p-Faster R-CNN Algorithm for Food Detection

Yanchen Wan[1(✉)], Yu Liu[1], Yuan Li[2], and Puhong Zhang[2]

[1] State Key Laboratory of Software Development Environment, Beihang University, Beijing, China
wyc491946376@163.com, buaa_liuyu@buaa.edu.cn
[2] The George Institute for Global Health at Peking University Health Science Center, Beijing 100191, China

Abstract. Eating healthily helps prevent disease, and it can be achieved by identifying the kinds and ingredients of the food to determine whether the diet is healthy. In this paper, we innovatively propose p-Faster R-CNN algorithm for healthy diet detection, which is based on Faster R-CNN with Zeiler and Fergus model (ZF-net) and Caffe framework. Before the input layer, the Gauss Pyramid is applied to form a multi-resolution pyramid of images, which expands the number and the scale of the samples. In the training stage, the multi-scale Spatial Pyramid Pooling Layer is added after the convolution layer to extract multi-scale features. To evaluate the performance of p-Faster R-CNN, we compare it with Fast R-CNN and Faster R-CNN. The experiment results demonstrated that p-Faster R-CNN increases the AP value of each kind of food by more than 2% compared with Faster R-CNN, and p-Faster R-CNN, Faster R-CNN are superior to Fast R-CNN in accuracy and speed. At last, the total dataset we established is used to construct the application of judging the healthy diet by uploading intake photos.

Keywords: Food health · Object detection · Faster R-CNN · Pyramid Convolutional neural network

1 Introduction

Healthy diet plays a key role in the prevention of chronic diseases, such as hypertension, blood fat, obesity, etc. High intake of Na is a contributing factor to hypertension, fat intake has a direct effect on blood fat, and calorie intake is directly associated with obesity. So with rapid identification of the kind of food, food calories, fat and Na degree on the basis of food ingredients, we can determine whether food is healthy. Using this idea, "intelligent diet" smartphone application can be constructed for the realization of innovative service model of Internet plus health management.

As the mainstream of artificial intelligence, machine learning has concerned "diet" in the field of agricultural production. Steven Melendez introduces a Climate Corporation dedicated to establish an agricultural digital analysis center [1]. By interpreting soil data, it guides growers to use seeds, water, pesticides, fertilizers more precisely. At the same time, machine learning also takes interest in "health" in the aspects of

I. Romdhani et al. (Eds.): CollaborateCom 2017, LNICST 252, pp. 132–142, 2018.
https://doi.org/10.1007/978-3-030-00916-8_13

medical. A variety of diseases can be measured and diagnosed by the related medical data. In 2016, a biosensor company called Sentrian has developed medical diagnosis system based on the data [2]. This system owns the information of patients, and makes early diagnosis by observing subtle associations. However, the above applications only regard diet or health alone. There is no relevant machine learning combing diet with health.

In this paper, we put forward the research target of healthy diet detection based on deep learning. We make the following major contributions through this paper:

1. We use Gaussian Pyramid to get multi-scale pictures.
2. We use multi-scale Spatial Pyramid Pooling Layer to increase feature diversity.
3. Object detection is applied in food and health fields.

This paper is divided into five sections, the third, fourth sections are the point. Section 1 is the research background and objectives. Section 2 presents object detection algorithm. Section 3 is a detailed description of the p-Faster R-CNN. Section 4 shows the experimental results, and Sect. 5 concludes the paper.

2 Related Work

Object detection began to integrate the neural network algorithm in 2012. Until 2014, Ross Girshick published [3] in CVPR2014, which marks the beginning of object detection in the deep learning. To avoid mapping for each proposal, Kaiming He published [4] in ECCV2014, and put forward the "spatial pyramid pooling" concept. Then, Ross Girshick's [5] is a combination of R-CNN and SPP-net. In 2015, Faster R-CNN first proposed "Region Proposal Network (RPN)", which incorporates the proposal extraction into convolution network. In order to realize the real-time detection, J Redmon published [6] in 2016. YOLO integrates proposal prediction and classification into a single neural network model. After that, the SSD algorithm based on YOLO was introduced, and the object detection technology is developing rapidly.

