A robust traffic scene recognition algorithm based on deep learning and Markov localization

Guoan Yang, Zirui Zhao, Zhengzhi Lu, Junjie Yang, Deyang Liu, Yong Yang, Chuanbo Zhou
School of Automation Science and Engineering
Xi'an Jiaotong University
Xi'an, Shaanxi 710049, China
e-mail: gayang@mail.xjtu.edu.cn

***Abstract*—This paper designs a traffic scene recognition module for the agent's perception system. First, we enabled the output features of the convolutional neural network to be the descriptor of the traffic scene and adapted to the cost function of the image sequence to construct the observation module of the agent. Second, we assumed that the movement of the agent would be recursively updated and wouldn't jump dramatically, which simultaneously possesses the Markov property, so the Markov localization algorithm was used to improve overall robustness. Third, the Kalman filter method was adopted to represent the probability distribution of the entire system using the first and second moments of the Gaussian distribution, so that the loop iteration in the state estimation can be transformed into a linear operation, and the penalty term in the standard variance of the observation probability can also be added to describe the reliability of the observation. Experimental results show that the agent can efficiently remove unreliable observations and achieve robust recognition accuracy of the traffic scene in all weather conditions.**

Keywords-traffic scene recognition; deep learning; convolutional neural network; Markov localization; Kalman filter

I. Introduction

Currently, we still rely on GPS + Lidar to realize the real-time positioning of an unmanned vehicle. However, the cost of high-precision radar is too expensive, and the amount of information obtained is not as sufficient as that of human vision. Furthermore, people rely on vision alone to obtain robust information about traffic scene for navigation of driving cars.

We know that the traffic scene recognition problem is how to overcome the influence of changes in appearance, such as illumination and seasonal changes. The problem is how to enable the unmanned auto agent to obtain basic information that is not interfered by illumination and appearance factors. Besides, it is difficult to find the visual feature descriptors with high robustness to appearance. Fortunately, in recent years, such a problem can be solved from another way of thinking: deep learning. A convolutional neural network (CNN) has been the most representative method in deep learning. This method, also known as end-to-end learning, is realized by a CNN with a large number of convolution layers. Therefore, we can sue a CNN model to automatically find the above-mentioned essential information in the robust recognition of traffic scenes.

In addition, unmanned vehicles as agents generally work in a relatively large and unfamiliar environment and this scene will vary as time passes. Therefore, the purpose of this paper is to solve the problem of dynamic traffic scene recognition under consideration of contextual relations, including (a) how to capture essential features such as dynamic scene information that does not change with appearance in the traffic scene, thereby constructing the scene description of the traffic environment; (b) how to use the above information to express the correlation between dynamic scenes; and (c) how to convert the features of the above basic information into accurate confidence after overcoming the impact of scene changes on the recognition process.

Finally, this paper proposes a new method to realize the robust recognition of dynamic traffic scene by combining both AlexNet model and Markov localization algorithm. Furthermore, we carried out experiments on open data sets to verify the recognition accuracy and robustness of our approach.

II. The Proposed Method

A. A CNN Representation for the Traffic Scene

The traffic scene is represented by a CNN model. Sunderhauf et al. [1] and Dai et al. [2] show that the CNN features trained by the scene classification data set have strong robustness for dynamic recognition of the traffic scene. Therefore, in this paper, we train AlexNet [4] using the scene-directed data set Places365 [3], and we use the output of the fully connected layer as the image description. In addition, Garg et al. [5] adopted a normalization method of feature descriptors, which significantly improved the robustness of fully-connected layers to changes in appearance, which can be expressed as:

$$\vec{f}_i^{\,'} = \frac{\vec{f}_i - \vec{\mu}_i}{\vec{\sigma}_i} \quad \forall i \tag{2.1}$$

where $\vec{f}_i$ represents the output of feature descriptors in the fully-connected layer FC6, $\vec{\mu}_i$ and $\vec{\sigma}_i$ represents the mean

978-1-7281-8822-5/20/$31.00 ©2020 IEEE

and variance of the image descriptor in the entire map database, respectively, while $\vec{f}_i^{'}$ denotes the normalized set of descriptor (NSD). Here, we use the NSD to represent essential information of the traffic scene.

