Attention Aware Deep Learning Object Detection and Simulation

Jiping Xiong(✉), Lingyun Zhu, Lingfeng Ye, and Jinhong Li

College of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua 321300, China
xjping@zjnu.cn

Abstract. Dish recognition has certain difficulties in specific applications. Because in the actual inspection, the dishes are filled with food, and the food occupy most of the space of the dishes, and only the edges of the dishes can be seen. If you use empty dishes for training, the accuracy will be low due to insufficient feature matching during actual detection. At the same time, due to the wide variety of foods, if we collect all the food during training, the pre-processing workload will be very large. Based on the above ideas, this paper analyzes the model through three visualization methods, improves Faster R-CNN, and proposes a Cross Faster R-CNN model. This model consists of Faster R-CNN and Cross Layer, which can fuse the low-level features and high-level features of dishes. During training, the model can focus the feature extraction on the edges of the dishes, reducing the interference of food on dish recognition. This method improves the detection accuracy without significantly increasing the detection time. The experimental results show that compared with Faster R-CNN, the accuracy and recall of Cross Faster R-CNN have increased to a certain extent, and the detection speed has basically not changed significantly.

Keywords: Deep learning · Object detection · Convolutional neural network · Visualization

1 Introduction

In recent years, artificial intelligence has developed rapidly. More and more fields combined with artificial intelligence have begun to improve people's daily live. With the growth of the economy, people's requirements for living standards have gradually increased. In view of the need to solve the problems of the traditional model in the catering industry, controlling costs is one of the most important factors. Among them, labor costs account for the majority. Especially at this stage, restaurants generally use manual billing, which is not only a waste of human resources. At the same time, queueing billing is a waste of customer time. Therefore, if machines can be used for manual billing, it can reduce costs on the one hand and save time and increase passenger traffic on the other.

Artificial intelligence developed slowly before 2012, but after the advent of machine learning, there was a dawn, and the most eye-catching aspect of machine learning is deep

H. Song and D. Jiang (Eds.): SIMUtools 2020, LNICST 370, pp. 1–14, 2021.
https://doi.org/10.1007/978-3-030-72795-6_1

learning. Deep learning has broken through many areas that are difficult for artificial intelligence machine learning to break through, bringing the possibility of the further development of artificial intelligence [1, 2]. AlphaGo is a typical artificial intelligence that uses deep learning. Deep learning's effects achieved in the fields of computer vision and natural language processing have far surpassed traditional methods [3, 4]. Deep learning is a learning method that approximates humans. Through a large amount of sample data, the machine learns the inherent laws and representation levels, and finally has the ability to analyze like humans. Convolutional neural network is a successful application of deep learning in the field of image vision. It has made great breakthroughs in tasks such as image classification [5–7], object detection [8–11], semantic segmentation [12–16], and instance segmentation [17]. Since *Hinton* [18] won the championship with *AlexNet* in the ImageNet competition in 2012, convolutional neural networks have achieved great development and have been applied in daily life, such as image subtitles [19, 20, 33], visual question and answer[21, 22, 34], and autonomous driving[23, 24, 35] which are recent some research focused on, and so on.

The current research on automatic billing has two directions, one is the billing for food identification, and the other is the billing for dish recognition. The latter cannot solve the problems of lighting, occlusion, deformation, etc. with traditional image processing methods, and will be subjected to many restrictions in practical applications. Therefore, more RFID chips are embedded in the bottom of the dish to identify the dish. This method has some disadvantages, especially the RFID chips are not resistant to high temperatures. The restaurant will easily damage the chips during high temperature sterilization. The replacement of the entire dish will increase the cost and is not conducive to the long-term stable development of the restaurant.

In order to solve the above problems, this paper applies convolutional neural network to dish detection. Based on Faster R-CNN, combining the characteristics of dishes and introducing an attention mechanism, a method of dish detection based on Cross Faster R-CNN is proposed. This method uses cross layer to fuse the edge, color and texture information of dishes with deep advanced features. The experimental results show that this method can reduce the miss rate of dishes and improve the detection accuracy. The main work of this article is as follows:

(1) Visualize Faster R-CNN, analyzing the features extracted by VGG16 (a convolutional neural network) layer by layer when detecting dishes, understanding the role of the convolution kernel of each layer. At the same time, using the heat map to determine which region features of the dishes have a greater response to the detection results.
(2) According to the visual analysis, the network structure of Cross Faster R-CNN is proposed by improving Faster R-CNN and combining the idea of cross-connect. The network fuses the low-level features of the dishes with the high-level features. Then the 1×1 convolution kernel is used to increase the nonlinearity of the network and improve the network's ability to express complex features.