Image processing commonly uses the method of pyramid. There are two types of pyramid: low-pass, bandpass. Low-pass Pyramid is smoothed by a suitable smoothing filter and sampled smooth image. Bandpass Pyramid is formed by interpolation algorithms between adjacent images [7]. Pyramid can produce image features of different resolutions in scale space. Image pyramid is the foundation, and feature pyramid is expansion. Multiscale feature extracted by SIFT show that multi scales can be very useful.

Faster R-CNN has made a perfect detection effect, but the training and testing are in a single scale image without considering the multi-scale feature extraction [8]. To solve this problem, this paper proposes p-Faster R-CNN algorithm for food detection, which implements multi-scale pyramid at the stage of image preprocessing and feature extraction.

3 p-Faster R-CNN

In [3], Ross Girshick uses two different scales of image (180 * 180, 224 * 224) to train the network. In the same network structure, the same fixed length feature is extracted in the SPP layer and tested in the ImageNet 2012 dataset, and the results are as follows:

From Table 1, compared SPP single-size trained with no SPP, the top-5 error is reduced by 0.62%, 0.38%, 0.72%, 0.85%, which shows that the SPP layer can improve the detection accuracy; and compared SPP single-size trained with SPP multi-scale trained, the top-5 error is reduced by 0.50%, 0.44%, 0.61%, 0.68%, which indicates that multi-scale images can improve the result of object detection.

Table 1. No SPP, single scale SPP, multi-scale SPP error contrast table

Model	SPP Type					
	No SPP		SPP single-size trained		SPP multi-size trained	
	Top-1 (error %)	Top-5	Top-1	Top-5	Top-1	Top-5
ZF-5	35.99	14.76	34.98	14.14	34.60	13.64
Convnet-5	34.93	13.92	34.38	13.54	33.94	13.33
Overfeat-5	34.13	13.52	32.87	12.80	32.26	12.33
Overfeat-7	32.01	11.97	30.36	11.12	29.68	10.95

In Faster R-CNN algorithm proposed by [8], the input image scale is single, and according to the real-time demand, SPP layer only using a scale feature extraction. So inspired by [4], we improved Faster R-CNN in consideration of the multi-scale characteristics and images, and proposed a p-Faster R-CNN algorithm. The improvement is mainly manifested in two aspects: Gauss Pyramid Layer for the input images; Multi-scale SPP Layer for feature extraction.

3.1 Gaussian Pyramid Layer

Training the network with different sizes of images can increase the scale invariance and reduce overfitting. The pyramid algorithm in computer vision can generate different scales images to achieve this goal. Here, we adapt the widely-used Gaussian Pyramid.

Gaussian Pyramid is a multiscale description of a graph. It usually consists of 2 steps: smoothing through a low-pass filter; sampling or interpolating a smoothing image to obtain a series of reduced or enlarged images [9].

For each input image in the network, we use 5 * 5 convolution filter and interlaced drop sampling.

$$G_k(x,y) = \sum_{i=-2}^{2} \sum_{j=-2}^{2} w(i,j) G_{k-1}(2x+i, 2y+j) \tag{1}$$

$$w(i,j) = \hat{w}(i) \times \hat{w}(j) \tag{2}$$

Here $G_k(x, y)$ is the layer k of image pyramid, $w(i,j)$ is convolution filter, $\hat{w}(i)$ is Gauss density distribution function. After Gaussian pyramid G_0 forms a N layer image pyramid G_1,G_2, …, G_n. Each input picture is processed into 5 scales images in p-Faster R-CNN.

3.2 The Spatial Pyramid Pooling Layer

The Spatial Pyramid Pooling Layer can not only accept input images of any size, but also generate fixed-length feature representation.

Feature Mapping
In object detection, the SPP layer can map the features only once compared to the R-CNN without extracting features for each proposal. We use spatial mapping to find the location of the representation in the feature map corresponding to the proposal of the original image. Suppose that (x', y') is the coordinate point on the feature map. (x, y) is the point on the original picture. Then there is the following conversion relationship between them:

$$(x, y) = (S * x', S * y') \tag{3}$$

S is the product of all strides in CNN. In turn, from (x, y) to (x', y'):

$$x' = \lfloor x/S \rfloor + 1 \tag{4}$$

$$y' = \lfloor y/S \rfloor + 1 \tag{5}$$

Multi-scale Feature Extraction
Adding a multi-scale Spatial Pyramid Pooling Layer after the last convolution layer in Faster R-CNN can generate multi-scale representation in the feature map. In [4], the network allows arbitrarily sized images through the SPP layer which normalizes uniformly.

