# Open Images V5 Detection Challenge: 5th Place Solution without External Data

Xi Yin, Jianfeng Wang, Lei Zhang
Microsoft Cloud & AI
{xiyin1,jianfw,leizhang}@microsoft.com

## Abstract

*This report describes our solution in the 2019 Open Images Detection Challenge (OID-C). The OID-C dataset is a large-scale object detection dataset with* 1.7*M images and* 12.2*M bounding box annotations of* 500 *classes. Different from other small-scale object detection dataset, OID-C has several unique features including: imbalanced class distribution, incomplete annotations with image-level negative labels, and hierarchical structured classes. These aspects make it challenging to train a robust object detector. Our solution is designed to tackle these challenges. First, we use class-aware sampling to solve the data imbalance problem. Second, we propose a novel soft label propagation to alleviate the incomplete annotation issue. Third, we develop a score aggregation method during inference to take into consideration the class hierarchy. Last but not least, we propose a robust ensemble method including score voting and box voting to fuse the results from multiple models. We have conducted an ablation study to show the effectiveness of each component in the proposed ensemble method. Our solution, without using any external object detection dataset, ranked the* 5*th place in the private leader-board.*

## 1. Introduction

Recent years have seen remarkable progress in object detection, thanks to the development of robust algorithms including two-stage detectors like Fast-RCNN [6] and Faster-RCNN [15], and one-stage detectors like YOLO [14] and SSD [11], and the availability of well-annotated datasets such as PASCAL VOC [4] and MS COCO [10].

However, when such algorithms are applied to real applications at scale, they usually suffer from new challenges as these methods require a huge amount of annotated training data that is usually expensive to acquire in a large scale. Recently, Open Images Detection Challenge (OID-C) dataset [8] is introduced for large-scale object detection. Although OID-C has $6.25\times$ classes of COCO [10] (500 *vs.* 80), the average bounding box annotations per image is only increased from 7.2 to 7.3. In fact, it is very time consuming, or practically impossible to annotate every instance of each class in all images. Therefore, how to handle the incomplete annotation problem is crucial for large-scale object detection.

Our first contribution is to propose a soft label propagation algorithm to generate more bounding boxes for training. First, the fundamental difference to previous label propagation method [13] is the use of "neutral" label, which we define as a bounding box without specific class label. During training, all Region of Interests (RoIs) matched to neutral labels are ignored. Neutral label is used to exclude a region from being sampled as background during training. Second, for the propagated positive labels, we use the confidence score of each box as the loss weight so that the detection results with higher confidences will contribute more. Moreover, our soft label propagation also takes consideration of the image-level negative labels, object-part relation, and the group box priors in OID-C dataset.

Our second contribution includes the proposed score aggregation method during inference. The 500 classes in OID-C are organized in hierarchical structure, which means these classes are not mutually exclusive. In this work, we use Faster R-CNN as our framework and train using softmax with cross-entropy loss for label classification, which we find to be easier to train compared to sigmoid with cross-entropy loss. During inference, to handle the hierarchical structure, we aggregate the confidence scores of each child class to all its ancestors before applying NMS. This operation guarantees that the probability of classifying a RoI into a parent class is always higher than those of its child classes.

Our third contribution is the proposed ensemble method. Although model ensemble has been widely used in previous work [12, 5], it is not clear what is the best way to conduct model ensemble. We propose to take full advantages of the detected bounding box locations and confidence scores to generate more robust results. Specifically, we propose score voting to aggregate the confidences from a set of closely located bounding boxes to the box with the highest confidence, and box voting to fuse the bounding box locations to produce a new bounding box. The proposed score voting and box voting boost the performance significantly.

All training is conducted on the OID-C dataset only without any external object detection dataset. Our solution ranked the 5th place in the private leader-board. We will elaborate the details of the top three contributions in our solution in the following three sections.

