Detection experiment report

Name: Wu Jiatai
UID: 3036213445

1 Introduction & background

Artificial intelligence has been a hot topic in recent years due to its unexpected performance to many specific tasks. Computer vision is one of the applications of AI in our daily lives. There are mainly three tasks in computer vision, including classification, segmentation and detection. Detection task is a combination of classification and localization. In this report, we will mainly focus on the detection task.

2. Related works

Thanks to the development of neural networks and GPU resources, there have been many excellent works for the detection task, and most of them are based on networks. In the first, Shift Windows method proposed to use a fix size of box and enumerate all the pixel to test whether it is a box for object. There is no doubt that it is not an efficient method and hence RCNN was proposed. It is a typical two-stage detection method that in the first stage, it first produces region proposal by selective search algorithm and then save the features obtained by CNN. In the second stage, it uses SVM as classifier and NMS to do the prediction. RCNN has put forward a reasonable framework so that many later methods still follow this framework. However, it is still not efficient for every region proposal has to go through the same CNN backbone, which is costly. Then SPP net manipulates the work by first using CNN to extract feature maps and then applied selective search method on the features map. This have saved many costs in convolutional operations. Later on, Fast R-CNN replace the SVM classifier as neuron layer with softmax function, and also provides ROI pooling method on region proposal. So far, methods mentioned above did not fully utilize the networks like selective search method so that it may cause troubles in model training. Faster R-CNN is the first one that was wholly made up of networks. It proposes Region Proposal Network (RPN) to replace the selective search algorithm and anchor box which is used continuously later. Faster R-CNN has reached a high performance and last for years. Finally, YOLO, a typical one-stage method, put forward the idea of grid cell and has a good advantage in fast and efficient detection with a little compromise in performance. In this experiment, I focused on another one-stage detection method Retinanet with high performance.

3. Methods

Retinanet uses ResNet as backbone, FPN to fuse features and two simple convolutional networks as classification and regression heads. The ResNet backbone is the one commonly used with four layers which could be adapted to different depths. In this experiment, I choose to analyze the networks with depth 50 and 101. FPN is a network to fuse different features extracted from different layers of ResNet, aiming to enrich the information of features. This technique is useful to combining features with different scales. FPN outputs five features with different size and then feed all these features into

later heads. Classification and Regression heads have almost the same structure, four convolutional operations with only final output channel dimension differs. These two heads convolute on five features from FPN and then concatenate them in the channel dimension. Regression head will predict four coordinates of each 9 anchors box, and the classification head predicts the logit for each class for corresponding anchor box. Finally, the article has provided a powerful loss function for detection called focal loss. This loss function deals with the imbalance problem of positive and negative samples of detection. Since there are a large number of predesigned anchor boxes and only a small of them are ground truth, then the loss of negative samples will dominate the loss of positive samples. Thus, focal loss controls the negative sample loss by a coefficient and the power of degree.

During my experiment, I have considered three aspects to improve performance on baseline model, and they are learning rate, data augmentation and also short connection in classification and regression heads.

- Learning rate
  I have tried two initial learning rate 1e-4 and 1e-5. And I do not consider using learning rate decay methods.

- Data augmentation
  As we have around two thousand training images but with one thousand and five hundred validation images, it is reasonable to do data augmentation to train such a deep neural network. Originally there is an augmentation method for horizontal flip. To have more various images, three methods are applied including color jitter, adding Gaussian noise and converting to gray scale images with probability 0.3, 0.2 and 0.1 respectively.
  Color jitter changes the brightness and contrast of images. Adding noise is also a commonly used method in augmentation. Converting to gray scale images are also reasonable.

- Short connections in heads
  As inspired by ResNet, short connections could be used to fit the residuals and solve the problem of gradient vanishing and explosion. Then a naïve way to thinking about manipulating the networks of head is to directly add a short connection. This aims to fit the small shift of boxes in regression head.

4. Experiment and analysis

In this section, I am going to show some results of my experiment. I will first go through my experiment process and then provide a summary table for all the results.

