Graphic Logo Detection with Deep Region-based Convolutional Networks

Yuanyuan Li [#1], Qiuyue Shi [#2], Jiangfan Deng [#3], Fei Su [#4]

[#] *School of Communication and Information Engineering, Beijing University of Posts and Telecommunications, Beijing, China*

[#] *Beijing Key Laboratory of Network System and Network Culture, Beijing University of Posts and Telecommunications, Beijing, China*

[1] liyuan94@bupt.edu.cn [2] shiqiuyue@bupt.edu.cn
[3] afanti@bupt.edu.cn [4] sufei@bupt.edu.cn

***Abstract*—Logo detection is a challenging task with many practical applications in our daily life and intellectual property protection. The two main obstacles here are lack of public logo datasets and effective design of logo detection structure. In this paper, we first manually collected and annotated 6,400 images and mix them with FlickrLogo-32 dataset, forming a larger dataset. Secondly, we constructed Faster R-CNN frameworks with several widely used classification models for logo detection. Furthermore, the transfer learning method was introduced in the training process. Finally, clustering was used to guarantee suitable hyper-parameters and more precise anchors of RPN. Experimental results show that the proposed framework outperforms the state of-the-art methods with a noticeable margin.**

***Index Terms*—Logo detection, Faster R-CNN, Data augmentation, Network modification.**

I. Introduction

Logo detection and recognition is a widely used task in many applications, such as intellectual property protection, automatic driving and product brand promotion. As the unique identification for a brand, logo plays an important role in business advertisement. Logo detection can help merchants optimize their marketing schemes, fast matching suitable products for customers. In addition, this technology also restrains the tendency of fake advertising thus guarantees healthy circumstances for network market.

In general, logo detection comes in two forms, including character form and graphic form. In recent works, graphic logos are widely researched since they contain more semantic information and cover more various scenes. However, it's quite challenging to detect logos from real-world images as the identical logo exists complex variations such as diverse shapes, illumination and occlusion. In [1], D. G. Lowe proposed SIFT, a feature composed of key-points, which is stable and invariant. Afterwards, S. Romberg et al. used SIFT key-points and corresponding representations to recognize logos in images [2]. Similarly, in [3], they further extracted SIFT features from different regions by bundle min-hashing and formed ultimate representations to learn graphic logos.

Since Convolutional Neural Networks (CNNs) achieved significant success on the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [4], CNNs have received attention again and acquired explosive progress. So far, several excellent detection frameworks with CNNs are proposed, followed by plenty of literatures and advances. R-CNN [5] used CNNs to recognize candidate proposals and to regress the bounding box coordinates. Fast R-CNN (FRCN) [6] improved the feature extraction process with an input image learned only once. Faster R-CNN [7] generated region proposals by CNNs and realized the end-to-end training. Obviously we can apply general object detection framework to logo detection. Previous efforts in [8–12] utilized Recursive Neural Networks (RNNs), R-CNN, FRCN or Faster R-CNN and achieved improvements on detection performance.

To our best knowledge, the existing public logo datasets are small. One of the available logo datasets is "FlickrLogos-32" [13], consisting of only 32 classes, 70 images for each class, 5644 logo objects, and 6000 non-logo images. Some previous literatures made efforts on data augmentation. Hoi et al. [10] created two bigger logo datasets called Logos-18 and Logos-160. However these two datasets are not public yet. In [12], data augmentation was made by synthesising context with logo templates. This method can expand data scale conveniently but seemed to make few contributions to detection performance.

CNNs and deep learning method have achieved remarkable success in both image classification and object detection. Recently some researches applied CNNs to logo detection and obtained better performance. In [9], they used Fast R-CNN with CaffeNet [14] and VGG_CNN_M_1024 [15] on FlickrLogo-32 dataset. The framework improved mAP from 0.568 to 0.7347, which showed the effectiveness of CNNs and Fast R-CNN. In [8, 11], RNNs and R-CNN were used for logo detection task. All the methods above implied CNNs' powerful representational capacity in logo detection.

In this paper, our main contributions are as follows:

1. To make reliable data augmentation, we manually collected and annotated 6,400 images and mixed them with FlickrLogo-32 dataset, forming a larger dataset.

978-1-5386-0462-5/17/$31.00 ©2017 IEEE.

2. We constructed Faster region-based convolutional networks with different structures, in which transfer learning is adopted.

3. To make our frameworks more suitable to the logo detection task, we cluster the training data by K-means and set proper hyper-parameters. The proposed method can outperform the state-of-the-art results.