## 2. Dataset

There are two major challenges in OID-C dataset. First, the class distribution is imbalanced, *i.e.*, there are many rare classes that are with very limited numbers of bounding boxes. Second, the dataset is sparsely annotated. In our solution, we mainly focus on solving these two challenges for dataset processing. Specifically, we use class-aware sampling to solve the data imbalance problem and develop a novel soft label propagation method to overcome the incomplete label problem.

### 2.1. Class-Aware Sampling

As shown in Figure 1, the original bounding box annotation distribution is imbalanced (red curve). For example, the class with the most number of annotations is "Man" with 1.4M bounding boxes while the class with the least number of annotation is "Pressure Cooker" with only 14 bounding boxes. Training on such an imbalanced distribution will certainly affect the performance on rare classes.

Class-Aware Sampling [5] has been explored in previous challenge solutions. Different from [5] that samples all classes with equal probabilities, we simply duplicate the images with rare classes such that the minimum number of bounding box annotations for each class is at least $N_{min}$. We empirically find that $N_{min} = 2,000$ works well in our experiments. The updated distribution is shown as the green curve in Figure 1. Note that as we duplicate the images, all annotations in the images will be duplicated so the numbers of annotations for other classes are increased as well. We observe an improvement of $\sim 5.0$ points in mAP with this sampling method.

### 2.2. Soft Label Propagation

OID-C dataset has image-level positive and negative labels. On average, there are 2.3 positive labels and 1.1 negative labels per image not considering hierarchical label expansion. The real label for all other unannotated classes are ambiguous. Although these classes are ignored in the evaluation, the impact during training is not negligible. In Faster R-CNN framework, all unannotated regions will be considered as background in RoI sampling, which will lead to false negative RoIs during training.

The incomplete label problem has been studied in previous work. Wu *et al*. [16] proposed soft sampling to decay the weight of each RoI based on the overlap with positive bounding boxes. This approach can reduce the negative impact of false negative RoIs to some extend. However, it will also ignore the rich background information that is useful to learn discriminative features. In contrast, we propose to propagate labels on the training images to generate more bounding boxes from a pre-trained model. We then train a new model with both the original and the propagated labels. Label propagation has been used in [13]. However, our approach is tailored to work with OID-C dataset that has many features like group box, image-level negative labels, and object-part relations. Our method differs to [13] in various aspects as described below.

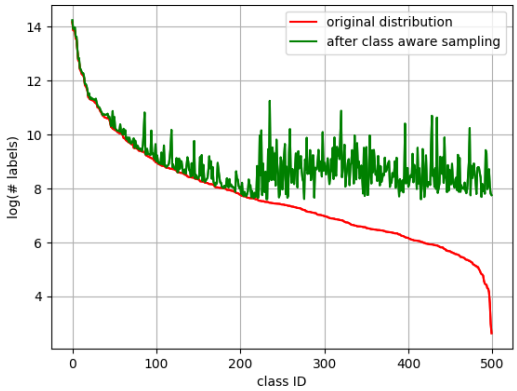


Figure 1. **Dataset label distribution before and after applying class-aware sampling.** X-axis: the class ID sorted by the number of annotated bounding boxes. Y-axis: $\log(\#\text{labels})$.

First, we apply a detection model, trained on the original labels, on the training images. And select all detection results with confidence larger than a pre-defined threshold $t_n$. Second, we remove all detection results with predicted class being labeled as a negative image-level label. These detected boxes are likely to be false positives. Third, we remove a detected box if it is overlapped with any of the annotated box with IoU larger than 0.5. This is to avoid potential conflict in RoI matching. For the remaining detected boxes, we generate new labels by considering the group box property and the object-part relations.

**Group Box** A group box consists of multiple instances of the same class. We can propagate labels given the clue of a group box. Specifically, for each group box in the ground-truth annotations, we select the detected box as a positive instance of that class if the IoA (intersection over the smaller box's area) is larger than 0.5. This can result in multiple instances being propagated based on one group box.

