Enhanced Semi-supervised Learning with RetinaNet Model

Yifan Zhang [1] **Liren Gao** [1] **Anna Xie** [1]

1. Introduction

In computer vision tasks, image classification is one of the most basic tasks. It can not only be used for many real products, such as Google Photo tags and AI content adjustment, but also lay the foundation for many more advanced visual tasks, such as object detection and video understanding. In 2012, Alex Krizhevsky proposed a CNN based model called AlexNet (2), which greatly improved the top-5 accuracy of ImageNet test set from 73.8% to 84.7%. Their method inherits LeNet's multi-layer CNN idea, but greatly increases the scale of CNN. In 2015, a new network called ResNet(1), or residual network, was proposed by a group of Chinese researchers in Microsoft Research Asia. ResNet solves the vanishing gradient problem well. Based on this, RetinaNet was proposed in 2017, which solved the problem of imbalance between positive and negative samples. In our work, we use RetinaNet as our baseline model.

With the rapid development of neural networks, the size of datasets is gradually insufficient to meet the needs of more large-scale network training. The process of dataset annotation is tedious and inefficient, so self-supervised and semi-supervised model training methods emerge spontaneously. In 2021, a semi-supervised method called Unbiased Teacher was proposed(3), which mainly involves the joint training of two progressive models, a teacher and a student, in which the weight of those over confidential pseudo labels was reduced based on the class balance loss. Also in 2021, an unsupervised learning method called DetCo was proposed, which makes full use of the contrast between the global image and the local image patch to learn the discriminant representation of object detection.(4) Here, the contrastive learning method has designed a detection friendly comparative pretext task to learn the large-scale unlabeled data. In our work, we try to use semi-supervised learning to train RetinaNet better.

[1]New York University, Courant Institute of Mathematics, New York, United States. Correspondence to: Yifan Zhang <yifan.zhang@nyu.edu>, Liren Gao <lg3405@nyu.edu>, Anna Xie <awx201@nyu.edu>.

2. Our Approach

In our approach, we train RetinaNet ResNet-50 using 30,000 labeled images and 512,000 unlabeled images. To deal with unlabeled images, we try two classical methods: consistency Regularization and pseudo-label.

2.1. Model Architecture

We use Retina-Network as our model, which is a classic one-stage detector. It has two import characters: Focal loss and Feature Pyramid Network.

2.1.1. Focal Loss

In the training process, the loss weight of positive samples is relatively large, while the loss weight of negative samples is relatively small. However, due to the large number of negative samples, even if the weight is small, a large number of samples will also bring great losses if they are stacked together. It is difficult to optimize to the optimal state in the process of training iteration, so Focal Loss has made improvements on this basis. The focal loss is defined as:

$$\mathrm{FL}\left(p_{\mathrm{t}}\right)=-\alpha_t\left(1-p_{\mathrm{t}}\right)^\gamma \log\left(p_{\mathrm{t}}\right) \tag{1}$$

Focal Loss is the core of Retinanet network. It surpasses two-stage network in precision and one-stage network in speed. With this loss function, this loss function can help us to reduce the negative effect of class-imbalance during training.

2.1.2. Feature Pyramid Network

As shown in 1, Feature Pyramid Network (FPN) is designed with a top-down pathway to enhance the features of different scales by using the rich semantic information contained in the high-level feature map. Therefore, we can use different feature map from FPN to obtain multiple scale targets. Here, we choose Resnet_50 as our backbone network.

2.2. Data Augmentation

Due to the limitation of training set, we used data augmentation to expand our data set. Here, we use the method of image flipping. To put it simply, we utilize image symmetry

