Take it cool, then warm up - 4th Place Solution to Open Images 2019, Visual Relationship track

Artur Kuzin
ODS.ai
X5 Retail Group
kuzin.artur@gmail.com

Abstract

This article describes the solution of the 4th place of the Open Images 2019 - Visual Relationship Challenge on Kaggle. Key features of the solution: 1) Subtrain from strong models of Instance Segmentation track 2) Filtering data. 3) Training with partial freezing of layers. 4) Sampling of images and proposals. 5) anokas's solution for triple relationships and average for "is" relationship.

1. Data

To train detector model I used the original dataset with Objects365 as additional external data. As a validation, I selected 3520 samples from validation set. The rest of the validation data I added to the train part. I believe that left too few samples in validation. Since not all improvements in validation correlated with the leaderboard. In addition, the metric is very dependent on the size of the dataset. For train relationship model I used dataset of 2018 year and original dataset of competition.

1.1. Filtering

One of the key features of the dataset is a large number of omissions in annotation. To remove part of poorly annotated samples I select all images where:

1. The number of marked objects above 3
2. There is at least one class object that occurs in the dataset below 1000.

1.2. Class selection for detection

I used 53 leaf classes as target for one group of model and other 4 classes as another target. And trained models for each group of classes.

1.3. Sampling for detection

Each class has the same contribution to the overall speed, so rare classes need to be predicted as well as frequent ones. To implement this, I used two samplers:

1. Image sampler. I implemented the sampler described in the solution of PFDen team for last year Object Detection track [1]
2. The sampler of the proposals. Described in Libra R-CNN paper [4] and implemented in mmdetection framework. [3]

1.4. Augmentation

I used Albumentation [2] library for augmentations. I used for the main part of the training following set:

- RandomBrightnessContrast, p = 0.5
- CLAHE, p = 0.5
- ToGray, p = 0.2
- Cutout, p = 0.9
- JpegCompression, p = 0.4, qualitylower = 70, qualityupper = 99
- RandomRotate90, p = 0.1
- RandomFlip, p = 0.5

2. Detection models

I used Cascade R-CNN models. This models was made from Instance Segmentation track models. All tensors of output layers was remap for new class output. So all layers were initialized from pretrained weighs. For final submission I used X-101-64x4d-FPN (c3-c5) and X-101-32x4d-FPN models.

3. Detection train

To speed up the whole training process, I trained the models defrosting parts sequentially. At first I trained with a frozen backbone until complete convergence. Then sequentially unfreeze groups of convolutions of backbone. Along with this, at each stage I added more samples with noisy annotation, gradually changing $count - threshold$ and $appearance - threshold$. After fully training the model, I turned off augmentations that add bias in the data, froze the backbone and trained only FPN, RPN, and head at maximum batch size (on 32Gb V100). At all stages, an input size of 1024x1024 was used. Along with SGD optimizer with default parameters.

4. Bbox predict

I chose the following values for nms: soft-nms, nms-threshold = 0.75, iou-threshold = 0.5, min-score = 0.0001, score-threshold = 0.0001 through grid search of parameters on validation.

Then I found the issue in the mmdetection github `https://github.com/open-mmlab/mmdetection/issues/300`, where it was recommended to increase the number of bboxes. I set it like this: nms-pre = 12000, nms-post = 2000, max-num = 2000.

For ensemble I used the implementation of Miras Amir implemetation of HTC ensemble from Kaggle imaterialist competition `https://github.com/amirassov/imaterialist/blob/master/mmdetection/mmdet/models/detectors/ensemble_htc.py`. I took two of the best checkpoints of two model (X-101-64x4d-FPN (c3-c5) and X-101-32x4d-FPN), MirrorTTA and 2 scalesTTA.

5. Relationship prediction

5.1. Triple relationship

For this part I implement anokas solution `https://www.kaggle.com/c/google-ai-open-images-visual-relationship-track/discussion/64630`. I took all possible pairs of two objects as a positive class (from vrd-bbox.csv). And all other pair combination of objects as a negative class (from vrd.csv). For each pair I created feature like: mutual distances, areas, angles, intersections. And then I trained LightGBM model through 4folds. For prediction I leave top 80 percent most confident bboxes, due to RAM limit (it takes about 256 Gb of RAM).

5.2. Is relationship

I did not manage to train a separate detector for all attributes. Therefore, I took just the average for each property. I took confidence as of confidence of bbox multiplied by the frequency of occurrence of each attribute for this label.

6. Hardware

- DGX-v1, dual Xeon 2698 v4, 8x V100 32Gb
- dl1: Threadripper 2950x, 3x 1080Ti
- dl2: Xeon 2699 v3, 6x 2080Ti
- dl3: i7 5960x, 3x 2080Ti
- dl4: Threadripper 1950x, 4x 1080Ti
- dl6: Threadripper 1950x, 4x 1080Ti
- nz-1: Threadripper 1950x, 4x 2080Ti

7. Frameworks

- Pytorch 1.1
- mmdetection (with fp16). I started with 0.6.0 version and in the middle of competition switch to 1.0rc2 version
- albumentations

References

[1] Takuya Akiba, Tommi Kerola, Yusuke Niitani, Toru Ogawa, Shotaro Sano, and Shuji Suzuki. PFDet: 2nd Place Solution to Open Images Challenge 2018 Object Detection Track. *arXiv e-prints*, page arXiv:1809.00778, Sep 2018.

[2] Alexander Buslaev, Alex Parinov, Eugene Khvedchenya, Vladimir I. Iglovikov, and Alexandr A. Kalinin. Albumentations: fast and flexible image augmentations. *arXiv e-prints*, page arXiv:1809.06839, Sep 2018.

[3] 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. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv e-prints*, page arXiv:1906.07155, Jun 2019.

[4] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra R-CNN: Towards Balanced Learning for Object Detection. *arXiv e-prints*, page arXiv:1904.02701, Apr 2019.