There is no multiscale pooling in Faster R-CNN, and the image can be divided into the length of 16 * 256-d with single scale. In order to improve the accuracy, we learn from the multi-scale pyramid pooling in [4]. At the end of the pooling layer, a multi-scale SPP layer is added to extract the features in four scales, which emphasizes both local and whole image information.

As shown in Fig. 1, regardless of size, the input pictures are divided into 16 blocks, 9 blocks, 4 blocks, 1 blocks, a total of 16 + 9 + 4 + 1 = 30 bin, which means the representation is the fixed 30 * 256-d. Then they are handled by the full connection layer and the classification layer.

In a word, a feature map is obtained by the convolution layer at the front, then features are extracted from the entire image by mapping. At last, we apply the spatial pyramid to each candidate window to gain the fixed length representation.

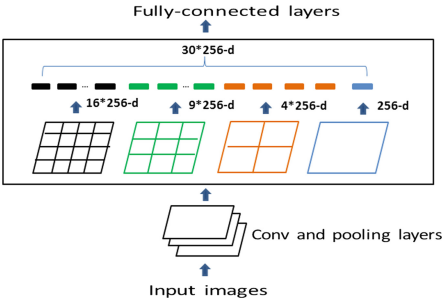

Fig. 1. Schematic diagram of the multi-scale SPP.

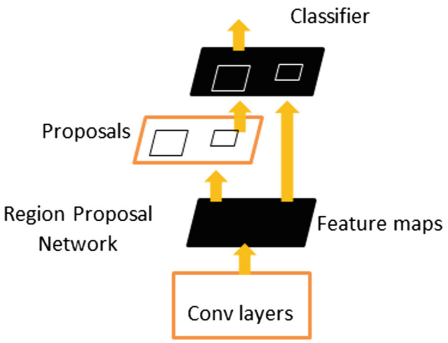

Fig. 2. Schematic diagram of Faster R-CNN.

3.3 p-Faster R-CNN

p-Faster R-CNN is an improved Faster R-CNN with pyramid algorithm.

Faster R-CNN

Faster R-CNN has two networks, RPN (Region Proposal Networks) and Fast R-CNN (detection network). As shown in Fig. 2, the RPN network accepts an input picture and outputs a series of proposals with the object's score. In order to generate accurate proposals, we slide a n * n window on the feature map to get a low dimensional feature vector. Then the feature vectors are fed into the classification network (Fast R-CNN).

Zeiler and Fergus Model (ZF-net)

The p-Faster R-CNN uses ZF-net [10] which has 5 conv layers and 3 fc layers. It takes 224 * 224 images (3 channels color planes) as input, and is convolved with 96 different filters whose size is 7 * 7 with a stride of 2. The feature maps are then passed through a linear function, pooled (3 × 3 regions, stride 2) and contrast normalized across feature maps. Similar operations are repeated in layer 2, 3, 4, 5. The last two layers are fully connected, and the final layer is a k-way softmax function [8].

p-Faster R-CNN

The p-Faster R-CNN adds the Gauss Pyramid Layer before the input of Faster R-CNN, and the multi-scale SPP layer before the full connection layer. As shown in Fig. 3, each input picture is sampled into 5 scales by the Gauss Pyramid Layer. The representation are extracted by dividing input pictures into 16 blocks, 9 blocks, 4 blocks, 1 blocks to form the fixed 30 * 256-d in the multi-scale SPP layer. The loss function, the training process, and the parameters are same with the Faster R-CNN.

