Few-example Logo Detection with Model Refinement

Bing Liu[1], Bing Li[2], Weiming Hu[3] and Jinfeng Yang[1]
[1]*College of Electronic Information and Automation, Civil Aviation University of China, Tianjin, China*
[2]*National Laboratory of Pattern Recognition Institute of Automation, Chinese Academy of Sciences, Beijing, China*

Keywords: Logo detection, Few-example, Three-stage, Model refinement.

Abstract: Logo detection is a laborious but strong practicality task that has a variety of technology applications. Since the fundamental of state-of-the-art detectors, large-scale annotated datasets, is cost-consuming, few-example logo detection is imperative and thought-provoking. In this paper, a three-stage Few-example Logo Detection Refined System (FLDRS) is proposed to detect logo with a few annotated samples. Specifically, the proposed detector is first initialized using large-scale generic target detection dataset with annotations, such as ImageNet, then further updated with large amount of synthetic logo images, and finally refined with a few annotated real examples. To make synthetic data more closer to real scene, a copy-paste-blend strategy is also presented in our model which not only characterizes many kinds of possible logo transformations but also takes the environment attribute of the logo type into consideration. The superior performance in FlickLogo-32 dataset demonstrates the efficiency of the proposed FLDRS.

1 INTRODUCTION

Logo detection is a sub-problem of object detection, which has been extensively researched because of its applications in many fields, such as commercial advertising, recommendation search, intelligent transportation systems and so on.

Previously, most traditional logo detection methods used hand-crafted feature to make the expression. For example, SIFT key points were used to perform the feature expression and recognize the logo image in (Lowe, D. G., 2004; Romberg, S. et al., 2011; Romberg, S. et al., 2013). Recently, deep feature has made great breakthroughs in varied object detection tasks. And Convolutional Neural Networks (CNN) based methods, Fast R-CNN (Girshick, R., 2015), Faster R-CNN (Ren, S. et al., 2015) and SSD (Liu, W. et al., 2016) for example, have gain significantly higher performances on PASCAL VOC benchmark (Everingham, M. et al., 2010) compared to traditional methods. CNN based methods are also studied in logo detection (Iandola, F. N. et al., 2015; Oliveira, G. et al., 2016; Bao, Y. et al., 2016; Li, Y. et al., 2017; Bianco, S. et al.,2017). Different from generic object detection, logo detection faces its unique challenges. First, some logos are too small to detect in the view as shown in Figure 1(a). Second, same logo may be very different for multifarious variations in sizes, colour, rotations, illumination, deformations and even more complex transformations, for example the pop-top cans in Figure 1(b). Third, different logos may be also difficult to distinguish for their similarity, such as the logo ‘Apple’ and ‘BMW’ shown in Figure 1(c) and (d). What’s worse, as the commonly used datasets shown in Table 1, there is no large-scale and completely annotated datasets used for logo detection, which is extremely important for deep learning, cause the annotation is time and energy consuming.

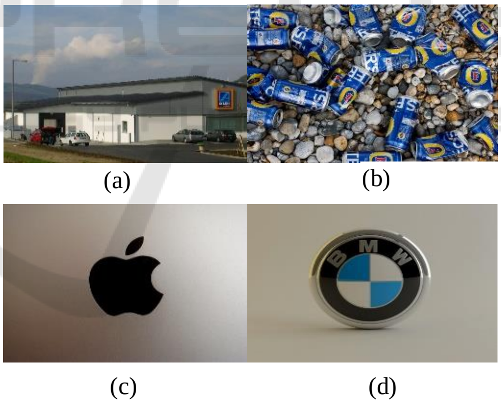

Figure 1: Challenges that logo detection facing.

Motivated by the above challenges, a three-stage few-example logo detection system is proposed in

Liu, B., Li, B., Hu, W. and Yang, J.
Few-example Logo Detection with Model Refinement.
DOI: 10.5220/0008096901220127
In *Proceedings of the International Conference on Advances in Computer Technology, Information Science and Communications (CTISC 2019)*, pages 122-127
ISBN: 978-989-758-357-5

Table 1: Statistics of logo detection datasets.