B. Coherent Matching Costs for Scene Sequences

Here, we use the framework of SeqSLAM [6] to achieve the matching between scene image sequences. First, we need to construct the cost matrix as follows:

$$D=\begin{array}{c} \overbrace{\begin{array}{cccccc} I_1 & I_2 & \cdots & I_j & \cdots & I_n \end{array}}^{\mathcal{Q},\ a\,view\ from\,Robots} \\ \begin{bmatrix} D_{1,1} & D_{1,2} & \cdots & D_{1,j} & \cdots & D_{1,n} \\ D_{2,1} & D_{2,2} & \cdots & D_{2,j} & \cdots & D_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ D_{i,1} & D_{i,2} & \cdots & D_{i,j} & \cdots & D_{i,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ D_{m,1} & D_{m,2} & \cdots & D_{m,j} & \cdots & D_{m,n} \end{bmatrix} \end{array} \left. \begin{array}{c} \\ I_1 \\ I_2 \\ \vdots \\ I_i \\ \vdots \\ I_m \end{array} \right\} \mathcal{M},\ a\,view\ from\,Maps \qquad (2.2)$$

$$D_{i,j}=1-\cos(\vec{f}_i', \vec{f}_j')=\frac{\|\vec{f}_i'\|\ \|\vec{f}_j'\|-\vec{f}_i'\cdot\vec{f}_j'}{\|\vec{f}_i'\|\ \|\vec{f}_j'\|} \qquad (2.3)$$

where $\vec{f}_i^{'}$ is the image I_i∈M , $\vec{f}_j^{'}$ is the image I_j∈Q , and they are the normalized descriptors of the coherent images in the FC6 layer.

Here, we perform the matching within sequences by summing the cosine distances, as shown in Eqs. 2.4 -2.6:

$$S_{T,j}=\min_{v\in\mathcal{V}}\sum_{t=T-d_s}^{T} D_{k,t},\quad k=j+v(t-T), \qquad (2.4)$$

$$v_T^{\text{esti}}=\arg\min_{v\in\mathcal{V}}\sum_{t=T-d_s}^{T} D_{k,t},\quad k=j+v(t-T). \qquad (2.5)$$

$$z_T=\underset{j=1,2,\text{L}\ ,m}{\operatorname{argmin}} S_{T,j} \qquad (2.6)$$

where $S_{T,j}$ represents the matching costs of the scene sequences, $D_{k,t}$ represents the cosine distances between the image obtained by the agent at time t and the image K on the map, T denotes the present time, j denotes the j frame selected from maps, v denotes the matching slope. Furthermore, we restrict the range of velocity v to prevent the occurrence of outliers. Then, by regulating the velocity values v, we can obtain the best matching $S_{T,j}$ and and v_T^{esti}, where v_T^{esti} is the estimation value of the agent's current velocity, as expressed in Eq. (2.5). After that, the position of the agent can be predicted at the next time T+1 based on the previous estimate. Therefore, the best matching result $z_T=\underset{j=1,2,\text{L}\ ,m}{\operatorname{argmin}} S_{T,j}$ is the output of the observation at time T. The matching process is shown in Fig. 1.

