Fine-tuning CNN on Beetles datasets

LE Van Linh

January, 2018

Abstract

In this study, we trained the convolutional neural network (CNN) [1] on three parts of beetles: pronotum, head, and body part. Then, the trained model was fine-tuned on pronotum images. The predicted landmarks on pronotum images were compared with the manual landmarks by calculating the distance among them. The result is evaluated by comparing the losses among training processes. Besides, a comparation with the last result (predicted landmarks that we obtained when trained from scratch on pronotum dataset) is also considered.

1 Dataset

The data includes images in three sets of the beetle: pronotum, body, and head (Fig.1). Each dataset includes 293 color images. For each dataset, the images are divided into two subsets: training and validation (called training set) include 260 images, and the testing set has 33 images. To have enough images for training process, the training sets have been combined together. Then, the training dataset was enlarged to 5460 images (1820×3) following the way that we change the values of pixels on the images. At this time, we have enough the images for training. However, another problem has occurred: the number of landmarks of each part is different: 8 landmarks on pronotum part, 11 landmarks on the body part, and 10 landmarks on the head part. We see that the trained model will be fine-tuned on pronotum dataset. So, we kept the number of the landmark on pronotum as a reference and we suppress some landmarks on body and head part. Specifically, we have removed *three* landmarks on the body part ($1^{st}, 6^{th}, 9^{th}$) and *two* landmarks on the head part ($5^{th}, 6^{th}$).

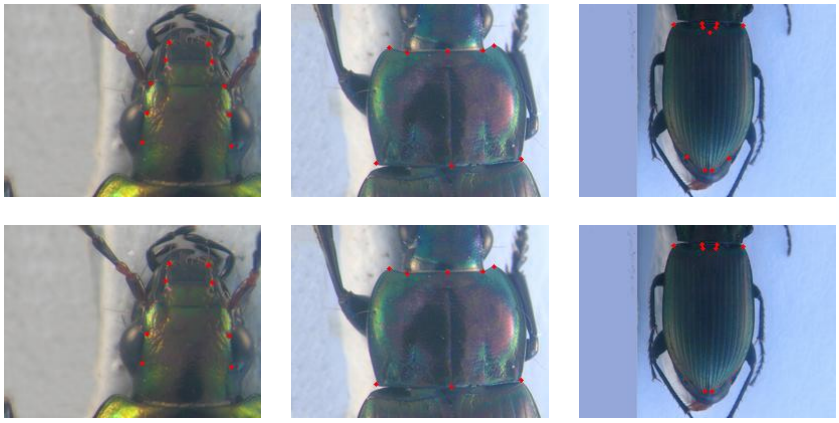


Figure 1: The dataset images. *Top row*: The images with their manual landmarks. *Bottom row*: The training images (some landmarks have been removed on head and body parts)

2 The model

In this study, we use the model that we have used to train on pronotum dataset. The network consists on three repeated-structure of a convolutional layer followed by a maximum pooling layer and dropout layer. The depth of convolutional layers increases from $32, 64$, and 128 with different size of the filter kernel: $3 \times 3, 2 \times 2$, and 2×2. All the kernels of pooling layers have the same size of 2×2. The dropout probability values used for dropout layers are $0.1, 0.2$, and 0.3. Then, three full connected layers have been added to the network. A dropout layer with probability of 0.5 was added between the first two full connected layers. The outputs of the full connected layers are $1000, 1000$, and 16, respectively. The output of the last full-connected layer corresponds to 8 landmarks (x and y coordinates) which we would like to predict (Fig. 2).

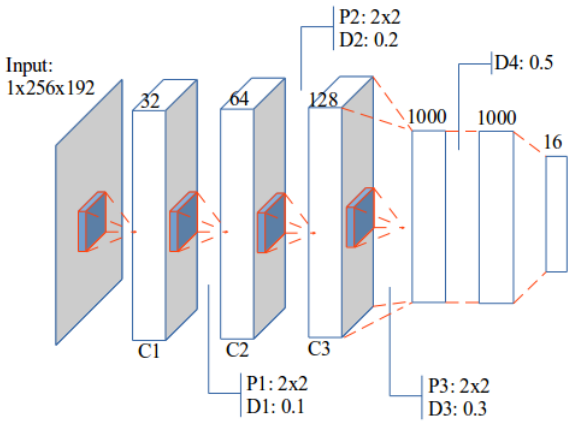


Figure 2: The architecture of CNN model

The parameters of CNN are shown in Table.x.

Parameter	Initial value	End value
Epochs	10000	
Training batch size	128	
Testing batch size	128	
Learning rate	0.03	0.0001
Momentum	0.9	0.9999

Table 1: The network parameters in proposed model

3 Experiments

3.1 Training on three parts of beetle

The dataset includes 5460 images was trained on the model with 10000 epochs[1]. The images are randomly divided into training set and validation set followed the ratio $6:4$. The learning rate began at 0.03 and decreased to 0.00001 during training. In vice versa, the momentum started at 0.9 and increasing to 0.999 at the end of the training process.

Fig.3 shows the losses during training process. At the beginning, the validation loss is always higher than the tranining loss, but from the 2000 epochs, the training loss begins stable while

[1]An epoch is a single pass through the full training set

the validation loss continue to decrease. At the end of training, the losses values are 0.00029 and 0.00009 for training and validation, respectively.