2 Related Work

2.1 Convolutional Neural Network

Convolutional neural network [25, 36, 37] (CNN) is a type of feed forward neural network with convolutional calculation and deep structure. It belongs to representational learning and can automatically learn high-level features from samples without artificial design features. The structure of a convolutional neural network generally consists of a convolutional layer, a pooling layer, and a full connected layer. Compared to the hidden layer of a traditional neural network, the convolutional layer can greatly reduce the amount of parameters due to the characteristics of convolutional weight sharing. Thus, CNN can train models with deeper network structures. At the same time, the convolutional layer and pooling layer in the convolutional neural network can respond to the translation invariance of the input features, and can identify similar features located in different positions in space.

In order to extract more advanced features, the current convolutional neural network generally has a deep structure and a large number of network layers. However, the weight of the convolutional neural network is updated by gradient back propagation. A deeper network structure will cause the mean and standard deviation of the data change, resulting in a covariate shift phenomenon, that is, the disappearance of the gradient. In order to solve the above problems, *Ioffe S.* [26] and others proposed batch normalization (BN) in 2015. The features will be obtained through the convolution layer, and the data will be scaled using two linear parameters to meet the variance is 1, the mean is 0, and then the activation function is used as the input of the next layer. The process of the BN layer is as follows:

$$\mu = \frac{1}{m}\sum_{i}^{m}\left(X^{(i)} - \mu\right)^2 \tag{2-1}$$

$$\sigma^2 = \frac{1}{\mathrm{m}}\sum_{i}^{m}\left(X^{(\mathrm{i})} - \mu\right)^2 \tag{2-2}$$

$$Z^{(i)} = \frac{X^{(i)} - \mu}{\sqrt{\sigma^2 + \xi}} \tag{2-3}$$

$$\hat{Z}^{(i)} = \alpha * Z^{(i)} + \beta \tag{2-4}$$

where X is the original data, μ is the mean, σ^2 is the variance, $\hat{Z}$ is the scaled data, α and β are two newly added parameters, which are trained by the convolutional neural network to replace the bias ξ. First input data $X = \left\{x^{(1)}, x^{(2)}, \Lambda, x^{(m)}\right\}$, calculate the mean μ and variance of the data σ^2, then normalize Z, train the parameters α and β, and update $\hat{Z}$ through linear transformation. In the forward propagation, the new distribution value is obtained through the learnable α and β parameters; in the back propagation, the sum α and β the related weights are obtained through the chain derivation.

2.2 Visualization

Although the convolutional neural network has excellent performance in computer vision tasks, it still hasn't the ability to use mathematical formulas to explain the deep-level feature expression of neural networks. And humans and computers have different ways of expressing features. Convolutional neural network is a type of end-to-end representation learning, and people can only understand the input and output ends. The meaning of the feature map in the middle layer still can't be clearly explained, so the convolutional neural network is also called a "black box" model.

With the in-depth research in recent years, the "black box" of convolutional neural networks has gradually been opened up, and the internal working principle of convolutional neural network can be intuitively understood in a visual way [27, 28]. Common visualization methods at this stage include the visualization of feature map, the visualization of convolution kernels, and the heat map which can be used to understand the activation characteristics of category.

The feature map is the output of the convolutional layer. The visualization of the feature map is to display the output of the convolution kernel in the form of an image [38, 39], which can help understand the role of the convolution kernel. This is commonly used in the visualization of convolutional neural networks.

The process of convolution is essentially the process of feature extraction. Each convolution kernel represents a feature. If some regions in the image respond more to a convolution kernel, the region can be considered to be similar to the convolution. Therefore, the visualization of the convolution kernel can also be regarded as an optimization problem, that is, to find an image with the largest response to a certain convolution kernel. The whole process is as follows:

First, get an image I randomly, and find the gradient of K to I for the convolution kernel to be visualized, that is:

$$Grad = \frac{\partial K}{\partial I} \tag{2-5}$$

Then update the image I by gradient ascent, that is:

$$I = I + \alpha * Grad \tag{2-6}$$

where α is the step size, which is 1 in this experiment.