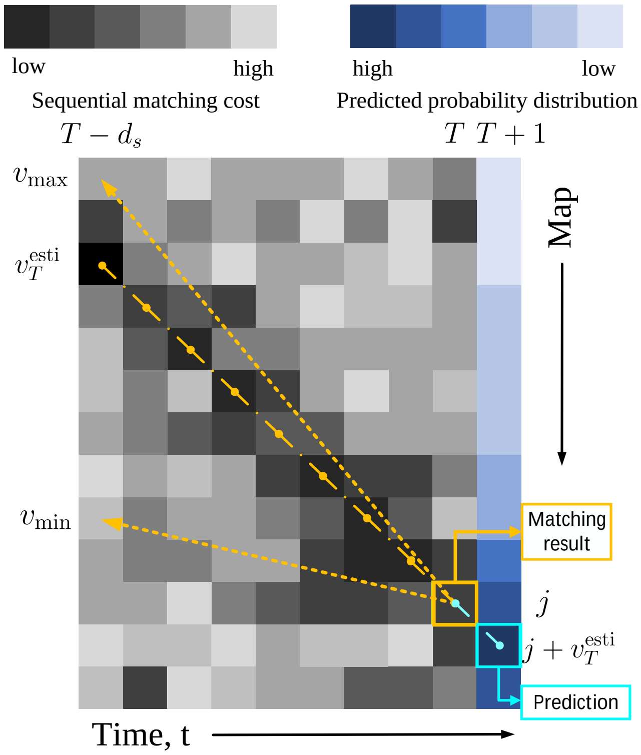


Figure 1. Schematic diagram of sequence matching algorithm

C. Measurement of Probability Distribution Estimation

After obtaining the results of sequence matching, this section will discuss how to convert the above values into a probability distribution, that is, a likelihood distribution $\tilde{p}=(z_T=j\,|\,x_T,M)$, so that we can use the probability-based Markov localization method. Thus, the Softmax function is used to map these values to the probability in the (0,1) range, and also adopts a sliding window with a length of N, W=[z_T-N/2,z_T+N/2] to calculate the probability distribution in N candidate frames close to the optimal matching result, as follows:

$$\tilde{p}(z_T=j\,|\,x_T,\mathcal{M})=\frac{e^{-S_{T,j}}}{\sum_{k\in\mathcal{W}} e^{-S_{T,k}}} \qquad (2.7)$$

After the matching cost value t is turned into probability, we further assume that the probability conforms to a Gaussian distribution. In this paper, we use the least squares method to estimate the Gaussian Distribution.

D. Markov Localization under Gaussian Distribution

Markov localization under Gaussian distribution is actually a Kalman filter.

In the process of navigation and positioning, people cannot only estimate the current state through the current observation, but also infer with their own memory. Hence, we use a hidden Markov chain to represent the above-mentioned process, as shown in Figure 2.

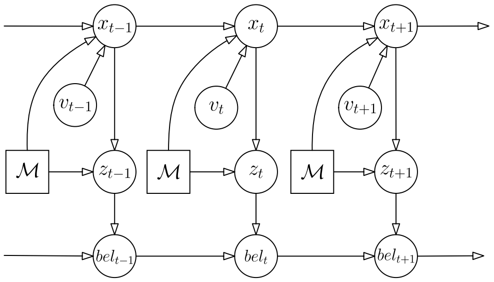


Figure 2. Markov chain diagram

As shown in the Markov localization algorithm in Algorithm 2.1, we have to calculate the confidence level of each state in the entire state space for each state update in the algorithm, so the complexity of the algorithm is O(n).

ALGORITHM 2.1 MARKOV LOCALIZATION ALGORITHM

```
Input: $bel(x_{t-1})$, $u_t$, $z_i$, $\mathcal{M}$
Output: $bel(x_t)$
for all $x_t$ do
    $\overline{bel}(x_t) \leftarrow \int p(x_t \mid u_t, x_{t-1}, \mathcal{M}) bel(x_{t-1}) dx_{t-1}$
    $bel(x_t) \leftarrow \eta\, p(z_t \mid x_t, m) \overline{bel}(x_t)$
end for
```

In this paper, the first and second moments of the random variables are used to parameterize the probability distribution. Here, we use the Gaussian distribution to describe the above-mentioned probability. Therefore, the Markov localization algorithm can be transformed into the Kalman filter localization algorithm, as shown in algorithm 2.2.

ALGORITHM 2.2 KALMAN FILTER LOCALIZATION ALGORITHM

```
Input: $\mu_{i-1}$, $\sigma_{i-1}$, $u_i$, $z_i$
Output: $\mu_i$, $\sigma_i$
$\bar{\mu}_i \leftarrow \mu_{i-1} + v_i^{\text{esti}}$
$\bar{\sigma}_i \leftarrow \sigma_{i-1} + \sigma_{v_i^{\text{esti}}}$
$K_i \leftarrow \bar{\sigma}_i (\bar{\sigma}_i + \sigma_o)^{-1}$
$\mu_i \leftarrow \bar{\mu}_i + K_i (z_i - \bar{\mu}_i)$
$\sigma_i \leftarrow (1 - K_i) \bar{\sigma}_i$
```