**Object-Part Relation** There are classes with object-part relations as described in [13]. However, most of the part classes are with much less numbers of annotations compared to the object classes. To take advantage of this object-part prior, we select the detected box as a positive part class

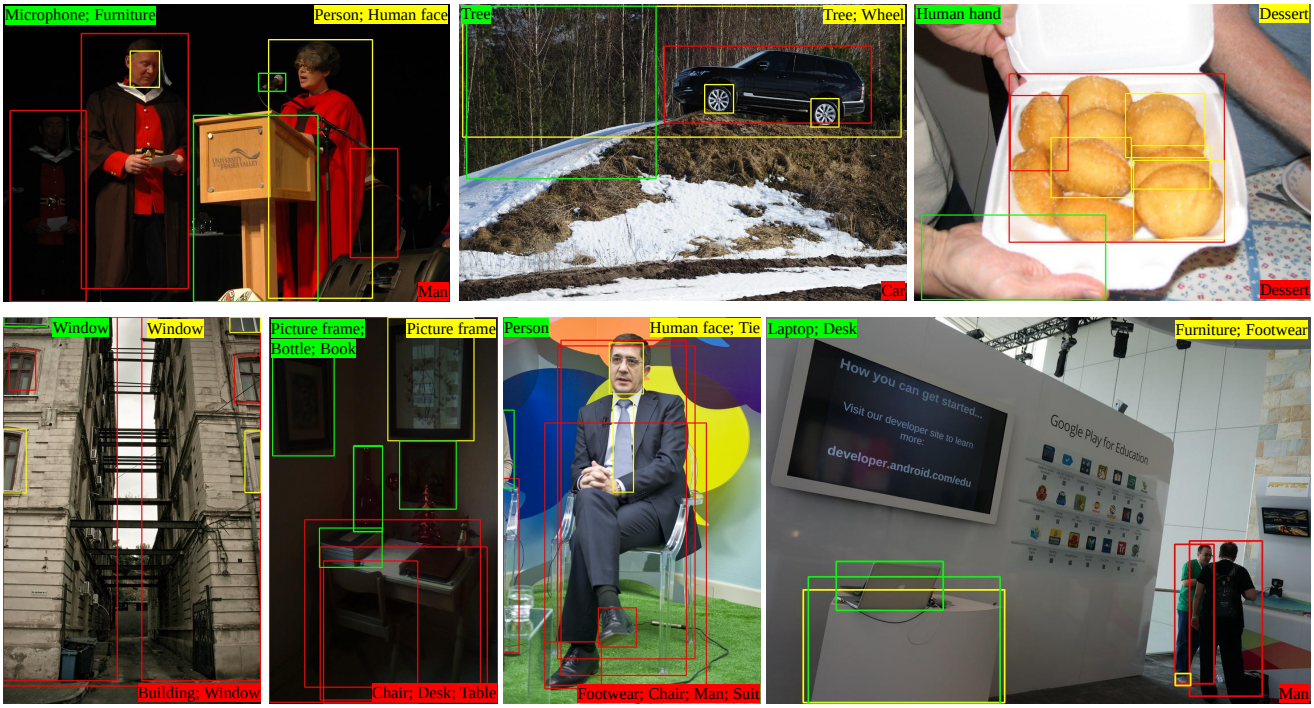


Figure 2. **Soft label propagation results.** red: ground-truth annotations of positive labels; yellow: propagated positive labels; green: propagated neutral labels. The class names are shown in the corner texts with the corresponding colors. Our propagation method can consistently generate valid labels to mitigate the sparse annotation problem. Best viewed in color.

label if the IoA with the object box is larger than $0.5$.

**Soft Label Assignment** We propagate two types of labels: positive labels and *neutral* labels. Positive labels are treated similarly to a ground-truth annotation. we define neutral label as a bounding box without specific class label, which is considered as neither positive nor negative during training. For detected boxes that are selected based on group box or object-part relation, they are considered as positive labels. For the remaining boxes, if its confidence is larger than a pre-defined threshold $t_p$, it is considered as positive. Otherwise, it is treated as a neutral label.