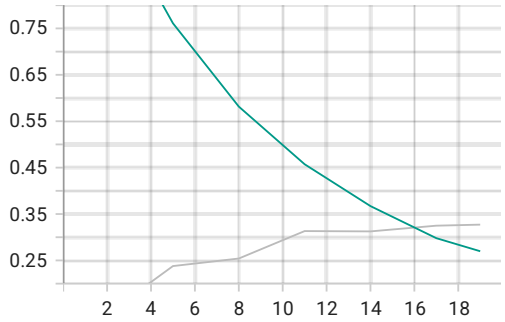

Figure 1 Baseline model

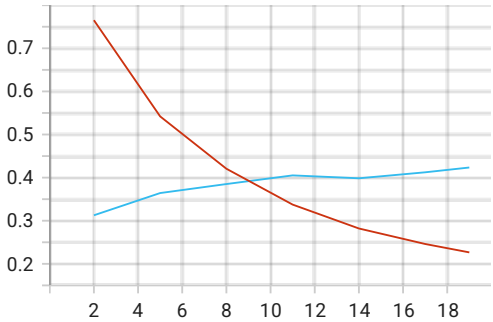

Figure 2 ResNet50 with learning rate 1e-5

In the beginning, train the baseline model with 20 epochs, record the epoch losses and validation mAP for each three epochs, and then visualize the plot as Figure 1. The line decreasing is the training epoch loss and the other one is the validation mAP. It is not meaningful to compare these two curves in one plot, but only for the convenience to discover the trend for each measurement. From the plot, we can see that the performance increase all the time but with slower rate in later epochs, and reaches only 0.32 mAP at maximum. One reason may be the model is underfitting for the validation score hasn't shown an signal to decrease namely overfitting. However, the validation score converging slowly may also result from the learning rate. Then I have changed the learning rate to 1e-5 and do the same experiment. Figure 2 shows the fitting process of the ResNet50 model with learning rate 1e-5. Looking at the validation score, it is obvious that such a learning rate has improved the model performance and the maximum mAP has reached to 0.42. This improvement is significant and thus I applied all the later models with learning rate 1e-5. Directly increase the depth of model from 50 to 101, apply the same learning rate 1e-5, but increase the number of training epochs to 30 due to its larger size. Then I got the excellent model with maximum mAP reaches to 0.4689.

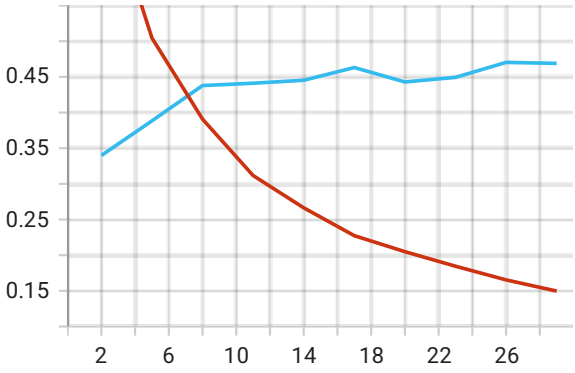

Figure 3 ResNet101 with learning rate 1e-5

After changing the learning rate, I add the augmentation on both ResNet50 and ResNet101, train them both with 30 epochs, and also record results. Figure 4 and 5 show two model results. The trends are quite similar, and the maximum mAP are 0.42 and 0.4639 for ResNet50 and ResNet101 respectively. Unfortunately, it seems that augmentation does not improve the model performance significantly compared to previous model mAP 0.4235 and 0.4689.

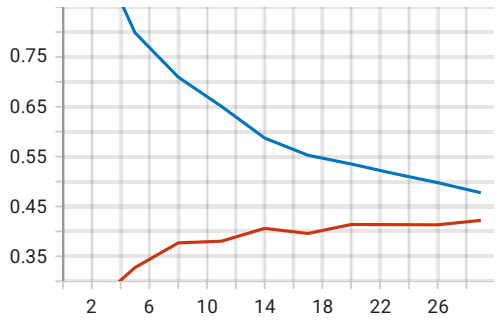

Figure 4 ResNet50 with augmentation

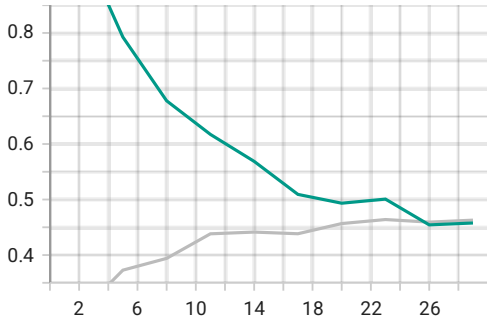

Figure 5 ResNet101 with augmentation

Then finally add the short connections in head, train both models with 30 epochs. Figure 6 and 7 represent the results adding the short connections. It can be seen that both performance are poorer and it seems no further progress could be made by more epochs due to the validation scores have been convergent for a long period. It shows that such kind of manipulation on network does not work well on this task.

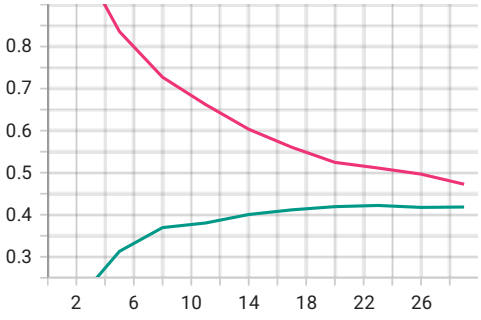

Figure 6 ResNet50 with short connection

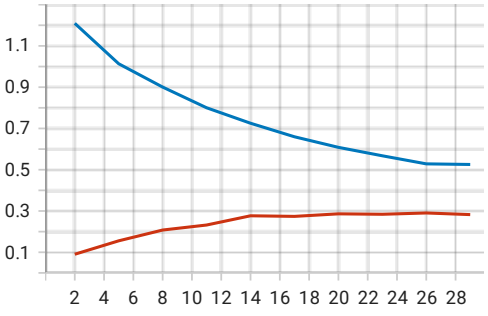

Figure 7 ResNet101 with short connection

The following table summarizes the results of different models with modifications mentioned in section 3. Mean Average Precision is the measurement on validation set. During the training, I will evaluate the model after every three epochs and then regard the optimal mAP as the performance score for that model.

Models	Learning rate	Augmentation	Short connection	mAP
Baseline	1e-4	No	No	0.32
ResNet50	1e-5	No	No	0.42
ResNet50	1e-5	Yes	No	0.42
ResNet50	1e-5	Yes	Yes	0.41
ResNet101	1e-5	No	No	**0.46**
ResNet101	1e-5	Yes	No	0.46
ResNet101	1e-5	Yes	Yes	0.29

On this table, we can conclude that the model with depth 101 and learning rate 1e-5, and without any other manipulation performs best.