In 2017, *Selvaraju R.R* [29] and others proposed a gradient weighted class activation map (Grad-CAM), which uses a heat map to display the degree of activation of the picture for each region of the output category, it is consistent with human visual characteristics. To some extent, it explains the correlation between the output of the convolutional neural network and the images. The essence of Grad-CAM is that the output category weights the feature map obtained from a certain convolution layer through gradients. The entire process is as follows:

First, calculating the probability y^c of the output category c of the convolutional neural network *softmax* layer for the gradient of the pixel value of the feature map A_{ij} obtained by the convolution layer (the original text calculates the output before

the *softmax* layer, and in the experiment it was found that the heat map obtained by calculating the output probability after the *softmax* layer is more obvious), namely:

$$\frac{\partial y^c}{\partial A_{ij}} \tag{2-7}$$

where y^c is the output probability of category c, A is the convolution feature map, and k is the number of channels in the feature map.

Then the global average of the partial derivatives is obtained, that is:

$$a_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}} \tag{2-8}$$

where a_k^c represents the partial linearization from the deep network in A, that is, the importance of category c relative to the k-th channel in A.

Then a_k^c and A are weighted and combined and processed by the *Relu* activation function, that is:

$$L_{Grad-Cam}^c = \mathrm{Re}LU\left(\sum_k a_k^c A^k\right) \tag{2-9}$$

where L is a class activation map for category c.

2.3 Attention Mechanism

In general, the Attention mechanism is to focus attention on important points and ignore other unimportant factors. In this article, the edge of the dishes where attention needs to be focused on, other places, including the center of the dishes, are ignored.

The general definition of attention mechanism is shown in formula (2-10) [30]:

$$Attention(Q, K, V) = soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2-10}$$

where $Q \in R^{m\times d_k}$, $K \in R^{m\times d_k}$, $V \in R^{m\times d_k}$.

May wish to take $q_t \in Q$, then the encoding result obtained for a single input vector q_t can be expressed as [23]:

$$Attention(q_t, K, V) = \sum_{s=1}^{m} \frac{1}{Z}\exp\left(\frac{\langle q_t, k_s\rangle}{\sqrt{d_k}}\right)v_s \tag{2-11}$$

Where Z is the normalization factor, and q, k, v are shorthand for *query, key, value*. $\sqrt{d_k}$ regulates the internal product of the activation function so that it is not too large. The $\exp\left(\frac{\langle q_t,k_s\rangle}{\sqrt{d_k}}\right)v_s$ in formula (2-11) is the core part of the attention mechanism. $\exp\left(\frac{\langle q_t,k_s\rangle}{\sqrt{d_k}}\right)v_s$ is mainly used to measure the similarity q_t with v_s. The entire formula can be understood as looking for a non-linear mapping relationship between q_t and v_s.

The general flow of formula (2-11) is the inner product of q_t and each k_s, and then the similarity of q_t and v_s is evaluated by *softmax*. The final result is a vector of d_v by weighted summation. Observing the calculation process, other parts of the input matrix besides q_t will also affect the calculation result of the vector d_v. Therefore, d_v is not only related to q_t, but also related to other parts of the input matrix, and q_t can be used as the representation of q_t in the global vector.

Based on the *Attention* mechanism[40], the *Multi-Head Attention* mechanism proposed by *Google* is used to further improve the coding ability of the model[31,32]. The *Attention* mechanism model used in this paper is mainly based on *Multi-Head Attention.* Compared with the basic model, the *Multi-Head Attention* mechanism has two differences. One is to map Q, K, V via matrix parameters, and then send them to the *Attention* model. Second, the original input is subjected to *Attention* operations without sharing parameters multiple times, and the output results are stitched. These two improvements can improve the description ability of the model. The specific model is as follows:

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{2-12}$$

where $W_i^Q \in R^{d_k \times \tilde{d}_k}$, $W_i^K \in R^{d_k \times \tilde{d}_k}$, $W_i^V \in R^{d_v \times \tilde{d}_v}$, the feature representation of the final output can be expressed as:

$$MultiHead(Q, K, V) = Concat(head_1, \Lambda, head_h) \tag{2-12}$$

3 Dish Detection Based on Cross Faster R-CNN

3.1 Visualizing Faster R-CNN

Visualizing Faster R-CNN can help understand its working principle in dish detection, further clarify the features that are activated when making image classification decisions. At the same time, when it involves tasks in a specific field, it can improve the network structure based on prior knowledge and the accuracy of the model for this task.

