Robust Logo Detection in E-Commerce Images by Data Augmentation

Hang Chen*, Xiao Li*, Zefan Wang*, Xiaolin Hu†
State Key Laboratory of Intelligent Technology and Systems, THU-Bosch JCML Center, BNRist, Institute for AI,
Department of Computer Science and Technology, Tsinghua University
Beijing, China
{chenhang20,lixiao20,wang-zf20}@mails.tsinghua.edu.cn,xlhu@tsinghua.edu.cn

ABSTRACT

Logo detection is an important task in the intellectual property protection in e-commerce. In the paper, we introduce our solution for the *ACM MM2021 Robust Logo Detection Grand Challenge*. The competition requires the detection of logos (515 categories) in e-commerce images, which has challenges such as long-tail distribution, small objects, and different types of noises. To overcome these challenges, we built a highly optimized and robust detector. We first tested many effective techniques for general object detection and then focused on data augmentation. We found that data augmentation was effective in improving the performance and robustness of logo detection. Based on the combination of these techniques, we achieved rank of 5/36489 in the competition. We achieved APs of 64.6% and 61.3% on the clean and noisy datasets respectively, which were improved by 8.1% and 19.5% relative to the official baseline.

CCS CONCEPTS

• **Computing methodologies** → **Object detection**.

KEYWORDS

Logo detection; Object detection; Data augmentation

ACM Reference Format:
Hang Chen*, Xiao Li*, Zefan Wang*, Xiaolin Hu†. 2021. Robust Logo Detection in E-Commerce Images by Data Augmentation. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3474085.3479227

1 INTRODUCTION

Logo detection is often required in intellectual property protection, especially in the field of e-commerce. It is a task of object detection in an specific area, which requires the location and recognition of logos. Through the detection of the logo, it is possible to distinguish whether the product is genuine or counterfeit. To advance the development of algorithms on this task, Alibaba proposed the Open Brands dataset [11], which is currently the largest logo detection dataset and contains product images and annotations from e-commerce platforms such as Taobao and Tmall.

The Open Brands dataset was used in *ACM MM2021 Robust Logo Detection Grand Challenge*. The competition was divided into two rounds. The preliminary round used clean images as the test set, while the final round used noisy images (Figure. 1). The same 584,920 images containing 1,303,563 instances were used as the training set in both rounds. Compared with the general object detection datasets such as MS COCO [14], the Open Brands dataset has the following challenges:

- **Noise** The test set in the final round contains different kinds of noises such as corruptions [10], Fourier domain adaption (FDA) [25], style transfer [6] and adversarial noise [21] (see the 2nd row of Figure 1).
- **Long-tail distribution** The number of instances in nearly half of the categories accounts for less than 1/1000, which is seriously unbalanced (Figure 2).
- **Small objects** The average size of the logos is less than 1% of the image size. 94.5% of the logos have a size smaller than 4% of the image.
- **Large scale dataset** The training set contains 585k images, which is about 5 times of the MS COCO dataset and requires much longer training time.

We ranked the fifth among 36489 teams in the competition. Code is available at https://github.com/tinyalpha/MM2021-Robust-Logo-Detection.

2 METHOD

The complete training set requires a lot of training time (e.g. 4 days for training a standard Faster R-CNN [16] with ResNet-50 [9] backbone and 24 epochs schedule on 8× 2080Ti GPU). There is no validation set. For quick verification, we randomly sampled a mini training set and a mini validation set (denoted by `mini-train` and `mini-val` respectively) from the training set. The `mini-train` and `mini-val` had 117k and 24k images respectively. Based on these sampled datasets, we tried the following techniques.

2.1 Techniques for Enhancing Backbone

We first investigated some techniques that had been proven effective in general object detection including Group Normalization [24], Weight Standardization [15], Label Smoothing [20] and GIoU

*Equal contribution.
†Corresponding author.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3479227

Figure 1: Examples of the Open Brands dataset. The first row demonstrates the training examples, with annotations on the top right. The second row demonstrates the testing examples in the final round of the competition, corrupted by different types of noises.

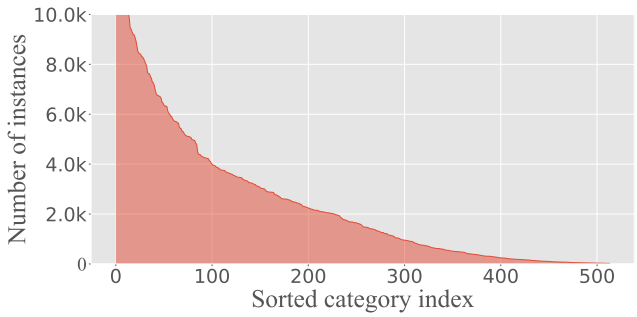

Figure 2: Number of instances of each category.

