is replaced by CIoU loss [16]. We train the proposed network with 501000 iterations with batch size of 256. For the first 1000 iterations, we use warm-up learning rate of 0.001 to 0.005. And the learning rate is decayed from 1 × 10^{-3} to 1 × 10^{-5} gradually. In recognition stage, the input images have size of 64 × 64 and the pixel value is normalized to [-1, 1]. We flip all the right turn signals horizontally to simulate the left turn signal to reduce the difficulty of network fitting. In train stage, loss function is Cross Entropy loss function. We train the network total 5000 iterations with learning rate decay from 0.01 to 0.001. All our methods are implemented on 1080Ti.

4.3. Experiment Results

To investigate the impact of different proposed methods, we conduct several ablation experiments to prove the improvements.

Detection results. In detection stages, the methods of Finer Detection Block and Dense anchor regression are taken into account. The detection results are shown in table 2. All the results are detected with possibility thresh of 0.5 and IoU thresh lager than 0.4. The basic ResNet-w36 model is the modified ResNet with base channel of 36, which has total three ResNet blocks. Our ResCat block based network ResCat-w36 has the same base channels with ResNet-w36, the difference is that the ResNet block is replaced by ResCat block.

**Table 2.** The ablation experiments about detection methods. The FPS of different methods are shown in the last column.

| methods       | IoU  | Precision | Recall | mAP@0.5 | FPS  |
|---------------|------|-----------|--------|---------|------|
| ResNet-w36    | 0.7633 | **0.9785** | 0.9310 | 0.6271  | **133.2** |
| ResCat-w36    | 0.7685 | 0.9781    | 0.9359 | 0.6276  | 127.3 |
| ResCat-w36+ DAR(α = 0.2) | 0.7963 | 0.9697 | 0.9499 | 0.6322 | 126.9 |
| ResCat-w36+ DAR(α = 0.01) | **0.8065** | 0.9698 | **0.9658** | **0.6363** | 126.7 |
The baseline network ResNet-w36 resulted in a Recall of 93.10%. Using the ResCat block explained in section 0, Recall improved by 0.49%. By combining the DAR strategy, our detection network further improves 1.40%. After reducing the value of α, we can get the best detection Recall of 96.58%. This comparison results indicate that our proposed methods are fast and robust enough to locate the taillight in real time.

Recognition results. In recognition stage, we test the influence of Siamese network architecture, LSTM, and GRU. The accuracy results can be found in table 3. We observe that the network trained with GRU obtains the better accuracy than LSTM based model. This improvement comes from fewer parameters. After introducing the short term information by using the Siamese Network, the accuracy further improves by 0.28%.

| Method            | Accuracy(%) |
|-------------------|-------------|
| No-Siam-CNN-LSTM  | 98.14       |
| No-Siam-CNN-GRU   | 98.43       |
| Siam-CNN-LSTM     | 98.27       |
| Siam-CNN-GRU      | **98.71**   |

We further test our whole recognition framework on sequence test dataset and show the results in table 4. The metric of Recall, Precision and F1 score are compared. Our proposed algorithm is able to tell the taillight signals effectively. The recall of Emergency is relatively low because of the abnormal situations such as the broken tailights. Besides, our detection model occupies 1.4M memory space and the recognition only occupies 495KB. The whole recognition framework can run with 103 FPS which indicates that it is a real-time method and easy to deploy on hardware platform. The result samples are shown in figure 6.

|                | Recall(%) | Precision(%) | F1 score |
|----------------|-----------|--------------|----------|
| Turn left      | 96.31     | 94.15        | 0.9521   |
| Turn right     | 96.50     | 94.40        | 0.9543   |
| Emergency      | 90.47     | 96.51        | 0.9339   |
| Off            | 94.09     | 96.55        | 0.9530   |
| Mean           | 94.34     | 95.40        | 0.9483   |

Figure 6. The samples of recognition results on test sequences. The flash taillight is shown in top right corner and the frame number in videos is shown in the top left corner.
5. Conclude
In this paper, we present a two stage recognition methods to achieve real-time and high accuracy taillight recognition. In detection stage, we first design a novel backbone block to improve the feature extracting ability for taillight detection. Besides, the dense regression method is also proposed and the improvement is verified in ablation experiments. In recognition stage, we propose Siamese CNN-GRU network to significantly improve accuracy by capturing the short-time difference and long-time periodicity simultaneously. The final recognition results show that our approach can achieve an outstanding performance.

6. References
[1] Chen D Y, Peng Y J, Chen L C, et al. Nighttime turn signal detection by scatter modeling and reflectance-based direction recognition[J]. IEEE Sensors Journal, 2014, 14(7): 2317-2326.
[2] Fröhlich B, Enzweiler M, Franke U. Will this car change the lane?-turn signal recognition in the frequency domain[C]/2014 IEEE Intelligent Vehicles Symposium Proceedings. IEEE, 2014: 37-42.
[3] Lee K H, Tagawa T, Pan J E M, et al. An Attention-based Recurrent Convolutional Network for Vehicle Taillight Recognition[C]//2019 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2019: 2365-2370.
[4] Hsu H K, Tsai Y H, Mei X, et al. Learning to tell brake and turn signals in videos using cnn-lstm structure[C]//2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2017: 1-6.
[5] Yoneda K, Kuramoto A, Suganuma N. Convolutional neural network based vehicle turn signal recognition[C]//2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS). IEEE, 2017: 204-205.
[6] Li Y, Cai Z, Tang J. Recognition algorithm for turn light of front vehicle[J]. Journal of Central South University, 2012, 19(2): 522-526.
[7] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431-3440.
[8] Wang J G, Zhou L, Pan Y, et al. Appearance-based brake-lights recognition using deep learning and vehicle detection[C]//2016 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2016: 815-820.
[9] Zhong G, Tsai Y H, Chen Y T, et al. Learning to tell brake lights with convolutional features[C]//2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2016: 1558-1563.
[10] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]//Advances in neural information processing systems. 2015: 91-99.
[11] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
[12] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
[13] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.
[14] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.
[15] Hochreiter S , Schmidhuber J. Long Short-Term Memory[J]. Neural computation, 1997, 9(8):1735-1780.
[16] Zheng Z, Wang P, Liu W, et al. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression[C]//AAAI. 2020: 12993-13000.
[17] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.