Dataset	Classes	Images	Published	Completely annotated
FlickrLogo-32 (Romberg, S. et al., 2011)	32	2240	√	√
Logo32-270 (Li, Y. et al, 2017)	32	8640	×	√
BelgaLogos (Joly, A. et al. 2009)	37	1321	√	×
LOGO-NET (Hoi, S. C. et al., 2015)	160	73414	×	√
WebLogo-2M (Su, H. et al., 2017)	194	2190757	√	×

this paper. Specifically, it first learns a model pre-trained on ImageNet from artificial logos, and then refines the model with a few real examples. With this system, few-example logo detection task and deep learning method can be compromised to the great extent. A copy-paste-blend method is also presented to utilize the specific environment attribute of each logo type to generate the artificial logos. Experiments has proved the efficiency of our proposed system.

This paper is organized as follows. Related work is discussed in section 2. The proposed FLDRS and artificial logo generation method are detailed in section 3. Section 4 shows the experimental results and the corresponding discussions. Section 5 concludes the paper.

2 RELATED WORK

In this section we discuss the closely related works in the fields of logo detection, logo datasets and deep learning with one/few shot learning.

2.1 Logo Detection/Logo Datasets

Traditional logo detection methods are established on hand-crafted visual features such as SIFT and HOG. These methods need not training and directly match the template to the query logo. Hand-crafted visual features are not robust to large deformation. Recently, deep features are also used in logo detection and gain significantly promotion. Deep learning inspires large dataset construction (Deng, J. et al, 2009). Though some datasets shown in Table 1 can be used for logo detection, most of them are usually unpublicized or not completely annotated. In this paper, we proposed a unified system to utilize the benefits of deep learning with only a few labelled examples.

2.2 Deep Learning With One/Few Shot Learning

Recently, various methods (Koch, G. et al., 2015;Vinyals, O. et al., 2016;Snell, J. et al., 2017;Sung, F. et al., 2018) have been explored for one/few shot learning to classify objects by metric learning, where only one or few annotated examples are used. Different with classification, detection task needs to localize the objects' location which limits the one/few shot learning methods usage. In this paper, we propose a system which can use only a few examples to detect logo and verifies its efficiency.

3 APPROACH

As shown in Figure 1, the proposed FLDRS consists of three phases: initial model, artificial model, refined model.

3.1 Three-Stage Method

Stage one. A logo detector with the backbone such as Faster R-CNN or SSD is first pre-trained on ImageNet. With the large-scale and annotated dataset, the detector can be well initialized which will be benefit to convergence of next phase. At this phase, the extracted feature is in a coarse-grained level. The model here named initial model.

Stage two. The pre-trained detector is further refined on the constructed artificial logo datasets. The artificial logo datasets are generated as the description in next subsection. Though the artificial logos are

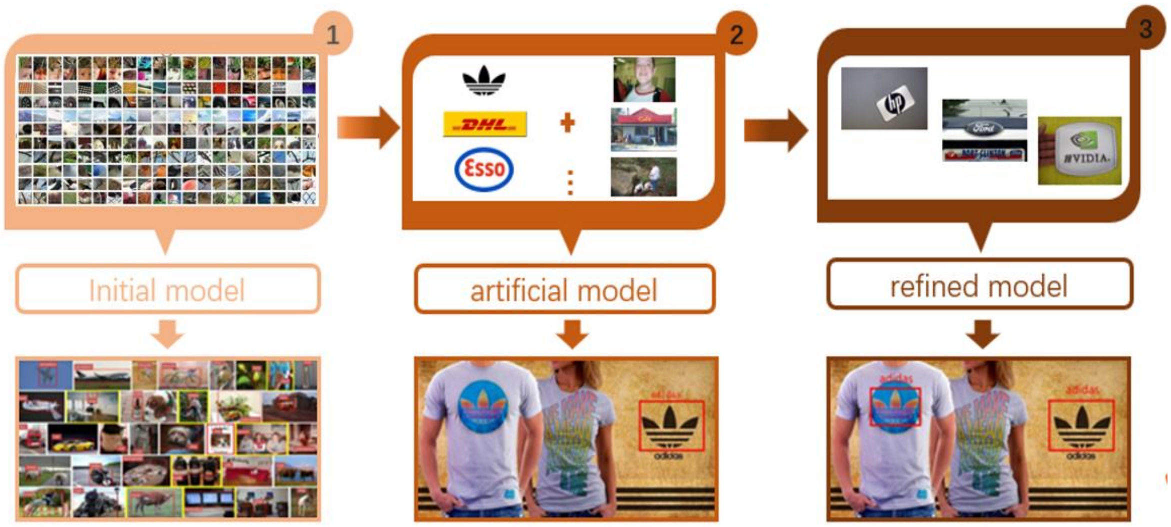

Figure 2: Overview of our FLDRS.

easy samples compared to real logos, which leads to the weakness of the detector to detect real logos, it further close to the optimum weights of ideal model. The model in this phase named artificial model.