Here, we use the square of the difference between the predicted result and the observed result as a penalty term to evaluate whether the observation is reliable, thereby preventing the jump in the observed value from affecting the final recognition result, as shown in Eq. 2.8.

$$\sigma_o = \sigma_{\text{fit}} + \underbrace{\theta(\mu_o - \bar{\mu}_i)^2}_{\text{penalty factor}} \qquad (2.8)$$

E. Algorithm Flowchart

First of all, the algorithm flowchart of the present study is shown in Figure 3.

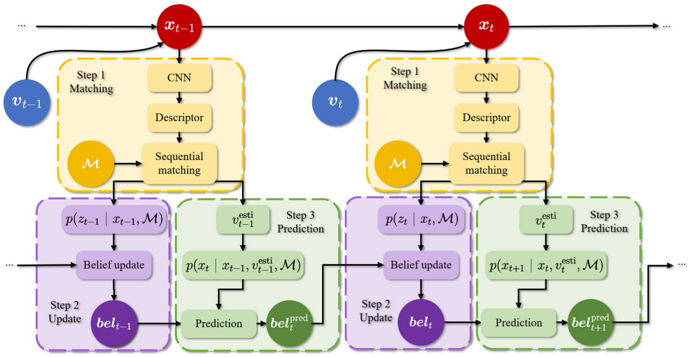


Figure 3. The algorithm flowchart

In the first step, CNN AlexNet is used to obtain the descriptor of the current image, calculate the values of coherent matching costs $S_{T,j}$, estimate the current velocity value v_T^{esti}, and obtain the observed likelihood probability distribution $\tilde{p}(z_i \mid x_i, M)$; the second step is to update the current confidence level $bel(x_i)$ based on the likelihood probability distribution $\tilde{p}(z_i \mid x_i, M)$ and the previously predicted confidence level $\overline{bel}(x_i)$; the third step is to obtain the probability distribution of the state transition $\tilde{p}(x_{i+1} \mid x_i, v_i^{esti}, M)$ by using the current velocity v_i^{esti}, and using $\tilde{p}(x_{i+1} \mid x_i, v_i^{esti}, M)$ and the previously confidence level $bel(x_i)$ to predict the confidence level of the next state $\overline{bel}(x_{i+1})$.

III. EXPERIMENT AND ANALYSIS

A. Experimental Data Set

In this paper, we use the RobotCar data set of Oxford University [5], which has collected data for nearly a year on a fixed route in Oxford, as shown in Figure 4 and Figure 5, thereby being very suitable for the robust traffic scene recognition in this paper.

This experiment is based on the work of SeqSLAM [6] and NSD-SeqSLAM [7] as a benchmark for comparative experiments. This experiment takes the data from cloudy days in spring as the benchmark, and takes cloudy days in autumn, sunny days in autumn, rainy days in autumn, cloudy days in winter and sunny days in spring as experimental data, as shown in Figure 5. We use GPS navigation information as the real value of the quantitative analysis of recognition results.

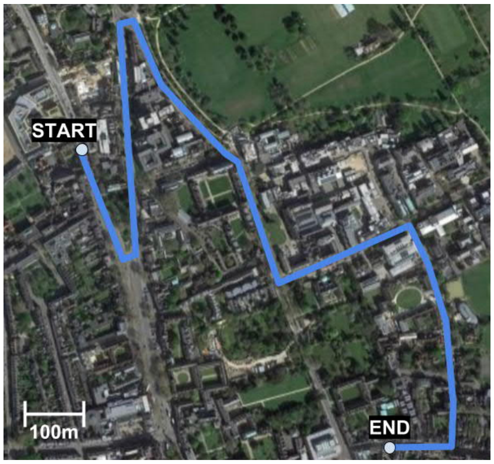
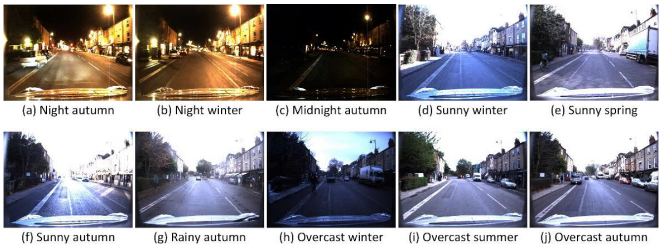