Feature Map Visualization

Visually understand the differences between the extracted features of the low-level convolution layer and the high-level convolution layer. And the visualization of the feature map obtained by Conv1_2 convolution layer, Conv2_2 convolution layer, Conv3_3 convolution layer, Conv4_3 convolution layer, Conv5_3 convolution layer and the RPN_Conv convolution layer in Faster R-CNN respectively, as shown in Fig. 1. For the convenience of analysis, each feature map shows only the first 16 channels.

It can be seen that the features extracted by the shallow convolution layer are similar to the edge and color information, and the retained image information is relatively complete. As the number of layers becomes deeper, the image feature information obtained through the convolution layer gradually decreases, and the entire feature map becomes more abstract.

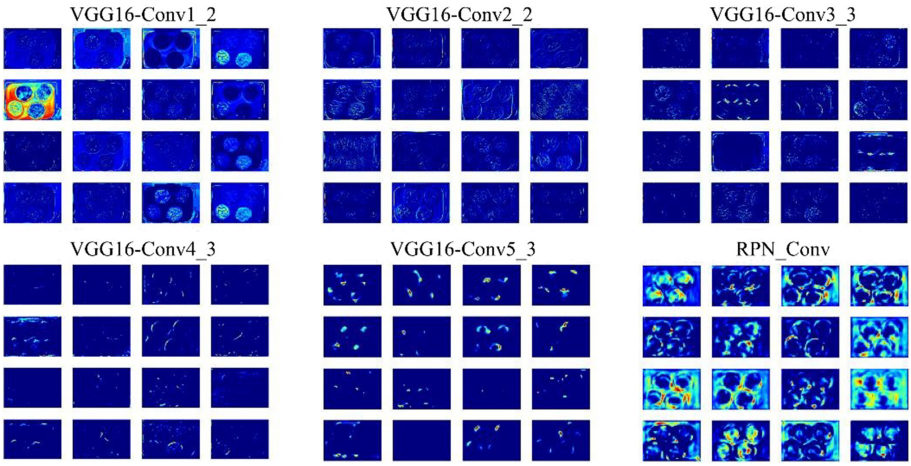

Fig. 1. Feature map extracted through each convolutional layer

RPN_Conv is obtained from the feature map obtained through the VGG16-Conv5_3 convolutional layer through a 3×3 convolutional layers and used as the input of the RPN network. From the visualization results, it can be seen that the information in this layer mainly relates to the characteristics of the dish, and also proves that the RPN network is effective for extracting foreground and background.

Convolution Kernel Visualization

The visualization of the convolution kernel can be seen from Sect. 2.2. It is necessary to randomly obtain a pair of original images. In this paper, the input image is used as the original image. Based on this, gradient rise is performed. The number of iterations is 100. The visualization results are shown in Fig. 2.

It is clear from the Fig. 2 that the low-level convolution kernel responds to features such as color, edge and texture, while the feature of interest in the high-level convolution kernel is an advanced abstracted information. These features are so complex that they are difficult to understand and explain. At the same time, it can also be found that the high-level convolution kernel is difficult to be visualized by the gradient rising method, and the obtained image has more noise points.

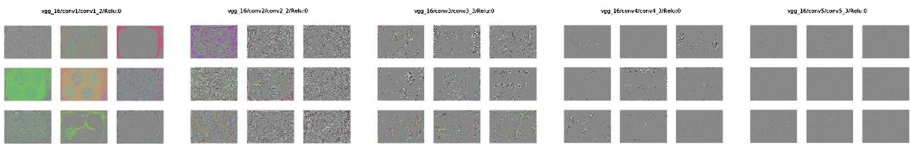

Fig. 2. Visualization of the convolution kernel at each layer

Gradient Weighted Class Activation Map

Using the Grad-CAM method to weighted and sum the feature maps obtained by the

convolutional layer in the convolutional netural network, the heat map obtained is shown in Fig. 3. It can be found that the features of the dishes corresponding to the feature map extracted by Conv5_3 which is the last convolutional layer of VGG16 in Faster R-CNN mainly involving the edge information of the dish.