Loss [17]. Besides, some other time-consuming techniques such as Cascade R-CNN [3], ResNeSt [27] and longer training schedule were also tried. These technologies were mainly used to enhance the backbone network. In most cases, we investigated these techniques individually.

2.2 Category Balance

In order to alleviate the long tail problem, we tried Seesaw Loss [23], BAGS [13] and EQLv2 [22] to balance the classifier, and RFS [8] to increase the probability of rare categories. Due to the class imbalance nature of the Open Brands dataset, rare categories were severely undertrained with vanilla cross-entropy loss. The reason was that the gradients of the rare categories was overwhelmed by the gradient of frequent categories. BAGS reduced the competition between rare and frequent categories by dividing categories into disjoint groups. EQLv2 and Seesaw Loss reweighted the gradients to balance the positive and negative sample gradients of each category. RFS resampled images containing rare categories.

The above balancing techniques were mainly developed based on the LVIS dataset [8]. We adjusted some hyper-parameters based on the differences between the LVIS and Open Brands datasets — we set the threshold of RFS to 0.002 and changed the grouping of BAGS so that the number of categories in each group is similar.

Figure 3: Examples of MixUp, CutOut, and our box-based CopyPaste. MixUp mixed two images. CutOut randomly added black rectangles. CopyPaste randomly pasted logos from another image.

2.3 Test Time Augmentation

We used multi-scale test with flip and soft-nms[1] to further improve the performance. Multi-scale test were only used for multi-scale trained model, and the scales were selected from the range of training.

2.4 Data Augmentation for Clean Images

In addition to the commonly used multi-scale training and random horizontal flipping, we also tried MixUp [26], CopyPaste [7], and CutOut [5]. This was motivated by observing that most of the images contained only one or two instances. Therefore, there were few positive samples in each image, which slowed down the convergence of the model. By introducing more instances implicitly or explicitly, MixUp and CopyPaste alleviated this problem (Figure 3).

However, the original CopyPaste data augmentation does not apply to the Open Brands dataset, because the mask annotations

are unavailable. Here we describe our box-based modification. For each image (denoted by I_A), during CopyPaste, we randomly select another image I_B from the training set. The set of bounding boxes $B = \{(x_1^i, y_1^i, x_2^i, y_2^i)\}_{i=1}^n$ in the image I_B first expands randomly in both up, down, left, and right directions. (x_1^i, y_1^i) and (x_2^i, y_2^i) are the coordinates of the upper left and lower right corners of the box. The expanded bounding boxes set is

$$B' = \{(x_1^i - l^i, y_1^i - t^i, x_2^i + r^i, y_2^i + b^i)\}_{i=1}^n$$

where l, t, r and b are uniformly sampled between 0 and L. In our experiments $L = 15$. Finally, we crop patches from image I_B according to B' and paste them to image I_A. The random expansion is to prevent the detector from overfitting because the box-based cropping will leave apparent boundaries.

When implementing CopyPaste, we used data augmentation such as random scaling for image I_B and repeated the pipeline for image I_A several times.

2.5 Data Augmentation for Noisy Images

In the final round of competition, all models were tested on a noisy test set for benchmarking robustness. We mainly improved the robustness of our model through data augmentation. Specifically, we used the following data augmentation methods:

Corruptions Image corruptions [10] included 15 different types of noise, such as Gaussian Noise, Motion Blur, and Brightness. They originally served as a tool to evaluate the robustness, but we used them as data augmentation. In addition to these standard corruptions, we also used spatter [12] and some color transformations [2].

FDA We used FDA [25] to enhance the model's robustness to frequency domain interference. Similar to CopyPaste, we randomly selected an image from the training set as the FDA's target image.

Style Transfer Style transfer [6] was used to reduce the model's preference for texture features. Since the competition did not allow to use extra data, we used the optimization-based style transfer method [6] and randomly selected style images from the training set. The gradient was calculated based on ImageNet [18] pre-trained VGG-19 [19].

3 EXPERIMENTS

We conducted ablation studies on the sampled `mini-train` and `mini-val` set to verify the above techniques, and applied the useful ones on the entire training set to report the test set performance. For ablation studies, Faster R-CNN with ResNet-50 backbone trained for 12 epochs was chosen as our baseline. Our code was based on the `mmdetection`[4] code base. Unless otherwise specified, all parameters followed their default settings.