Figure 4. Path of experimental data set



(a) Night autumn (b) Night winter (c) Midnight autumn (d) Sunny winter (e) Sunny spring

(f) Sunny autumn (g) Rainy autumn (h) Overcast winter (i) Overcast summer (j) Overcast autumn

Figure 5. Some samples in the RobotCar dataset

B. Experimental Analysis Index

The first index of the performance evaluation of our approach is the precision-recall curve (PR curve). The correct match is called "true positive (TP)", the incorrect match is called "false positive (FP)", and the match discarded by the algorithm is "false negative (FN)". Precision is defined as the proportion of the selected match being a true positive. In addition, the recall rate is the ratio of the true positive to the total number of correct values.

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \qquad (3.1)$$

We use variance as a threshold to construct the PR curve. Another important index is the area under the PR curve (AUC curve) with threshold radius, which is calculated as follows:

$$\text{AUC} = \sum_{i=1}^{N-1} \frac{p_i + p_{i+1}}{2} \times (r_{i+1} - r_i) \qquad (3.2)$$

where N is the number of matched images, p_i is the precision at point i, and r_i is the recall rate at point i.

C. Experimental Results and Analysis

The matching results based on our experiments are shown in Figure 6. It can be seen that the matching results are more stable and form a smooth curve through Kalman filtering, and the variance of the Gaussian distribution is also effectively suppressed.

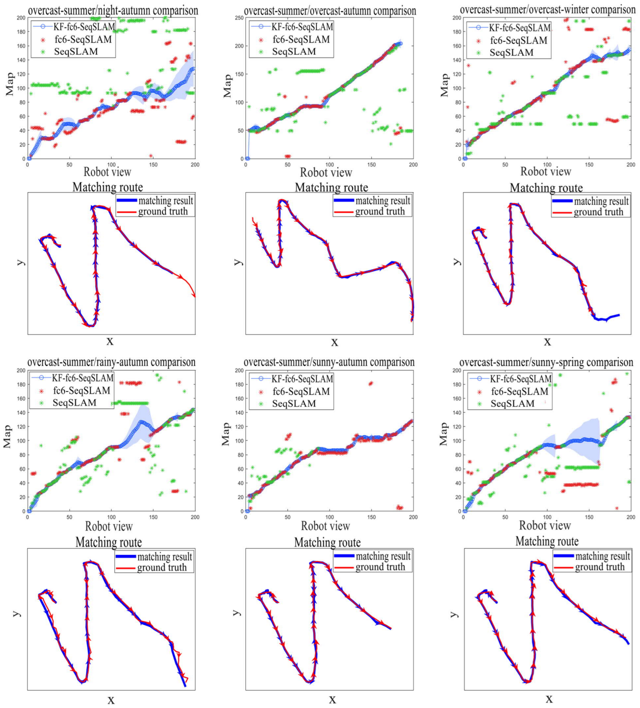


Figure 6. Schematic diagram of matching results, where the blue is from our method, red is from SeqSLAM using NSD-FC6 features, and the green is from the original SeqSLAM. It can be seen that using our method can filter out unstable matching results and make the matching results more stable.