This network is optimized using the loss function of multiple tasks (regression function and classification function):

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}\left(p_i, p_i^*\right) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}\left(t_i, t_i^*\right) \tag{6}$$

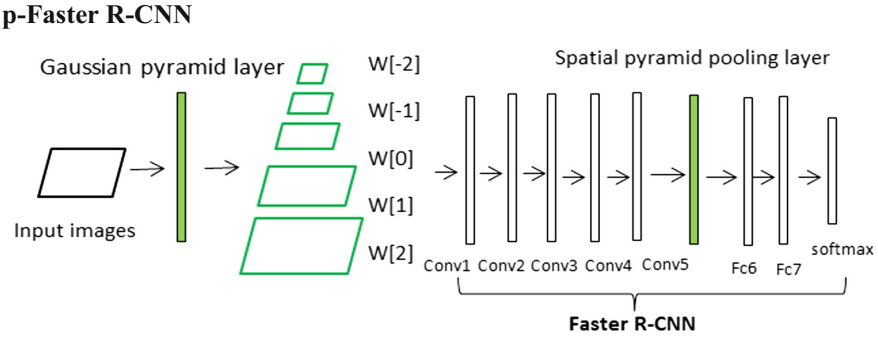

Fig. 3. The architecture of p-Faster R-CNN.

i is the index of anchor in the mini-batch, p_i is the predicted probability of anchor i being an object. p_i^* is the ground-truth label. t_i^* is a 4 dimensional vector representing the coordinate of the proposal. t_i^* is the coordinate of positive anchor. The classification loss L_{cls} is log loss over two classes (object vs. not object). The regression loss L_{reg} can be represented as:

$$L_{reg}\left(t_i, t_i^*\right) = R\left(t_i - t_i^*\right) \tag{7}$$

R is the robust loss function. The term $p_i^* L_{reg}$ means L_{reg} is activated when $p_i^* = 1$ and disabled otherwise ($p_i^* = 0$). The above two terms are normalized with N_{cls} and N_{reg}, and λ is a balancing weight.

4 Experiment

4.1 Software and Hardware

Deep learning framework of this project is Caffe with python interface. We need to configure Caffe and pycaffe. Due to the long training time, we equip GPU and CUDA computing architecture assisted GPU complex calculation. The graphics card is GeForce GTX TITAN X, and the corresponding driver is nvidia-375.39.

4.2 Datasets

The database includes 400 kinds of single food which refers to the original ecological food, not mixed with other foods, such as fruits, vegetables, steamed bread, etc. The food is divided into western food and Chinese food.

For western food, we download the Food-101 from the Computer Vision Lab (CVL). It includes 101 kinds of western food, and each kind has 1000 pictures sized 40–100k. For Chinese food, 300 kinds are listed according to the national standards of food classification system first of all. Then they are collected from Baidu and Google by crawler. And each class contains about 500 pictures.

For object detection, the input image must be tagged imitating the data format of PASCAL VOC2007. We mark the location of the object with a rectangular block, and generate the corresponding XML file with the frame's coordinates and the object class.

4.3 Implementation

As shown in Fig. 4, an arbitrary scale image (3 channel color plane) is used as input. In Gaussian Pyramid Layer, the image G_0 is sampled 4 times as G_1, G_2, G_3, G_4, and they are the input of the ZF-net.

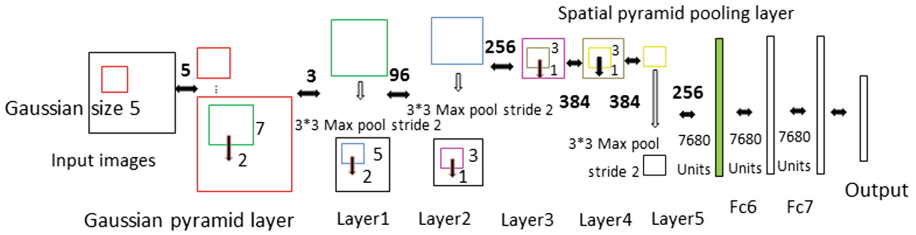

Fig. 4. Details of the network architecture of p-Faster R-CNN.

Then through 5 layers of convolution and pooling: the first layer has 96 different convolution filters whose kernel size is 7 * 7 with stride 2 and pooling window is 3 * 3 with stride 2; the second layer owns 256 different 5 * 5 filters and 3 * 3 pooling window with both stride 2; the third layer consists of 384 different 3 * 3 filters with stride 1, not pooling; The fourth layer is the same as the third one. There are 256 different 1 * 1 filters with stride 1 and 3 * 3 pooling window with stride 2 in the fifth layer. Then the multi-scale Spatial Pyramid Pooling Layer is connected.