For example, in Fig. 3, the circumference of Xiao_yuan_Pan is round, while Shuang_Er_Pan has two protrusions. In the process of extracting features from Conv1 to Conv5, the heat is gradually on the edge of the dishes. When it comes to Conv5, the heat of Xiao_yuan_Pan is at the edge of Xiao_yuan_Pan (top right), and the heat of Shuang_Er_Pan is concentrated on the two protrusions (bottom right).

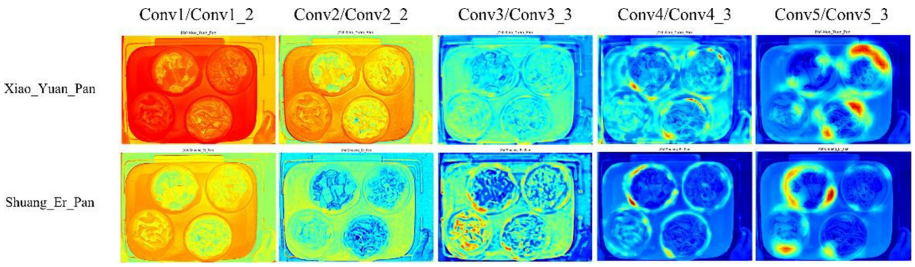

Fig. 3. Thermal map obtained by convolution of the layers corresponding to the categories Xiao_Yuan_Pan and Shuang_Er_Pan

Because transfer learning is used to train the dish detection model, VGG16′s pre-training weights come from the ImageNet dataset. At the same time, the Conv1 and Conv2 repeated convolution block weights are fixed when training the model. The heat maps of Conv1_2 and Conv2_2 are not aimed at the detection result of the dishes, and should be a general heat map for extracting shallow features.

3.2 Cross Faster R-CNN

Figure 4 is a schematic diagram of the Cross Faster R-CNN model. A Cross Layer is added to the Faster R-CNN, and the edge and texture features are combined with the deep-level advanced features extracted by VGG16 to enhance the model's feature expression ability for targets such as dishes.

Cross Faster R-CNN is mainly constructed by Faster R-CNN's basic modules and cross layers. As shown in Fig. 4, the extraction process of feature maps is completed by VGG16 and cross layers separately, then feature fusion is performed. A 1×1convolution kernel (Conv7) is used to modify the number of channels of the fused feature map to obtain the final feature map. Then the RPN network of Faster R-CNN is used to predict the background and foreground to get proposal boxes of the foreground. After that the proposal boxes and feature maps are passed through ROI Pooling (Region of Interest Pooling) to output feature maps with the same size. Finally, predicting the category and location of the object by the Full Connection network (FC).

The cross layer is composed of two convolutional layers and two pooling layers. The first convolutional layer (Conv2_6_1) which is composed of 128 3×3 convolution

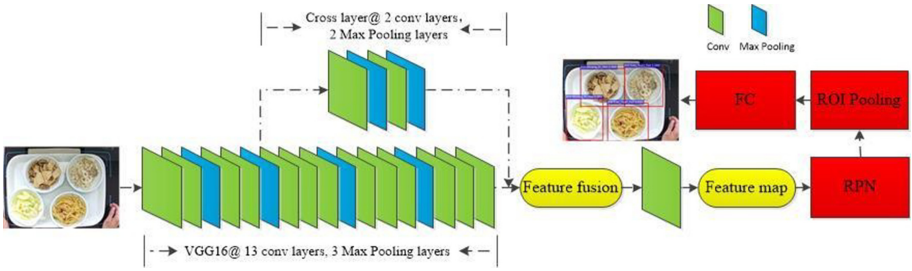

Fig. 4. Cross Faster R-CNN model

Table 1. Cross Faster R-CNN structure

Cross Faster R-	Cross Faster R-	Cross Faster R-
Cross Faster R-	Cross Faster R-	Cross Faster R-

kernels, followed by a 2×2 maximum pooling layer (Max Pooling2_6_1) reduces the height and width of the feature map, and then increases the number of channels of the feature map through a convolution layer (Conv2_6_2) composed of 256 1×1 convolution kernels, and finally a 2×2 maximum pooling layer (Max Pooling2_6_2) reduces the height and width of the feature map so that the feature dimensions obtained by cross layer match the feature dimensions obtained by VGG16. Because VGG16 speeds up training by pre-training weights, and fixed the weights of the Conv1 and Conv2 layers, the purpose of the 3×3 convolutional layer through a receptive field in the cross layer is to ensure the low-level characteristics of the shallow dishes, while the 1×1 convolutional layer is added to reduce the amount of parameters and increase the number of channels of the feature map and increase its nonlinear expression ability.