The introduction of neutral label is a fundamental difference compared to prior work [13]. The detection results we are less confident about will become neutral labels. The benefits of using neutral label are in two-folds. First, it can avoid potential false positive from the false detection result if it is considered as positive label. Second, it can avoid potential false negatives if we remove this box during training that makes the region as background.

For propagated positive labels, we take each box's confidence score as the loss weight for the matched RoI samples during training so that the box with higher confidence will have larger contribution in the loss computation. The loss weight for the ground-truth bounding box is $1$. For neutral labels, we will ignore all RoIs that are matched to them. In other words, the loss weight for neutral label is $0$.

Figure 2 shows some example images with our soft label propagation. We observe that our method can consistenly generate high quality labels for second round training. In our experiments, we set $t_n = 0.3$ and $t_p = 0.7$. On average, we have propagated $2.3$ positive labels and $2.0$ neutral labels per image. We observe around $1.0$ point gain in mAP with our proposed soft label propagation. Using the confidence scores of propagated positive labels as the weights in the loss function brings another $0.3$ point gain in mAP.

## 3. Modeling

### 3.1. Framework

We use Faster R-CNN [15] as our framework. Our backbone is based on ResNeXt-152 [17] with Feature Pyramid Network (FPN) [9]. We use class-aware sampling and soft label propagation to process the dataset. We use softmax cross-entropy loss for label classification and smoothed L1 loss for bounding box regression. The main novelty in our work is to use a confidence score for each bounding box to weight the loss of the positive RoIs that are matched to these boxes. This applies to all the loss functions in both stages. All RoIs matched to neutral labels are ignored.

### 3.2. Score Aggregation

The classes in OID is organized in a semantic hierarchical structure. These classes are not mutually exclusive for training softmax with cross-entropy loss. However, we empirically find that softmax with cross-entroy loss is easier to train and performs better than the sigmoid with cross-entropy loss used in [13].

During inference, the detector may not detect the parent and child classes at the same time. Therefore, we should expand the detection results from child classes to all its ancestor classes. We have explored two ways to perform label expansion during inference: before and after NMS.

The first approach is to perform label expansion after NMS. Specifically, we duplicate the bounding box and confidence score from a child class to all its ancestor classes after inference. Another NMS is applied on all 58 parent classes after label expansion in order to remove redundant boxes when the model can detect a similar box of the parent class in the first place. This label expansion and NMS operation can improve the mAP for about 2.2 points.

The second approach is to perform score aggregation before applying NMS. In our work, we select the top $1,000$ RoIs from Region Proposal Network (RPN) for the second stage. These RoIs are used for classification and box regression, after which per-class NMS is applied. We apply a softmax activation function on the class logits from the classification layer, which results in 500 confidence scores of each class for each RoI. For each parent class, we add the confidence scores from all its child classes to its own confidence. The aggregated scores are then used in the NMS operation. We find the score aggregation approach work better than the first approach with an extra 0.2 gain in mAP. We use this approach in our work.

### 3.3. Implementation Details

Our work is implemented in PyTorch and based on the maskrcnn-benchmark repository [12]. We will describe the training and testing details of a single model. Model ensemble will be presented in the next section.

**Training** ImageNet [3] pre-trained models are used to initialize our models. The following setting is used by default unless otherwise specified. The batch size is set to 32 and the model is trained for 800K iterations. We use an initial learning rate of 0.02 and weight decay of 0.00005. Learning rate is divided by 10 at 400K and 750K iterations. The input image is scaled to have minimum and maximum sizes of 800 and $1,333$ pixels. Random flipping is used as data augmentation during training. Images without annotations are removed from training.

**Testing** We evaluate the last checkpoint after training. No test time augmentation is used. The input image is scaled the same way as in training. For each image, we select $1,000$ RoIs from RPN for the second stage. After performing score aggregation and NMS operation (with IoU threshold sets to 0.5 if not specified), we select all detection results with confidence score larger than 0.0001 while limiting the maximum number of detection per image to be 600.