In the SPP layer, 4 layers of spatial pyramid (1×1, 2×2, 3×3, 4×4, a total of 30 bins) are used to extract features. Each window generates the representations of 7680 (30×256) in length. These features are sent to the full connection layer.

There are 50% of the positive samples, 50% of the negative samples. We use a learning rate of 0.001 for 60k mini-batches, and 0.0001 for the next 20k mini-batches. The threshold of NMS is set to 0.7.

4.4 Experimental Result

We select the food pictures that are not trained to test and get good visualization results (Fig. 5). We test the some of the food dataset, calculate the accuracy of detection. The AP of each type remains above 0.7, as shown above (Table 2).

Fig. 5. Visualization results of p-Faster R-CNN

Table 2. The AP of part of food using p-Faster R-CNN.

Category	AP
apple-pie	0.7401
almond	0.8689
bread	0.7454
cheesecake	0.8077
chicken_wings	0.8884
hamburger	0.8944
hot_dog	0.9021
ice-creams	0.8699
macadamia	0.8748
peanut	0.7471
pine_nut	0.8741
raisin	0.8964
soup	0.9976
spaghetti_carbonara	0.9454
sesame	0.8403

4.5 Experimental Contrast

Experimental Dataset

We select 20 kinds of food in food-101 and mark them as VOC2007 data format for experimental comparisons. They are apple-pie, bread, cheesecake, baby bac ribs, baklava, beef carpaccio, beef tartare, chicken wings, club sandwich, donuts, egg, french fries, hamburger, hot dog, ice-cream, onion ring, pizza, steak, soup, spaghetti.

Contrast Between p-Faster R-CNN and Fast R-CNN

We use VOC 2007 and VOC 2012 dataset for Fast R-CNN and p-Faster R-CNN, and obtain their mAP and running speed. As Region Proposal Network (RPN) shares full-image convolutional features with the detection network in p-Faster R-CNN, thus enabling nearly cost-free region proposals [8]. It shows the framework of p-Faster R-CNN algorithm is better. As shown in Table 3, "method" represents algorithm; "prop" represents the algorithm and number of proposals; "*val*" is the validation set; and "*test*" is the test set. "07": VOC 2007 trainval, "07 + 12": union set of VOC 2007 trainval and VOC 2012 trainval.

Table 3. mAP Comparison between Fast R-CNN and p-Faster R-CNN

Method	Prop	Training data	Val (mAP)	Test (mAP)	Time (ms)
Fast RCNN	SS,2000	VOC 07	66.2%	63.9%	1830
Fast RCNN	SS,2000	VOC 07+12	68.5%	67.7%	1830
p-Faster RCNN	RPN,300	VOC 07	69.1%	68.0%	342
p-Faster RCNN	RPN,300	VOC 07+12	72.3%	70.7%	342

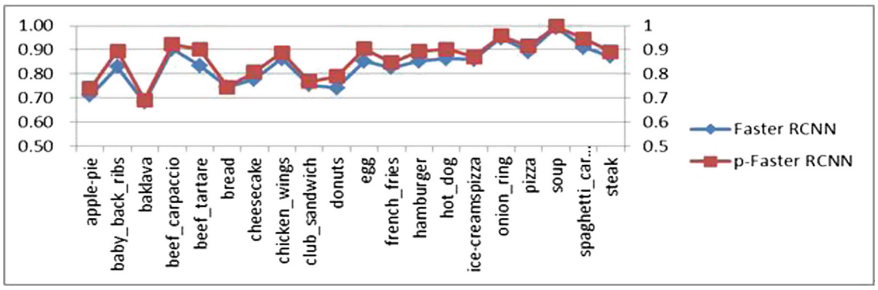

Fig. 6. The AP of Faster RCNN and p-Faster RCNN in each class (Color figure online)

Contrast Between Faster R-CNN and p-Faster R-CNN

We test the above experimental dataset, respectively using Faster RCNN and p-Faster RCNN. And the AP results are shown in Fig. 6. The abscissa is the name of each kind of food, a total of 20; the ordinate represents the AP value (the Maximum is 1, the interval is 0.1). The blue line represents the AP of each food for Faster RCNN, and the red line is that for p-Faster RCNN. As can be seen, for baby_back_ribs, beef_tartare, egg, p-Faster RCNN results improved significantly that AP value increased by about 5%. For cheesecake, donuts, hamburger, hot_dog, the results of p-Faster RCNN slightly improved that the AP value increased by about 2%. By contrast, the red line has been always above the blue line, which means that the detection accuracy of p-Faster R-CNN is slightly higher than that of Faster R-CNN for each kind.