Stage three. A few annotated real examples are lastly used to further the model. As the weights of the model is very close to the optimum value, only a few real examples can migrate the weights to the local optimum. The last model named refined model.

In the proposed system, the weights of the logo detector are gradually shifting to the optimum value and only a few annotated examples are used in practical. Note, the cost of ImageNet annotation is not taken into consideration as the datasets is commonly used in deep learning of computer vision for model initialization.

3.2 Artificial model with copy-paste-blend

The artificial model is an important part of our FLDRS. In this Section, we will introduce our approach to generate artificial logo data for training artificial model. The copy-paste-blend method, automatically copying logo templates, pasting them on random scene images and then blending them, is used to generate logo data.

Logo templates images collecting. To generate artificial logo images for a given logo class, we need an template image for each class. Specifically, it's different from (Su, H. et al., 2017), which is using pure logo as the template. We adopt a template selection method based on common context by utilizing the environmental attributes of the logo type, as Figure3 shows the example, logo templates are extracted one part from the FlickrLogos-32 training set which provides the binary segmentation mask rather than using the rigid template image and the other part from prior knowledge about the logo. It is not only a good combination of the inherent environmental attributes, more in line with the actual circumstance, but also can get better blending with scene images.

Transformations of the logo templates. In order to increase the diversity of logo templates, we adopt a variety of image augmentation methods, including geometric transformation, colour transformation and noise disturbance, on the basis of using real logo instances, which already has corresponding transformation of environmental attributes, to make them more consistent with various circumstances in the real world.

Scene images collecting. To collect scene images, we select high quality scene images from Google website by the keywords “indoor”, “outdoor”, “people”, “sports”, “party” and so on. These images contain various places where logos may appear, so the artificial images have a great diversity. Besides, for accuracy of the training model, we exclude images where is no logo in the background.

Generating artificial logo images with blending. Given the logo template image transformations described above, we generate a number of variations for each logo including every transformation, and then utilise them to make artificial logo images in the scene images by cover a logo template in the previous step at a random location with the Poison Blending (Pérez, P., et al, 2003)). The blending step smoothens out the boundary artificial logo images between the

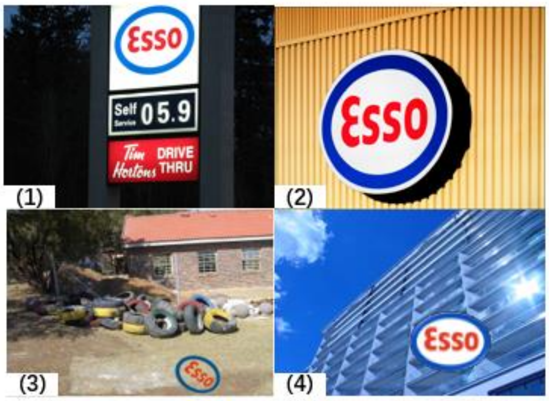

Figure 3: (1)(2) shows the "esso" logo in real world; (3) shows the generation result of the method (Romberg, S. et al., 2011), and (4) is ours. Ours can be closer to the real images.

logo templates and the scene images, which can reduce the edge information that interferes with the model training. Figure 4 shows two blending modes. Although it is not visually perfect results, they improve performance reliably of the model training .

4 EXPRIMENTS

In our experiments, both the training and testing processes are carried out on a TITAN Xp GPU.

Dataset. Most datasets have not been completely annotated or published, as the Table 1 shows. With that in mind, we utilize FlickrLogo-32(Romberg, S. et al., 2011) as our experiment dataset, which is the priority of many researchers. It contains 32 classes logo images and each class has 10 training images which is very suitable for our task for few training examples. The results are based on the artificial logo number which is set to 100 and the division of the training set/testing set which is set by 10/60 during the training refined model.