II. Proposed Method

This section mainly presents our logo detection framework, which is established on the basis of Faster R-CNN. To break through the bottleneck of data scale, we manually collect and annotate new images from the Internet. Finally, we explore network optimization strategies to achieve better detection performance.

A. FlickrLogo-32 Datasets and Augmentation

FlickrLogo-32 dataset contains 8240 natural scene images, including 32 classes, 70 images for each class and 6000 non-logo images. To augment the data scale for deep learning, in [12], Hang Su et al. expanded FlickrLogo-32 by randomly adding pre-processed logo templates on these non-logo subsets of the dataset. However, experiments showed that this work can hardly make any promotion, since new added objects may be not exactly matching with surrounding contexts.

To keep the same data distribution with FlickrLogo-32, we firstly analyze the original data to confirm collection and annotation rules. To avoid image repetition, we check duplication among new collections. Ultimately, 200 new images for each class are mixed with the original FlickrLogo-32, thus forming a new dataset, called Logo32-270, which contains 270 samples for each class, 8640 images in all.

B. Logo Detection Based on Faster-RCNN

Faster R-CNN can be used in many detection tasks with special modifications. This detector consists of two stages, Region Proposal Network (RPN) and Fast R-CNN. RPN stage proposes object candidates by CNNs. Fast R-CNN extracts feature for each proposal by RoI-Pooling and realizes classification and bounding box regression. The two stages have a shared backbone, whose parameters are usually initialized by pre-trained classification models. This strategy facilitates end-to-end training and faster convergence.

In this paper, we construct Faster R-CNN frameworks with different backbones, as shown in Figure 1. In [9], the classification network CaffeNet and VGG_CNN_M_1024 were used with Fast R-CNN. In [10], CaffeNet, VGG_CNN_M_1024 and VGG16 [15] played the role of shared partition. To seek the best performed structure, apart from the previous networks, we take both ZF [16] and ResNet [17] into account. All designed frameworks are tested on both original FlickrLogo-32 dataset and augmented Logo32-270.

C. Network Optimization

1) Transfer Learning: In general, the shared convolutional layers extract image features from multiple levels. In Faster R-CNN, early layers are fine-tuned on classification models. Due to the lack of data scale of FlickrLogo-32, training the model from scratch may cause over-fitting. Therefore, we apply transfer learning and use pre-trained models on 1k-class ImageNet dataset to initialize the filters, which have been well learned on complicated images and hold better representational capacity.

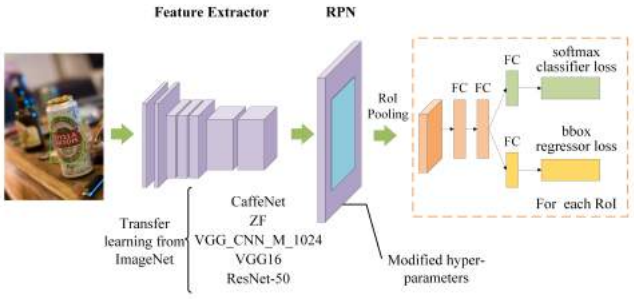

Fig. 1. Faster R-CNN framework with different backbones

2) Clustering by K-means: As we know, CNN models contain many hyper-parameters, which are not learnable during training and rely on manual setting. In Faster-RCNN, how to choose proper anchors is a key problem. Empirically, scales and ratios of candidate bounding boxes should fit most instances in the dataset. According to this assumption, K-means is used to choose anchors. Firstly, we consider the size of instances in the training set to generate four clustered centers, which are used to pre-set anchor sizes. Then we cluster ratios of them in the same way and obtain four pre-setting ratios. Experimental results imply that this modification is helpful and can improve the detection accuracy in an obvious margin.

III. Experiments

In order to verify the performance of our networks and modifications, we conducted a set of experiments on both FlickrLogo-32 and Logo32-270.

A. Experimental Setup

1) Preparation: FlickrLogo-32 is used as the basic dataset. To make a simple augmentation, we horizontally flip the training images to double the amount. Our experiments are conducted on NVIDIA GTX TITAN X GPUs. Every single Faster R-CNN framework is trained end to end for 50k iterations with a learning rate of 0.001 and for next 20k iterations with 0.0001. The momentum is 0.9 and the weight decay is 0.005.