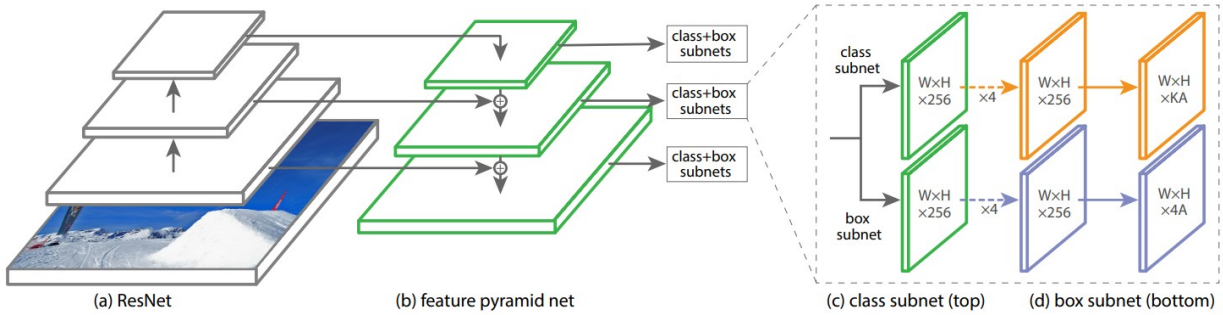

Figure 1. Model Architecture of RetinaNetwork

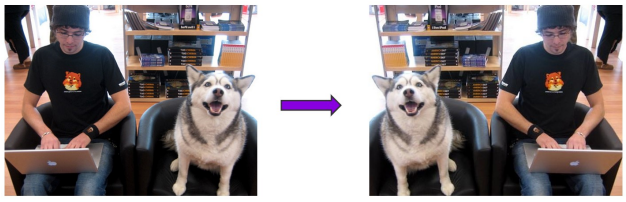

Figure 2. Image Flipping

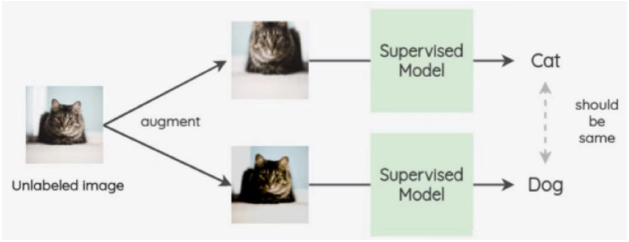

Figure 3. Consistency Regularization

(left and right, up and down) to get more pictures with the same content but different representations. With this method, we can increase the size of the data set on the premise of ensuring the quality of training set labeling. Secondly, we also consider adding some noise to the labeled images, or changing some RGB values for data augmentation. In this way, we can also ensure the consistency between the image content and its label, while changing the value of the data input to expand the dataset. A example of image flipping are show in 2.

2.3. Unlabeled Image

We have 521,000 unlabeled images in training dataset. In order to make better use of these images and realize semi-supervised learning, we try to introduce two different methods to process these images: consistency regularization and pseudo-label.

2.3.1. Consistency Regularization

As shown in 3, The idea used in this method is that the model prediction of the unlabeled image should remain unchanged even after adding noise. For an input, its prediction should be consistent even if it is subject to slight interference. The principle is simple: when the input changes very little, the output of the network should not change. For example, before and after adding noise to an image, the result should not be affected.

For each unlabeled mage, consistency regulation requires an approximate output with noise injected randomly twice. The idea behind this is that if a model is robust, the output should be approximate even if the input is perturbed.

For consistency regulation, how to inject noise and how to calculate consistency are different among different method. The noise can be injected either through the randomness of the model itself (such as dropout) or directly by adding noise (such as Gaussian noise), or through data augmentation. Finally, we can use cross entropy to calculate consistency.

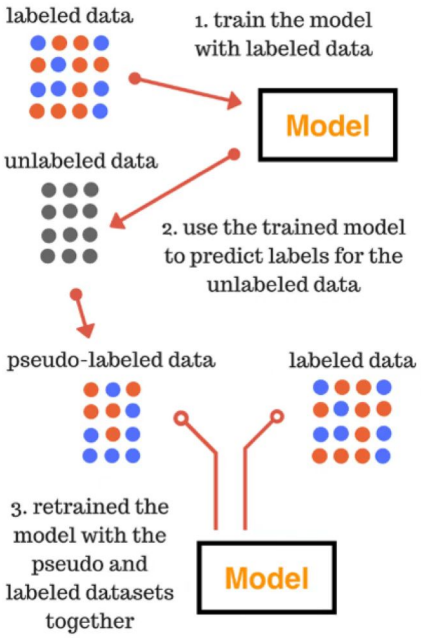

Figure 4. pseudo-label

2.3.2. PSEUDO-LABEL

The second method is called pseudo-label, which is shown in 4. The principle is very simple: we first train a basic model with a small amount of labeled data, then use it to predict a large number of unsupervised data to obtain pseudo label. After that, we use both unlabeled data and labeled to train the model, and then predict a large number of unsupervised data to obtain pseudo label again, and then train the model again. Through continuous circulation, we can finally achieve convergence.