Table 4 shows the AP value of Faster R-CNN and p-Faster RCNN. The mAP for Faster RCNN is 0.8360, and p-Faster RCNN is 0.8741. Compare both AP and mAP, p-Faster RCNN algorithm is more superior without considering the running speed.

4.6 Application

300 types of Chinese food and 100 kinds of food in food-101 are used as the dataset of the application.

As shown in Fig. 7, the food health application is divided into three layers. The client layer is the user interface. The business logic layer can calculate calories and other contents of food after detecting its type, and judge whether it's healthy by criteria. The data storage layer mainly manages and transfers the food information.

Table 4. The AP of food using Faster R-CNN and p-Faster R-CNN

Category	Faster R-CNN	p-Faster R-CNN
apple-pie	0.7099	0.7401
baby_back_ribs	0.8282	0.8955
baklava	0.6828	0.6902
beef_carpaccio	0.9024	0.9225
beef_tartare	0.8342	0.9033
bread	0.7454	0.7454
cheesecake	0.7776	0.8077
chicken_wings	0.8634	0.8884
club_sandwich	0.7541	0.7674
donuts	0.7408	0.7900
egg	0.8533	0.9043
french_fries	0.8264	0.8467
hamburger	0.8527	0.8944
hot_dog	0.8639	0.9021
ice_cream	0.8597	0.8699
onion_ring	0.9494	0.9591
pizza	0.8916	0.9150
soup	0.9932	0.9976
spaghetti_carbonara	0.9091	0.9454
steak	0.8718	0.8918
mAP	**0.8360**	**0.8741**

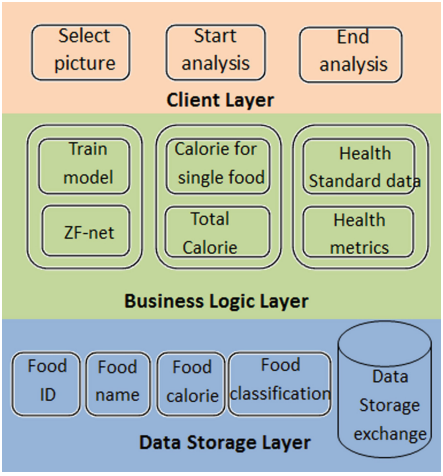

Fig. 7. The Application system architecture diagram.

5 Conclusion

This paper introduces the process of food data acquisition, the improvement process of p-Faster R-CNN algorithm, the network model, and the whole process of network configuration, training and testing. The correlation algorithms are compared and analyzed by the AP and mAP of each kind of food, which shows the superiority of the p-Faster R-CNN in a full range.

Acknowledgements. This research is supported by The National Key Research and Development Program of China (2016YFC1300205).

References

1. Melendez, S.: How machine learning will change what you eat. Mind and Machine (2016)
2. Siva, N.: Machine learning will keep us healthy. Lancet (2016)
3. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 10, pp. 2–6 (2014)
4. He, K., Zhang, X.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1904–1916 (2014)

5. Girshick, R.: Fast R-CNN. In: Computer Vision and Pattern Recognition, vol. 9, pp. 3–10 (2015)
6. Redmon, J., Divvala, S., Girshick, R.: You only look once: unified, real-time object detection, vol. 6, pp. 1–4 (2015)
7. Andelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J., Ogden, J.M.: Pyramid methods in image processing. RCA Eng. **29**(6), 33–41 (1984)
8. Ren, S., He, K.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 1137–1149 (2015)
9. Lan, Z., Lin, M., Li, X.: Beyond Gaussian pyramid: multi-skip feature stacking for action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 204–212 (2015)
10. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53