Fine-tuning trained models on pronotum datasets

LE Van Linh

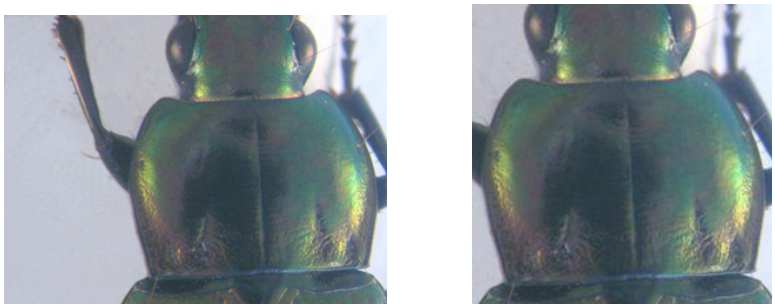
March 30, 2018

Abstract

In this study, we fine-tune on the trained models on pronotum dataset, such as VGG16, VGG19, VGG-CNNS, and ResNet50. Most of the models are Convolutional Neural Networks (CNN) for classification problems. They are trained on ImageNet database. After training, the models are able to classify 1000 classes as the output. To study how does a network which was designed for classification problem work on regression problem, we fine-tune the trained model on pronotum dataset. The experiments will implement on two steps: freeze and unfreeze some layers in trained models. At the end of the experiment, a comparison between the fine-tune losses will be discussed.

1 Dataset

The dataset includes 293 RGB-images of beetle's pronotum. The images were taken by the same camera with the same conditions of resolution of 3264×2448. The images in the dataset were divided into two subsets: training (including 260 images) and testing (including 33 images). For each image, a set of 8 manual landmarks have been set by biologists. Depending the input of the pre-trained models, the images are down-sampled to fit with the models. Firstly, the images are down-sampled to a resolution of 326×245 and the coordinates of the manual landmarks are also re-scaled. Secondly, the centroid point of manual landmarks is calculated for each image. The centroid point is considered as the center of the new image that has been cropped from the down-sampled image. The size of the new image is depending on the input of the trained model that we would like to fine tune. Fig. 1 presents an example in dataset after down-sampling and crop the image to fit with the input size of neural network.



(a) (b)

Figure 1: An example in dataset. *a)* presents the image after down-sampling. *b)* presents the cropped image from down-sampling image which used as the input of CNN

2 The models

2.1 VGGs models

The models are the improved versions of the models used by the VGG team in ILSVRC-2014 competition [1]. The models were designed to evaluate the depth of the network by using an architecture with very small (3×3) convolution filters and pushing the depth to $16 \rightarrow 19$ weight layers. Table. shows the architecture of the VGGs models.

Layer	$VGG-16$	$VGG-19$	$VGG-CNNS$
0	Input(3,224,224)	Input(3,224,224)	
1	CONV(64,3,1)	CONV(64,3,1)	
2	CONV(64,3,1)	CONV(64,3,1)	
3	POOL(2)	POOL(2)	
4	CONV(128,3,1)	CONV(128,3,1)	
5	CONV(128,3,1)	CONV(128,3,1)	
6	POOL(2)	POOL(2)	
7	CONV(256,3,1)	CONV(256,3,1)	
8	CONV(256,3,1)	CONV(256,3,1)	
9	CONV(256,3,1)	CONV(256,3,1)	
10	POOL(2)	CONV(256,3,1)	
11	CONV(512,3,1)	POOL(2)	
12	CONV(512,3,1)	CONV(512,3,1)	
13	CONV(512,3,1)	CONV(512,3,1)	
14	POOL(2)	CONV(512,3,1)	
15	CONV(512,3,1)	CONV(512,3,1)	
16	CONV(512,3,1)	POOL(2)	
17	CONV(512,3,1)	CONV(512,3,1)	
18	POOL(2)	CONV(512,3,1)	
19	FC(4096)	CONV(512,3,1)	
20	DROP(0.5)	CONV(512,3,1)	
21	FC(4096)	POOL(2)	
22	DROPOUT(0.5)	FC(4096)	
23	FC(1000)	DROP(0.5)	
24	-	FC(4096)	
25	-	DROP(0.5)	
26	-	FC(1000)	

Table 1: The architecture of VGG-16,VGG-19, and VGG-CNN-S

2.2 ResNet50 model

ResNet50[2] was designed as a residual learning framework to ease the training of networks. The network had a depth of up to 152 layers—8x deeper than VGG networks. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task.

The parameters which used to setup the network during fine-tuning, have been shown in Table.2. These parameters of every fine-tuning process are the same.

Parameter	Initial value	End value
Epochs	5000	
Training batch size	128	
Testing batch size	128	
Learning rate	0.000001	
Momentum	0.9	

Table 2: The network parameters in fine-tuning model

3 Experiments

The dataset includes 260 image in 3-channels was used to fine-tune on each trained model. Table.3 shows the losses during fine-tuning (training and validation loss). During fine-tuning, the learning rate and momentum were kept the same on all trained model (0.000001 and 0.9, respective). Each model was fine-tuned in two steps: *freeze and un-freeze* some "lower" layers in 5000 epochs.

Model	Training loss	Validation loss
VGG-16 (unfreeze)	11.00030	9.41890
VGG-16 (freeze)	12.52632	12.75353
VGG-19 (unfreeze)	10.90212	8.58467
VGG-19 (freeze)	11.00235	9.29593
VGG-CNNS (unfreeze)	8.56958	13.49900
VGG-CNNS (freeze)	8.42673	13.41841
ResNet (unfreeze)	0.05461	115.05157
ResNet (freeze)	0.04200	122.41291

Table 3: The losses during fine-tuning the trained models

From the losses in Table.3, the classification models can be used to fine-tune on a regression problem. However, the result seems that not good as we expect. Besides, to ensure the fine-tuning work, the learning rate have been set to a very small value (0.000001). This will spend a lot of time for fine-tuning. The lowest of validation value is $sqrt(8.58467) \approx 2.929$.

In the list of result, the train loss of ResNet is very impressive but the range between training and validation is very large. This problem appears as an overfitting on this model.

4 Conclusions

In this study, we have fine-tuned some trained models on pronotum dataset (RGB format). All the models have been worked well for a regression problem. Despite, the efficiencies of fine-tuning processes are not good enough as we expect. Besides, the limitation of data had overfitting reappear.

References

[1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.