Table 1. **Single model performance (mAP) on the validation set.** All models are trained with ResNeXt-152 backbone and FPN. No test time augmentation is used. Regular NMS with IoU threshold 0.5 is applied during inference.

| model → | X152 | -SE | -DCN | -cascade |
|---|---|---|---|---|
| label expansion | 69.47 | 69.62 | 70.21 | 69.42 |
| score aggregation | 69.59 | 69.66 | **70.38** | 69.50 |

## 4. Model Ensemble

### 4.1. Single Models

We trained different models with ResNeXt-152 backbone and FPN. Besides the regular version ("X152"), we have trained other model variants with SE block [7] ("X152-SE"), DCN block [2] ("X152-DCN"), and cascade R-CNN [1] ("X152-cascade"). As the training for DCN is very slow, the model is only trained for 600K iterations. For cascade R-CNN, we use three stages and the IoU threshold for RoI positive/negative assignment is set to 0.3, 0.4, 0.5 for each stage. The loss weight for each stage is set to 0.5, 0.5 and 1.0 respectively. This is tailored for OID-C dataset that evaluates AP at IoU threshold 0.5.

The performance of these models on the validation set is shown in Table 1. We compare the proposed two different approaches of label expansion and score aggregation to deal with the hierarchical class structure. The score aggregation consistently works better than the label expansion version.

### 4.2. Expert Models

Although class-aware sampling improves the performance of rare classes, the performance on different classes still varies a lot. To improve the performance on classes that have relatively low AP, we train several expert models. Specifically, we select 219 classes where the AP is below average to form a training subset. To improve training speed, we further separate these classes into 20 groups based on the hierarchical structure such that similar classes and all sibling classes are within the same group.

We train one model for each of the 20 groups. X152-DCN is used as the backbone and the model trained on the full dataset is used for fine-tuning. The first three stages of ResNeXt-152 are frozen to avoid over-fitting. The number of classes in each group ranges from 5 to 20. Models are trained for 50K to 200K iterations based on the training set size in each group. The initial learning rate is set to 0.002 and reduced similarly as the training of a single model.

Table 2. **Model ensemble performance comparison on the validation set.** The first row shows the operations applied in order.

| models↓ | NMS | Score Voting | Box Voting | Soft-NMS | mAP @ Val |
|---|---|---|---|---|---|
| single models | $\tau = 0.5$ | NO | NO | NO | 71.95 |
| single models | NO | NO | NO | $\sigma = 0.2$ | 72.10 |
| single models | $\tau = 0.6$ | NO | NO | $\sigma = 0.2$ | 72.27 |
| single models | NO | $\tau = 0.5$ | NO | NO | 73.61 |
| single models | NO | $\tau = 0.6$ | NO | $\sigma = 0.2$ | 74.02 |
| single models | NO | $\tau = 0.6$ | $\tau = 0.6$ | $\sigma = 0.2$ | 74.22 |
| single + expert models | NO | $\tau = 0.6$ | $\tau = 0.6$ | $\sigma = 0.2$ | **75.22** |

For dataset processing, we use class-aware sampling and soft label propagation similar to what we do on the full set. Moreover, as each group only considers a small subset of classes, the bounding box annotations of other classes will not be used. To avoid generating potential false negatives during training, for all classes in each group, we treat the bounding box of their ancestor classes as neutral labels so that the region is ignored from sampling RoIs.

### 4.3. Ensemble Method

For model ensemble, we first add the detection results from all single models and expert models. Then we perform bounding box ensemble for each class separately. We observe that simply applying NMS on top of all detection results gives very limited gain because it cannot take full advantage of all the detection results. As the evaluation metric (AP) is highly dependent on the confidence score (or the order) of the bounding boxes, the key in model ensemble is to explore how to use the bounding box locations and confidence scores to generate new bounding box and/or change the order of the bounding boxes.