3.1 Techniques for Enhancing Backbone

Table 1 shows the results of the experiment. Techniques such as Group Normalization (GN), Weight Standardization (WS), Cascade, ResNeSt, and a long training schedule (2x) helped improve the performance. Label Smoothing (LS), GIoU Loss had a negative influence. Therefore, they were not used in subsequent experiments.

3.2 Category Balance

We studied the effect of category balance techniques such as RFS, Seesaw Loss, BAGS, and EQLv2. We found that they only improved the performance when used alone. For example, RFS and EQLv2 brought improvements of 5.8% and 2.1% in AP respectively, but brought improvement of 5.4% when used together. We attributed this to the difference of distributions between the LVIS and Open Brands datasets.

3.3 Test Time Augmentation

We studied the impact of the scales (*i.e.* short side of the image) in multi-scale test. The results were shown in Table 2. In this experiment, we used a different baseline that integrated some data augmentation techniques. These data augmentations covered the scales in the table. From the table, the improvement of AP_{val} correlated positively with the number of scales. Inference time increased rapidly with the number of scales. Therefore, our final model used a configuration of 7 scales.

3.4 Data Augmentation for Clean Images

We studied the effect of data augmentation techniques including MixUp, CutOut and box-based CopyPaste. We noticed that when data augmentation was used, the model took more time to converge. Therefore, in this experiment, we used a 2x training schedule (*i.e.* 24 epochs) for all models. The results were shown in Table 3. All three data augmentation methods brought significant improvements. It was worth mentioning that our modified box-based CopyPaste further improved AP by 2.5% on the basis of MixUp augmentation.

3.5 Data Augmentation for Noisy Images

To ensure the convergence of the model, we used the same 2x schedule baseline as in the experiment described in Section 3.4. However, for evaluating robustness, our `mini-val` set was no longer applicable because it only contained clean images. Therefore, we still trained the model on the `mini-train` set but reported the results on the noisy test set (denoted by `testB`). This test set was the official test set, which is described in Section 4.

The results are shown in Table 4. These data augmentation methods all improved the performance of the model on `testB`. When combined together, these data augmentation methods improved

Table 1: Effects of techniques for enhancing backbone.

GN	WS	LS	GIoU	Cascade	S50	RFS	2x	AP_{val}
								49.9
✓	✓							51.6
		✓						49.4
			✓					49.2
				✓				55.1
				✓	✓			57.2
						✓		55.7
						✓	✓	56.4

Table 2: Effects of multi-scale test.

scales	AP_{val}
	60.8
640, 800, 960	62.1
640, 720, 800, 960, 1120	62.3
480, 560, 640, 720, 800, 960, 1120	63.0
420, 530, 640, 720, 800, 920, 1040, 1160, 1280	**63.2**

Table 3: Effects of data augmentation for clean images.

MixUp	CutOut	CopyPaste	AP_{val}
			53.6
✓			54.7
	✓		53.9
✓		✓	**57.2**

Table 4: Effects of data augmentation for noisy images.

FDA	Corruptions	Style Transfer	AP_{testB}
			34.92
✓			37.95
	✓		39.93
✓	✓	✓	**42.53**

AP_{testB} by 7.6%, while only increased AP_{val} by 1.1%. Clearly, they significantly improved the robustness of the model.

4 FINAL RESULTS

The competition had two rounds. We term the test set used in the preliminary round as `testA`, and the test set used in the final round as `testB`. The former is a set of clean images with the same distribution as the training set. The latter contains different kinds of noises as shown in the 2nd row of Figure 1. In this experiment, we applied the techniques verified on the `mini-train` to the entire training set and reported the results on `testA` and `testB` respectively. The results on `testA` and `testB` were returned from the competition organizer.

Table 5: Final Results on `testA`. * indicates training 20 epochs on the `trainR`.

TTA	RFS	Cascade	S101	MST	DataAug	AP_{testA}
						56.50
✓						56.82
✓	✓					61.52
✓	✓	✓				63.62
✓	✓	✓	✓			63.86
✓	✓	✓	✓	✓		64.50
	✓	✓		✓	✓	**64.64***

Table 6: Final Results on `testB`. * indicates training 20 epochs on the `trainR`. † indicates the techniques used in the preliminary round, including TTA, RFS, Cascade, MST, and DataAug.