Cross Faster R-CNN does not fix the size of the input image. Table 1 shows the detailed network structure of the entire model more intuitively. VGG16 standard input is used as an example, the input image size is fixed at $224 \times 224 \times 3$. There are 11 categories and 9 anchors.

Note that the RPN_cls_score layer and the RPN_bbox_pred layer are independent of each other in the RPN structure block, and each layer is directly connected to the RPN_Conv layer. Similarly, in the FC structure block, the cls_score layer and the bbox_pred layer are also independent of each other and connected with the FC7 layer. For the convenience of explanation, the offset is not calculated in the parameters.

4 Experiment

4.1 Experimental Data

The data set used in the experiment was collected from the actual restaurant using a collection of collection and identification equipment, and are the dishes and food being sold and used in the restaurant. Samples were labeled and verified by 14 researchers and

the correct rate of the labeled files was higher than 99.9%. Figure 5 shows some of the original samples and labeled samples. In these data, each dish is equipped with each type of food. The network obtained by training in this way can improve the accuracy rate during the test.

We tried to put different dishes into different dishes to show the correspondence between food and dishes, but the results of such training were not ideal, so we introduced the attention mechanism and put the same food in each dish. In this way, the neural network will pay attention to the edges of the dishes during training, so as to extract the edge features of the dishes and reduce the interference of the food.

(a) (b)

Fig. 5. Experimental dataset. (a) original sample; (b) labeled sample.

There are a total of 31,111 samples of 10 types of dishes in the entire dataset. Each sample contains 1 to 5 types of dishes and the number of each type is about 3000 to 5000, as shown in Fig. 6. The training set contains 17,421 samples, the validation set contains 7,467 samples, and the test set contains 6,223 samples.

At the same time, 150 samples were collected during the actual use of the restaurant, as shown in Fig. 7. The pictures collected in actually will have some additional background interference, such as chopsticks, mobile phones, payment cards and other items. Labelling these background interferences (labeled as background) to participate in the training, so that items such as chopsticks, payment cards, and mobile phones will be recognized as background during prediction, and will not interfere with the recognition of dishes. Cross Faster R-CNN does't fix the input size of the samples. The resolution of each sample in this dataset is $(690 \pm 10) \times (520 \pm 30)$.

4.2 Analysis of Results

The experiments were performed on a PC, the operating system was Windows 10, and the RTX 2080ti with a GPU of 11G was used to accelerate training and model testing. The models used were implemented under Python 3.6 and Tensor flow 1.8.0.

The test is performed on the test set and the data set collected in actual use, as shown in Table 2 and 3, where Table 2 is the test set result and Table 3 is the actual collected data set result. The evaluation indicators are accuracy, recall and detection speed. Among them, the accuracy rate is a comprehensive index of TOP-1 classification accuracy rate and IOU of the regression box is bigger than 0.5, the recall rate is the ratio of correctly distinguishing foreground and background, and the detection speed is the

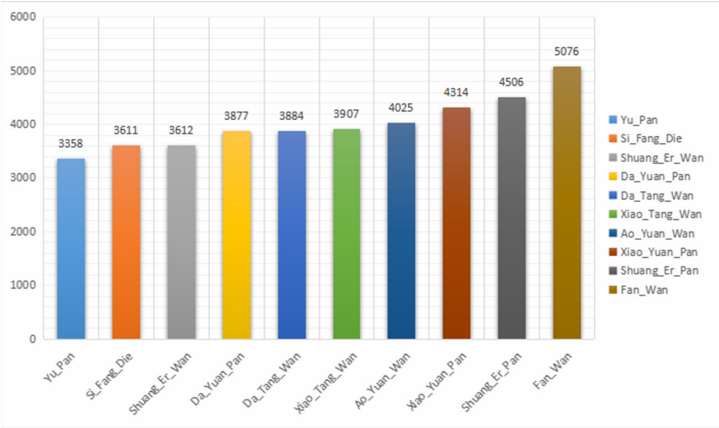

Fig. 6. Number of dishes per type

Fig. 7. The actual collection data set

time required to averagely detect an image under the GPU (excluding the first model loading time during detection). Both the test set and the actual data set are labeled, so it is only necessary to check whether the label is consistent with the actual prediction result.