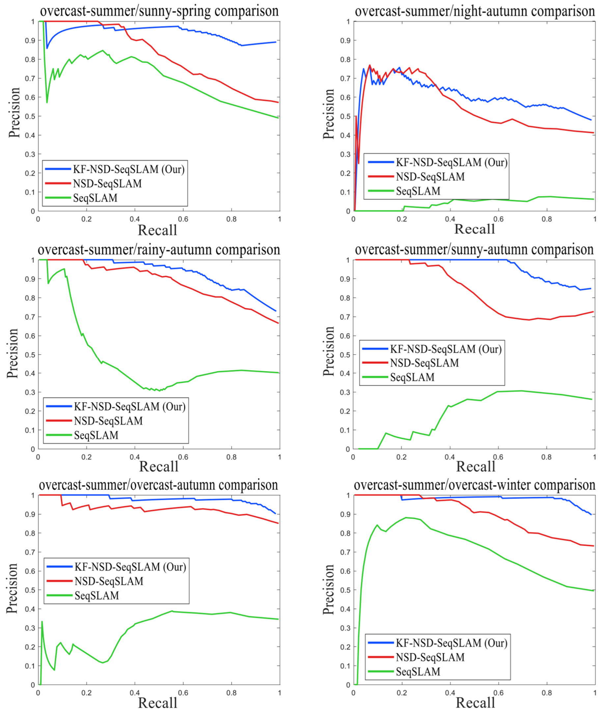


Figure 7. The matching results of the PR curve. Blue is from our method, red is from SeqSLAM with NSD-FC6 features, and green is from the original SeqSLAM. By using our method, we can see that the matching results are more stable and most of the curves are at the top compared to the other method.

Through the analysis of the matching results, the PR curve can be obtained, as shown in Figure 7.

From Fig. 7, we can see that the overall matching results are improved, the stability is more obvious, and the precision is significantly improved under a certain recall rate.

Finally, the AUC curve varies with the threshold of the location matching radius, as shown in Figure 8.

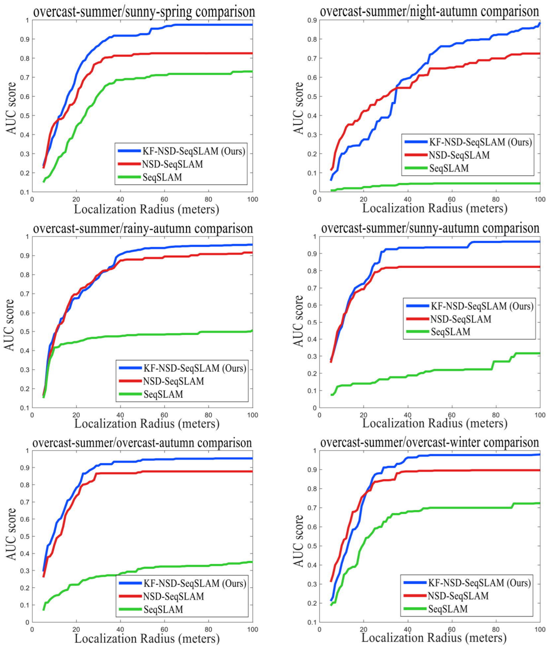


Figure 8. Matching results of the AUC curve. Blue is from our method, red is from SeqSLAM with NSD-FC6 features, and green is from original SeqSLAM. It can be seen that after using the Kalman filtering, the matching result is more stable and the overall matching effects are improved. However, when the matching radius is 5 - 25 m, the improvement effect is not obvious, and in some cases, the matching performances will drop.

From the above experimental results, it can be seen that the overall matching performances were improved after adding Kalman filtering, and there are better results within the threshold radius of 30 - 100 m. However, the localization matching within the radius threshold of 5 – 25 m is still insufficient.

IV. Conclusion

In this paper, based on the SeqSLAM recognition method of the dynamic traffic scene, the normalized descriptors extracted from the AlexNet model were adopted, and the Markov localization algorithm was also added under a Gaussian distribution. Moreover, a more robust scene recognition and localization approach under all-weather conditions was constructed without increasing the overall complexity of the algorithm. The experimental results show that the proposed method possesses a better recognition performance compared to SeqSLAM.

V. Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 61673314, 61573272, and in part by the National Key R&D Program Project of China under Grant 2018YFB1700104.

References

[1] Lowry S, Sünderhauf N, Newman P, et al. Visual place recognition: A survey[J/OL]. IEEE Transactions on Robotics, 2016, 32(1): 1-19. DOI: 10.1109/TRO.2015.2496823.

[2] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in Neural Information Processing Systems, 2012: 1097-1105.

[3] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[4] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1-9.

[5] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning[J]. AAAI Conference on Artificial Intelligence, 2016.

[6] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2818-2826.

[7] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.