Evaluation. For the quantitative performance measure of logo detection, we utilize the mean Average Precision (mAP) for all logo classes. A detection is considered being correct when the Intersection over Union (IoU) between the predicted and groundtruth exceeds 50%.

Model and training. In FLDRS, We can nest any detection model within it. And we compare the proposed FLDRS with three state-of-the-art object detection approaches:

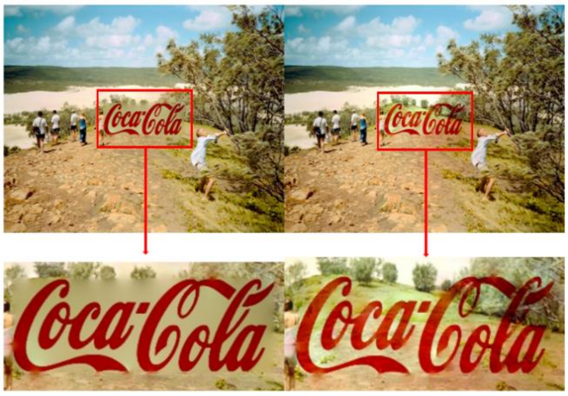

Figure 4: Different blending modes used during generating artificial logo images.

(1) Faster R-CNN (Ren, S. et al., 2015): A state-of -the-art detector which detects objects in two stages.

The first stage is a region proposal network (RPN) that generates candidates and then carries out more refined classification and regression process. The architecture is trained in an end-to-end fashion using a muti-task loss. For training, we use SGD+ momentum with a value of 0.9, and learning rate of 0.001 for training artificial model with 30000 iterations. And then we reduce the learning rate to 0.0003 for training the refined model with 60000 iterations.

(2) SSD (Liu, W. et al., 2016): A state-of-the-art regression based detection model. The networks are trained for 60,000 iterations using Nesterov Optimizer for the artificial model training, and then in the last phase we use 90,000 iterations to train the final model.

(3) YOLO v2 (Redmon, J. et al., 2017): Another bounding box regression based multi-class detection model. For training the artificial model, the learning rate during 0-1000 iterations we set is 0.001, then the learning rate during 1000-25,000 iterations is 10 times of the original learning rate of the 0.01, then the learning rate decreases successively after a certain number of iterations as the multiplier of 0.1. For training the refined model, the set-up is same as the Faster R-CNN.

Results. Table 1 compares the performance of the three state-of-the-art detectors when trained our FLDRS on FlickrLogo-32 dataset. It can be found that FLDRS performs better than the results in one phase that using only 10 logo images. And Faster R-CNN performs better than the others which also verifies that two-stage detector is usually slightly better than the one-stage ones. This is consistent to the original finding that one-stage detector is sensitive to the size of objects and performs worse on small targets which appear frequently in the FlickrLogo-32 dataset.

Table 2: Comparison of the detectors trained on FlickrLogo-32 dataset in our system. (evaluation: mAP).

Method	Training setting	
	Initial (10 real)	FLDRS (10real+100artific -ial)
Faster R-CNN	50.3	**57.8**
SSD	51.2	55.2
YOLO v2	50.0	54.0
(Su, H., et al. 2017)	50.4	55.9

The mAP's changing curves during different iterations are demonstrated in Figure 5. We can see the fastest rate of change of most curves is 10,000-20,000 iterations. Our model also changes rapidly between 30,000-40,000 iterations, because during training the refined model, real logo images are added later. And we can see blending strategy has a bit improvement for the performance.

The impact of different templates in artificial model on performance is showing in the Table 3.

Table 3: The impact of different templates.

Template Change	Pure	Pixel level	Ours
mAP	55.9	56.3	**57.8**

Table 4: The impact of different transformations (Geo: geometric).

Trans	Geo	Color	Noise	None	All
mAP	56.5	54.5	52.6	53.1	**57.8**

In this experiment, we use Faster R-CNN as our basic detection framework, 10 real world logo images and 100 artificial logo images as the training examples. We can see that our template selection method performs the best than the other two. Because Our method combines the advantages of the other two and fully utilizes the environmental attributes of the logo.