One of the common criticisms of the false labeling method is that no matter whether the false labels on the samples are correct or not, these labels have a high degree of confidence. If a large number of unlabeled samples are labeled incorrectly and used for training, there will be a large number of noise samples in the training set, which will seriously affect the performance of the model.

3. Experimental Result

Due to the time limitation, we only train our model with labeled data. And we are still working on the training process with unlabeled data. Therefore, in this section, we will only show some parts of our result.

For the training process, we firstly resize the dataset images

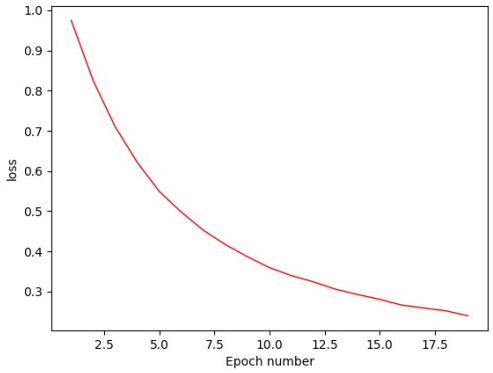

Figure 5. Loss

to a fixed image size, 512*512 pixel. Meanwhile, in order to update each bounding box stay in a correct relative position, we also update the position of the bounding box. As for the learning rate, we are choosing 3e-4 with adam optimizer. Figure 5 is the visualization of our loss value during training process. The loss is keeping decreasing, which demostrates our model works.

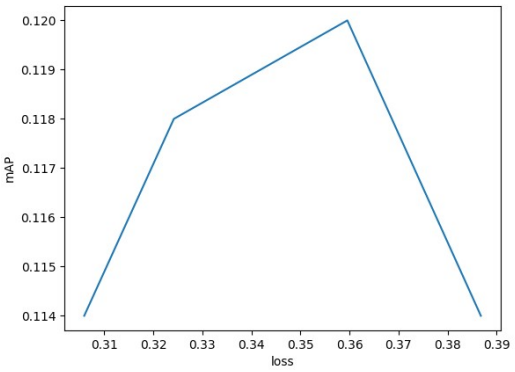

Figure 6. MAP

Figure 6 shows the mean average precision on validation dataset in different epochs. During evaluation, we resize the output regression bounding box back to correct relative location of original image.Also, we iterate the Threshold from upper to lower using 0.2 as step until the output prediction is found in the interval.

When loss is less then 0.36, mAP keeps growing. While aft-

ter loss reaches 0.36, mAP decreases rapidly, which means our model is overfitting on the training dataset and we need to stop the training process. We save the model which has the highest mAP value on validation dataset for futher experimental analysis.

IoU	AREA	MAXDETS	MAP
0.50:0.95	ALL	100	0.120
0.50	ALL	100	0.167
0.75	ALL	100	0.135
0.50:0.95	SMALL	100	0.008
0.50:0.95	MEDIUM	100	0.042
0.50:0.95	LARGE	100	0.140

Table 1. Experimental Results

Tabel 3 shows the final experimental results. We can observe that, under the setting of IoU=0.5, area=all, maxDets=100, our model achieves the best mAP of 0.167.

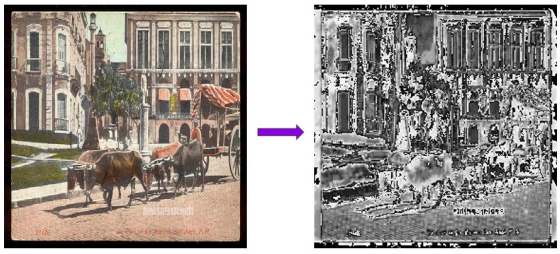

Figure 7. Feature Visualization

4. Visualization and Limitation

We visualize the feature of an input image, which is shown in Figure 7. Although the figure shows that our model owns the ability to capture high-level features, our model still fails to identify the key object in the image.

We are sorry about that we still left a lot of work unfinished. Firstly, we can make improvement on overfitting problem. Currently we have several ideas for this:

- We can use more data augmentation methods to extend the data space on the training set.
- unlabeled data is unused in current model. We can probably generate some pseudo label using that data for training.

And for the model part:

- Feature Pyramid network is still not enough for the small data, as we can see on the table 3 above. We probably need much more things to solve this.
- The design of the threshold is definitely a bug here, in fact only a few bounding box will be returned with our threshold descent algorithm. The threshold needs to be redesigned.
- Finally perhaps we need to give the model more improvements so that it can learn from those unlabeled data.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[3] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection, 2021.

[4] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection, 2021.