To this end, we propose a novel score voting and box voting method to better fuse the bounding box locations and confidence scores. This is based on the assumption that if there are more bounding boxes detected at a similar location from multiple models, the probability that there should be a true detection is higher. Therefore, we propose to refine both the bounding box location and the confidence score based on the IoU between the detected boxes.

**Score Voting** Given a pre-defined IoU threshold $\tau$, conventional NMS will remove all bounding boxes if they are overlapped with a box of higher confidence with IoU larger than $\tau$. In contrast, we use score voting that will accumulate the confidence scores from all removed boxes to the box with the highest confidence. Moreover, the confidence score is weighted by the IoU score.

**Box Voting** Different from conventional NMS and score voting that will remove the bounding boxes with lower confidences, box voting is proposed to fuse the bounding boxes to produce a new box. Starting from the box with the highest confidence, we find a list of bounding boxes with IoU larger than $\tau$ to this box. Then we calculate a weighted average of all the boxes to generate a new box. The weight for each bounding box is the confidence score.

Table 3. **Performance comparison on the test set.**

| team↓ | public | private |
|---|---|---|
| MMfruit | **68.17** | **65.89** |
| imagesearch | 68.16 | 65.34 |
| Prisms | 67.17 | 64.21 |
| PFDet | 65.45 | 62.22 |
| Omni-Detection (Ours) | 63.14 | 60.41 |

Compared to conventional NMS and soft-NMS, our proposed score voting and box voting can take better use of all bounding box information to generate more robust results. In Table 2, we compare the performance on various combination of the proposed voting methods. We have several observations: 1) simply applying NMS operation on all detection results yields 1.6 points gain in mAP compared to the single best model in Table 1. 2) Soft-NMS can improve the performance to some extend by down-weighting the confidence scores of overlapped box with lower confidences. 3) The proposed score voting and box voting can boost the performance significantly, which shows the effectiveness of the proposed voting methods in model ensemble. 4) The expert models can bring 1.0 point gain in mAP. Finally, in our experiments, we apply score voting and box voting at the same time with $\tau = 0.6$ to keep more boxes. Soft-NMS is applied afterwards using the Gaussian-based confidence re-weighting with $\sigma = 0.2$.

Table 3 shows the performance comparison on the test set. Despite that most of the top ranked teams have used external object detection dataset like Object365, our solution ranked the 5th without using any external dataset.

## 5. Conclusions

In this report, we have described the details of our solution to the Open Images V5 Detection Challenge. We have made three major contributions. First, our soft label propagation can alleviate the incomplete annotation problem to a large extend. Second, the score aggregation method can

well handle the hierarchy structured class semantics during inference. Third, we propose a novel and robust model ensemble method with score voting and box voting that can boost the performance significantly. With these approaches, our method ranked the 5th place without using any external object detection dataset.

## References


[1] Z. Cai and N. Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, 2018. 4

[2] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *ICCV*, 2017. 4

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4

[4] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 111(1):98–136, 2015. 1

[5] Y. Gao, X. Bu, Y. Hu, H. Shen, T. Bai, X. Li, and S. Wen. Solution for large-scale hierarchical object detection datasets with incomplete annotation and data imbalance. *arXiv preprint arXiv:1810.06208*, 2018. 1, 2

[6] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 1

[7] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 4

[8] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018. 1

[9] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 1

[12] F. Massa and R. Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018. Accessed: [11/2018]. 1, 4

[13] Y. Niitani, T. Akiba, T. Kerola, T. Ogawa, S. Sano, and S. Suzuki. Sampling techniques for large-scale object detection from sparsely annotated objects. In *CVPR*, 2019. 1, 2, 3, 4

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1

[15] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 3

[16] Z. Wu, N. Bodla, B. Singh, M. Najibi, R. Chellappa, and L. S. Davis. Soft sampling for robust object detection. *arXiv preprint arXiv:1806.06986*, 2018. 2

[17] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 3