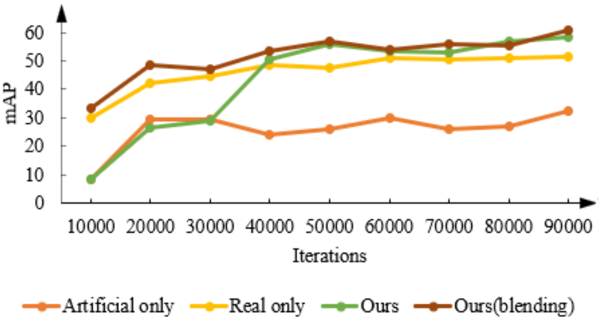

Figure 5: mAP values during different iterations.

In Table 4 we evaluate the performance about the several improvements in the phase of the data generation of the artificial model. As the Table 4 shows, each of these transformations has a bit improvement, and geometric changes lead to the greatest improvement (+3.4%). We think it is because the geometric transformations of logos are the most common in the real world. On the contrary, noise disturbance plays a slight negative role (-0.5%).

5 CONCLUSION

In this paper, we propose a three-stage Few-example Logo Detection Refined System (FLDRS) to detect logo with a few labelled examples. In the system, a method used for generating artificial logos taking both logo transformations and environment attribute of specific logo type into consideration is presented. In addition, with the blending strategy the generated artificial data is closer to real logo compared to other approaches and improves the logo detectors. We demonstrate the conspicuous effect of our proposed system with the excellent performance even only a few annotated examples are available. More logo detection methods based on limited examples will be further studied in the future considering the importance of this task and we hope that more scholars can participate in this study in the era of big data.

REFERENCES

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110.

Romberg, S., Pueyo, L. G., Lienhart, R., & Van Zwol, R. (2011, April). Scalable logo recognition in real-world images. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval* (p. 25). ACM.

Romberg, S., & Lienhart, R. (2013, April). Bundle min-hashing for logo recognition. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval* (pp. 113-120). ACM.

Iandola, F. N., Shen, A., Gao, P., & Keutzer, K. (2015). Deeplogo: Hitting logo recognition with the deep neural network hammer. *arXiv preprint* arXiv:1510.02131.

Oliveira, G., Frazão, X., Pimentel, A., & Ribeiro, B. (2016, July). Automatic graphic logo detection via fast region-based convolutional networks. In *2016 International Joint Conference on Neural Networks (IJCNN)* (pp. 985-991). IEEE.

Bao, Y., Li, H., Fan, X., Liu, R., & Jia, Q. (2016, August). Region-based CNN for logo detection. In Proceedings of the International *Conference on Internet Multimedia Computing & Service* (pp. 319-322). ACM.

Bianco, S., Buzzelli, M., Mazzini, D., & Schettini, R. (2017). Deep learning for logo recognition. *Neurocomputing*, 245, 23-30.

Li, Y., Shi, Q., Deng, J., & Su, F. (2017, December). Graphic logo detection with deep region-based convolutional networks. In *2017 IEEE Visual Communications & Image Processing (VCIP)* (pp. 1-4). IEEE.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2), 303-338.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91-99).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision & pattern recognition* (pp. 7263-7271).

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database.

Joly, A., & Buisson, O. (2009, October). Logo retrieval with a contrario visual query expansion. In *Proceedings of the 17th ACM international conference on Multimedia* (pp. 581-584). ACM.

Hoi, S. C., Wu, X., Liu, H., Wu, Y., Wang, H., Xue, H., & Wu, Q. (2015). Logo-net: Large-scale deep logo detection & br& recognition with deep region-based convolutional networks. *arXiv preprint* arXiv:1511.02462.

Su, H., Gong, S., & Zhu, X. (2017). Weblogo-2m: Scalable logo detection by deep learning from the web. *In Proceedings of the IEEE International Conference on Computer Vision* (pp. 270-279).

Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop* (Vol. 2).

Vinyals, O., Blundell, C., Lillicrap, T., & Wierstra, D. (2016). Matching networks for one shot learning. In *Advances in neural information processing systems* (pp. 3630-3638).

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems* (pp. 4077-4087).

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., & Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition* (pp. 1199-1208).

Su, H., Zhu, X., & Gong, S. (2017, March). Deep learning logo detection with data expansion by synthesising context. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 530-539). IEEE.

Pérez, P., Gangnet, M., & Blake, A. (2003). Poisson image editing. ACM Transactions on graphics (TOG), 22(3), 313-318.