Table 2. Test results

	Accuracy(%)	Recall(%)	Speed(s)
Faster R-CNN	99.87	99.51	0.1453
Cross Faster R-CNN	99.91	99.67	0.1527

Analysis of the above table shows that the detection time of Cross Faster R-CNN has little increased regardless of whether it is in the test set or the actual collected data set. The accuracy and recall of Cross Faster R-CNN are higher than Faster R-CNN, it also proves that for the detection of dishes, the fusion of low-level features can improve

Table 3. Test results of actual collected data sets

	Accuracy(%)	Recall(%)	Speed(s)
Faster R-CNN	99.52	99.21	0.1501
Cross Faster R-CNN	99.64	99.45	0.1538

the detection accuracy. At the same time, from the experimental results, Cross Faster R-CNN may have a better background suppression than Faster R-CNN.

5 Conclusion

By visualizing Faster R-CNN and analyzing the importance of features in dish detection to the detection results, this paper proposes a Cross Faster R- CNN model. The cross layer is used to combine the low-level features at the shallow level with the high-level features at the deep level. The features of the low-level layer include the features of Conv1, and the feature of the high-level layer is the feature of Conv5. The dataset has been filled with each type of food in each type of dishes to improve the detection accuracy. The experimental results show that the model improves the detection accuracy of the dishes and suppresses the background interference when the detection time does not increase significantly. In the following work, research will focus on optimizing the structure of the network model to reduce the time required for detection.

References

1. Jia, X., Li, R.: Deep learning and artificial intelligence. Neijiang Technol. **41**(06), 78–78+84 (2020)
2. Jiang, D., Wang, Y., Lv, Z., Wang, W., Wang, H.: An energy-efficient networking approach in cloud services for IIoT networks. IEEE J. Sel. Areas Commun. **38**(5), 928–941 (2020)
3. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
4. Qi, S., Jiang, D., Huo, L.: A prediction approach to end-to-end traffic in space information networks. Mobile Networks and Application (2019). online available
5. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
6. Breier, B., Onken, A.: Analysis of video feature learning in two-stream CNNs on the example of zebrafish swim bout classification. arXiv:1912.09857 (2019)
7. Jiang, D., Wang, W., Shi, L., Song, H.: A compressive sensing-based approach to end-to-end network traffic reconstruction. IEEE Trans. Network Sci. Eng. **7**(1), 507–519 (2020)
8. Levine, S., Pastor, P., Krizhevsky, A., et al.: Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. Int. J. Robot. Res. **37**(4–5), 421–436 (2018)
9. Liang, T., Ling, H.: MFPN: a novel mixture feature pyramid network of multiple architectures for object detection. arXiv:1912.09748 (2019)
10. Liu, Y., Jin, L.: Exploring the capacity of sequential-free box discretization network for omnidirectional scene text detection. arXiv:1912.09629 (2019)