Preliminary†	GN + WS	RobustAug	LongerSchedule	AP_{testB}
				41.80
✓				51.75*
✓	✓			56.01
✓	✓	✓		59.94
✓	✓	✓	✓	**61.26**

We used Faster R-CNN with ResNet-101 backbone as the baseline. The baseline was trained on the full training set, denoted by `trainF`, for 24 epochs (2x schedule). To reduce the training overhead, we also built a training set with only 60% of the images, denoted by `trainR`. The `trainR` was obtained by only downsampled the images belonging to the most frequent categories. Some of the models in this section were trained on the `trainR` and the performance usually differed from trained on the `trainF` by less than 1% AP.

4.1 Preliminary Round

Table 5 shows the results of the preliminary round. TTA in the table means soft-nms and flip augmentations. DataAug means general data augmentation, *i.e.* MixUp, CopyPaste and CutOut. CopyPaste repeated three times, and its probability of selecting images aligned with the RFS. Although we did not use all the techniques due to time constraint, the final model still achieved an AP of 64.54%, which was 8.14% higher than the baseline.

4.2 Final Round

The results of the final round are shown in Table 6. The models in rows 2 and 3 were both trained on the `trainR`, while the other ones were trained on the `trainF`. We added GN, WS, RobustAug and LongerSchedule to our best model of preliminary round. Among them, RobustAug means data augmentations for robustness (*i.e.* Corruptions, FDA, and Style Transfer). The LongerSchedule indicates an extension of 7 epochs based on the 2x training schedule. In addition, for TTA, we added a 7-scale multi-scale test. With strong data and test time augmentation, our model achieved AP of 56.01% (+12.21% AP compared with the baseline).

5 CONCLUSION

In this work, we built a highly optimized and robust detector through data augmentation and other techniques. We achieved APs of 64.64% and 56.01% on the `testA` and `testB` sets respectively, and ranked the fifth among 36489 teams. We hope that our work can promote the protection of intellectual property rights in e-commerce.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Nos. 61836014, U19B2034) and THU-Bosch JCML center.

REFERENCES

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. 2017. Soft-NMS – Improving Object Detection With One Line of Code. arXiv:1704.04503 [cs.CV]

[2] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. 2020. Albumentations: Fast and Flexible Image Augmentations. *Information* 11, 2 (2020). https://doi.org/10.3390/info11020125

[3] Zhaowei Cai and Nuno Vasconcelos. 2018. Cascade R-CNN: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6154–6162.

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. 2019. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155* (2019).

[5] Terrance DeVries and Graham W Taylor. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552* (2017).

[6] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, NV, USA, 2414–2423. https://doi.org/10.1109/CVPR.2016.265

[7] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. 2020. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation. *arXiv preprint arXiv:2012.07177* (2020).

[8] Agrim Gupta, Piotr Dollár, and Ross Girshick. 2019. LVIS: A Dataset for Large Vocabulary Instance Segmentation. arXiv:1908.03195 [cs.CV]

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[10] Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261* (2019).

[11] Xuan Jin, Wei Su, Rong Zhang, Yuan He, and Hui Xue. 2020. The Open Brands Dataset: Unified brand detection and recognition at scale. *arXiv:2012.07350 [cs]* (Dec. 2020). http://arxiv.org/abs/2012.07350 arXiv: 2012.07350.

[12] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. 2020. imgaug. https://github.com/aleju/imgaug. Online; accessed 01-Feb-2020.

[13] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. 2020. Overcoming Classifier Imbalance for Long-tail Object Detection with Balanced Group Softmax. arXiv:2006.10408 [cs.CV]

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*. Springer, 740–755.

[15] Siyuan Qiao, Huiyu Wang, Chenxi Liu, Wei Shen, and Alan Yuille. 2019. Micro-Batch Training with Batch-Channel Normalization and Weight Standardization. *arXiv preprint arXiv:1903.10520* (2019).

[16] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *CoRR* abs/1506.01497 (2015). arXiv:1506.01497 http://arxiv.org/abs/1506.01497

[17] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 658–666.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. 2015. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* 115, 3 (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[19] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1409.1556

[20] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2818–2826.

[21] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[22] Jingru Tan, Xin Lu, Gang Zhang, Changqing Yin, and Quanquan Li. 2020. Equalization Loss v2: A New Gradient Balance Approach for Long-tailed Object Detection. *arXiv preprint arXiv:2012.08548* (2020).

[23] Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. 2021. Seesaw Loss for Long-Tailed Instance Segmentation. arXiv:2008.10032 [cs.CV]

[24] Yuxin Wu and Kaiming He. 2018. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 3–19.

[25] Yanchao Yang and Stefano Soatto. 2020. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4085–4095.

[26] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[27] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. 2020. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955* (2020).