2) Baseline Results: We set all the experimental results on FlickrLogo-32 as baselines for latter modifications, shown in Table I. Obviously, the result of VGG16 is the best due to its complexity and powerful presentation ability. It is unexpected that ResNet-50 gets the last rank, where the data scale is too small to fit the complicated ResNet-50 structure. Therefore, in the later experiments, ResNet-50 is not used. The mAP's changing curves are demonstrated in Figure 2. Although all

TABLE I
FLICKRLOGO-32 BASELINE RESULTS

Backbone	mAP / 48k iters	mAP / 70k iters
CaffeNet	**0.786**	0.785
ZF	**0.804**	**0.804**
VGG_CNN_M_1024	**0.807**	0.795
VGG16	**0.835**	0.828
ResNet-50	**0.72**	0.709

networks are trained for 70k iterations, the results show the best performance is at 48k iterations. Therefore, these results are chosen as baselines for further comparison.

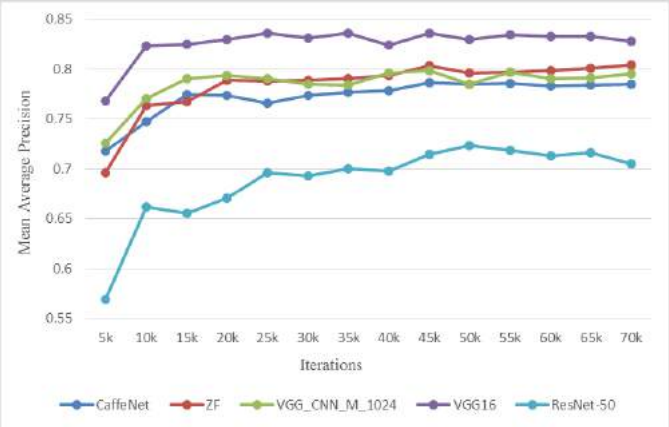

Fig. 2. mAP values versus iterations with five networks

B. Data Augmentation

As described in Section III-A, we collected and annotated 200 new images for each class in FlickrLogo-32 and mixed them with the original dataset. To verify the performance of this augmentation method, we train the new dataset with every structure. FlickrLogo-32 and Logo32-270 share the same test subset, that is, we add all newly collected images into the training set. Thus for each class, there are 240 images in the training set and 30 images for testing. The comparisons are displayed in Table II. For simplification, we mark FlickrLogo-32 as D1 and Logo32-270 as D2.

TABLE II
FLICKRLOGO-32 AND LOGO32-270 RESULTS

Backbone	mAP / D1 (48k iters)	mAP / D2 (48k iters)	mAP / D2 (70k iters)
CaffeNet	0.786	0.845	**0.854**
ZF	0.804	0.848	**0.866**
VGG_CNN_M_1024	0.807	0.853	**0.865**
VGG16	0.835	0.888	**0.893**

Experimental results imply that our data augmentation can improve the detection accuracy significantly, especially at 70k iterations. For simpler networks (such as CaffeNet), augmented dataset can obviously relieve over-fitting (6.8 percent points up).

C. Network Optimization

In Faster R-CNN, RPN generates anchors based on manually given scales and ratios. For example, in PASCAL VOC detection network, anchor scales are [8, 16, 32], and ratios are [0.5, 1, 2]. If the basic anchors match the training set better, RPN can fit the ground-truth more precisely. Anchor settings are strongly relative to data attributes rather than network structure. So we choose ZF as the basic network to explore the most appropriate anchor settings and then apply them to other backbones. Three schemes are used to modify the original anchor settings. The first one named M1 directly adds value 4 to scales list [8, 16, 32]. The second (M2) and the third schemes (M3) mean clustering objects in the training set. The only difference between M2 and M3 is that, in M2 we cluster objects' areas then extract the square root, while in M3 we extract the root of objects' areas then cluster them. Table III shows the different trials on FlickrLogo-32 dataset.

TABLE III
CLUSTERING BY K-MEANS ON FLICKRLOGO-32

Method	scales	ratios	mAP / D1 (48k iters)	mAP / D1 (70k iters)
ZF	[8,16,32]	[0.5,1,2]	0.804	0.804
ZF_M1	[4,8,16,32]	[0.5,1,2]	**0.810** ↑	0.804
ZF_M2	[9,23,36]	[0.5,1,1.8]	0.788 ↓	0.796 ↓
ZF_M3	[5,14,29]	[0.5,1,1.8]	0.797 ↓	0.801 ↓
ZF_M3	[4,12,20,32]	[0.5,1,1.6,2.8]	**0.810** ↑	**0.810** ↑

As shown in Table III, M3 and M1 can both improve mAP at 48k iterations. However, M3 is comparatively more effective and stable since the scales and ratios are clustered from practical training set rather than obtained by simply adding an extra anchor scale as in M1. Furthermore, we test M3 on other networks to verify its effectiveness, which are presented in Table IV.