11. Jiang, D., Huo, L., Song, H.: Rethinking behaviors and activities of base stations in mobile cellular networks based on big data analysis. IEEE Trans. Network Sci. Eng. **7**(1), 80–90 (2020)
12. Zhou, Q., Yang, W., Gao, G., et al.: Multi-scale deep context convolutional neural networks for semantic segmentation. World Wide Web **22**(2), 555–570 (2019)
13. He, Y., Fritz, M.: Segmentations-leak: membership inference attacks and defenses in semantic image segmentation. arXiv:1912.09685 (2019)
14. Zhao, L., Tao, W.: JSNet: joint instance and semantic segmentation of 3D point clouds. arXiv: 1912.09654 (2019)
15. Yu, Q., Xu, D.: C2FNAS: coarse-to-fine neural architecture search for 3D medical image segmentation. arXiv:1912.09628 (2019)
16. Jiang, D., Wang, Y., Lv, Z., Qi, S., Singh, S.: Big data analysis based network behavior insight of cellular networks for industry 4.0 applications. IEEE Trans. Ind. Inform. **16**(2), 1310–1320 (2020)
17. Bai, M, Urtasun, R.: Deep watershed transform for instance segmentation. In: /Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5221–5229 (2017)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105(2012)
19. Johnson, J., Karpathy, A., Fei-Fei, L.: Densecap: Fully convolutional localization networks for dense captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4565–4574 (2016.
20. Jiang, D., Huo, L., Lv, Z., Song, H., Qin, W.: A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. IEEE Trans. Intell. Transp. Syst. **19**(10), 3305–3319 (2018)
21. Young, T., Hazarika, D., Poria, S., et al.: Recent trends in deep learning based natural language processing. IEEE Comput. IntelligenCe Mag **13**(3), 55–75 (2018)
22. Jiang, D., Huo, L., Li, Y.: Fine-granularity inference and estimations to network traffic for SDN. PLoS ONE **13**(5), 1–23 (2018)
23. Milz, S., Arbeiter, G., Witt, C., et al.: Visual slam for automated driving: Exploring the applications of deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 247– 257(2018)
24. Wang, Y., Jiang, D., Huo, L., Zhao, Y.: A new traffic prediction algorithm to software defined networking. Mobile Networks and Applications, 2019 (2019). online available
25. Gu, J., Wang, Z., Kuen, J., et al.: Recent advances in convolutional neural networks. Pattern Recogn. **77**, 354–377 (2018)
26. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
27. Wagner, J., Kohler, J.M., Gindele, T., et al.: Interpretable and fine-grained visual explanations for convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9097–910 (2019)
28. Jiang, D., Zhang, P., Lv, Z., et al.: Energy-efficient multi-constraint routing algorithm with load balancing for smart city applications. IEEE Internet Things J. **3**(6), 1437–1447 (2016)
29. Selvaraju, R.R., Cogswell, M., Das, A., et al.: Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626 (2017)
30. Zhong, C., Zhou, H., Wei, H.: A 3D point cloud object recognition method based on attention mechanism. Key R&D plan projects of the Ministry of Science and Technology (2019)
31. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 998–6008. [S.1.], Long Beach, USA; IEEE (2017)

32. Jiang, D., Li, W., Lv, H.: An energy-efficient cooperative multicast routing in multi-hop wireless networks for smart medical applications. Neurocomputing **220**, 160–169 (2017)
33. Zhou, Y., Zhang, R.: A brief analysis of subtitle translation of documentary wild china from the perspective of eco-translatology. Theory Pract. Language Studies (TPLS) **9**(10), 1301–1308 (2019)
34. Hong, J., Park, S., Byun, H.: Selective residual learning for Visual Question Answering. Neurocomputing **402**, 366–374 (2020)
35. Liu, D., Cui, Y., Chen, Y.: Jiyong Zhang; Bin Fan, Video object detection for autonomous driving: Motion-aid feature calibration. Neurocomputing **409**, 1–1 (2020)
36. Chirra, V.R.R., Uyyala, S.R., KishoreKolli, V.K.: Deep CNN: a machine learning approach for driver drowsiness detection based on eye state. Revue d'Intelligence Artificielle, 33(6), EI (2020)
37. Vo, S.A., Scanlan, J., Turner, P., Ollington, R.: Convolutional Neural Networks for individual identification in the Southern Rock Lobster supply chain. Food Control **118**, 107419 (2020)
38. Ebani, E.J., Kaplitt, M.G., Wang, Y., Nguyen, T.D., Askin, G., Chazen, J.L.: Improved targeting of the globus pallidus interna using quantitative susceptibility mapping prior to MR-guided focused ultrasound ablation in Parkinson's disease. Clin. Imaging **68**, 94–98 (2020)
39. Madhar, S.A., Mraz, P., Mor, A.R., Ross, R.: Empirical analysis of partial discharge data and innovative visualization tools for defect identification under DC stress. Int. J. Electr. Power Energy Syst. **123**, 106270 (2020)
40. Luo, M., Wen, G., Yang, H., Dai, D., Ma, J.: Learning competitive channel-wise attention in residual network with masked regularization and signal boosting. Expert Syst. Appl. **160**, 113591 (2020)