TABLE IV
M3 RESULTS ON OTHER BACKBONES

Backbones	mAP / D1 (48k iters)	mAP / D1_M3 (48k iters)	mAP / D1 (70k iters)	mAP / D1_M3 (70k iters)
CaffeNet	0.786	**0.792**	0.785	**0.793**
VGG_CNN_M_1024	**0.807**	0.802	0.795	**0.801**
VGG16	0.835	**0.845**	0.828	**0.837**

M3 clustering method can improve detection performance except one miss (VGG_CNN_M_1024 at 48k iterations). Experimental resuts show that better pre-set hyper-parameters can bring stable improvement to object generation and regression.

D. Comparison and Discussion

In this part, we merge two innovations in Section III-B and III-C on the best performed VGG16 network, and compare them with the state-of-the-art results, shown as Table V. It is obvious that data augmentation is the most effective way to improve the detection accuracy. VGG16 with augmentation

and clustering pre-settings gives the best results. Some results are listed in Figure 3.

TABLE V
OUR RESULTS VS PREVIOUS WORKS

Method			mAP
Bag of Words (BoW) [3]			0.545
Bundle of Min Hashing [3]			0.568
BD-FRCN-M_2 [9]			0.7347
BD-FRCN-M_1 [9]			0.7314
Deep Logo [18]			0.744
RealImg [12]			0.811
Ours	augmentation	clustering	
VGG16_D1	×	×	0.8353
VGG16_D1_M3	×	√	**0.8447**
CaffeNet_D2	√	×	0.8545
VGG_CNN_M_1024	×	√	0.8647
ZF_D2	√	×	0.8662
VGG16_D2	√	×	0.8927
VGG16_D2_M3	√	√	**0.9035**

Fig. 3. Visualization of some test images by VGG16_D2_M3

IV. CONCLUSION

In this paper, we construct several frameworks for logo detection based on Faster R-CNN, in which VGG16 performs the best. To solve the problem of limited data in FlickrLogo-32, we manually augment the dataset which can be publicly available soon. During the training stage, transfer learning and pre-setting better hyper-parameters are proposed to predict proposals more precisely. In the future, we will explore modification of the RPN structure to generate anchors from multiple level features for various scales. Besides, FlickrLogo-32 dataset provide mask annotations, we can realize detection task with the assist of segmentation information.

ACKNOWLEDGE

This work is supported by Chinese National Natural Science Foundation (61372169, 61532018).

REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
[2] S. Romberg, L. G. Pueyo, R. Lienhart, and R. Van Zwol, "Scalable logo recognition in real-world images," in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*. ACM, 2011, p. 25.
[3] S. Romberg and R. Lienhart, "Bundle min-hashing," *International Journal of Multimedia Information Retrieval*, vol. 2, no. 4, pp. 243–259, 2013.
[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
[6] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.
[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
[8] E. Francesconi, P. Frasconi, M. Gori, S. Marinai, J. Sheng, G. Soda, and A. Sperduti, "Logo recognition by recursive neural networks," in *International Workshop on Graphics Recognition*. Springer, 1997, pp. 104–117.
[9] G. Oliveira, X. Frazão, A. Pimentel, and B. Ribeiro, "Automatic graphic logo detection via fast region-based convolutional networks," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 985–991.
[10] S. C. Hoi, X. Wu, H. Liu, Y. Wu, H. Wang, H. Xue, and Q. Wu, "Logo-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks," *arXiv preprint arXiv:1511.02462*, 2015.
[11] Y. Bao, H. Li, X. Fan, R. Liu, and Q. Jia, "Region-based cnn for logo detection," in *Proceedings of the International Conference on Internet Multimedia Computing and Service*. ACM, 2016, pp. 319–322.
[12] H. Su, X. Zhu, and S. Gong, "Deep learning logo detection with data expansion by synthesising context," in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*. IEEE, 2017, pp. 530–539.
[13] "Dataset: Flickrlogos-32," http://www.multimedia-computing.de/flickrlogos/, accessed August 20, 2015.
[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.
[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[16] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.
[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[18] F. N. Iandola, A. Shen, P. Gao, and K. Keutzer, "Deeplogo: Hitting logo recognition with the deep neural network hammer," *arXiv preprint arXiv:1510.02131*, 2015.