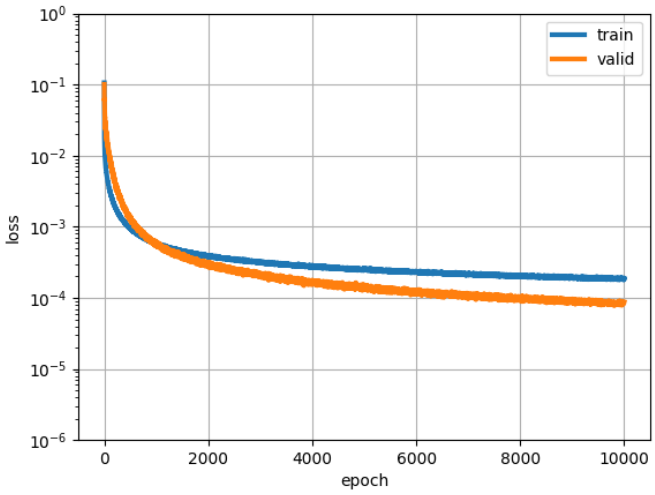


Figure 3: The losses during training on the images of three parts

Fig.4 shows the predicted landmarks on some images in test set.

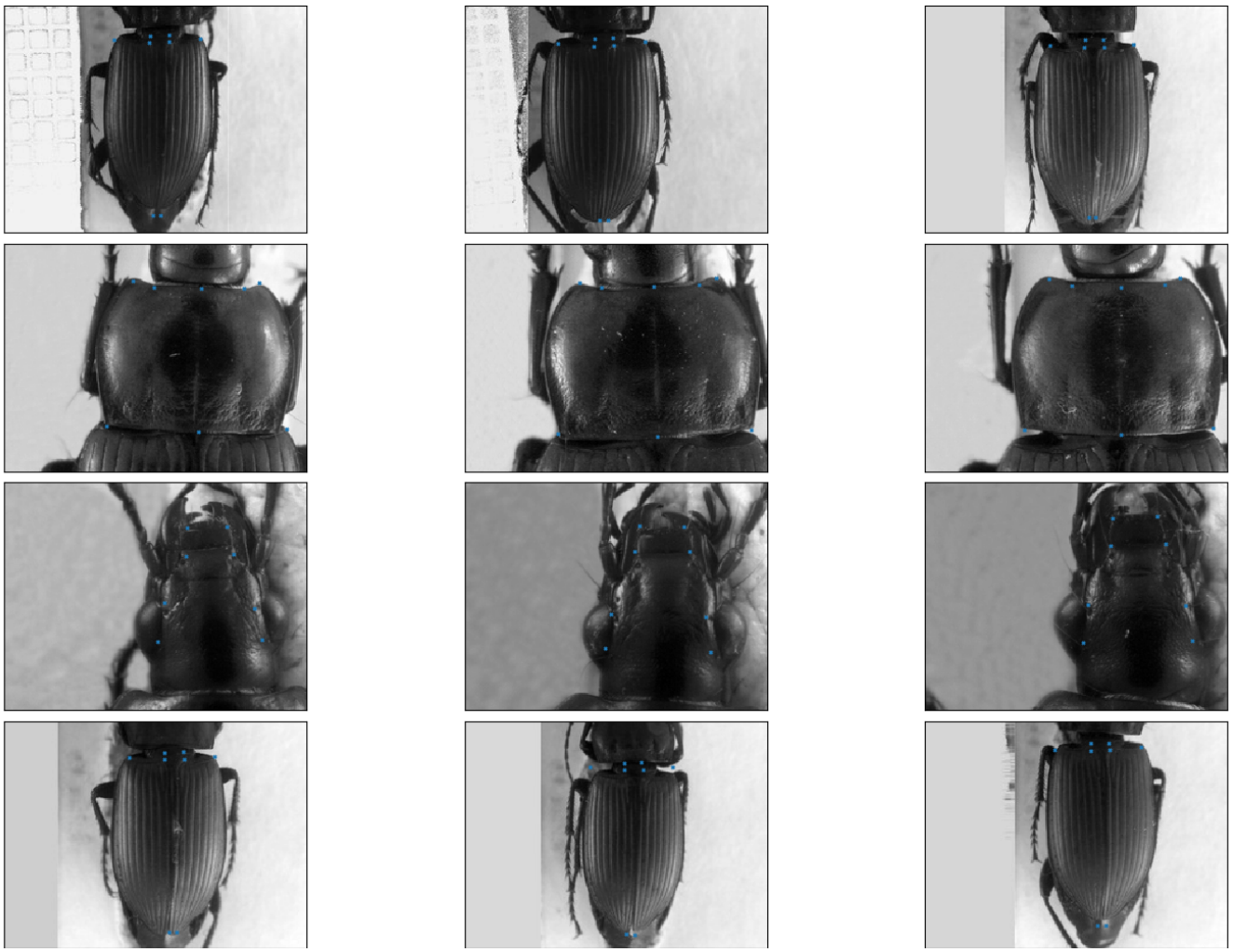


Figure 4: The blue points present for the predicted landmarks on the images in test set

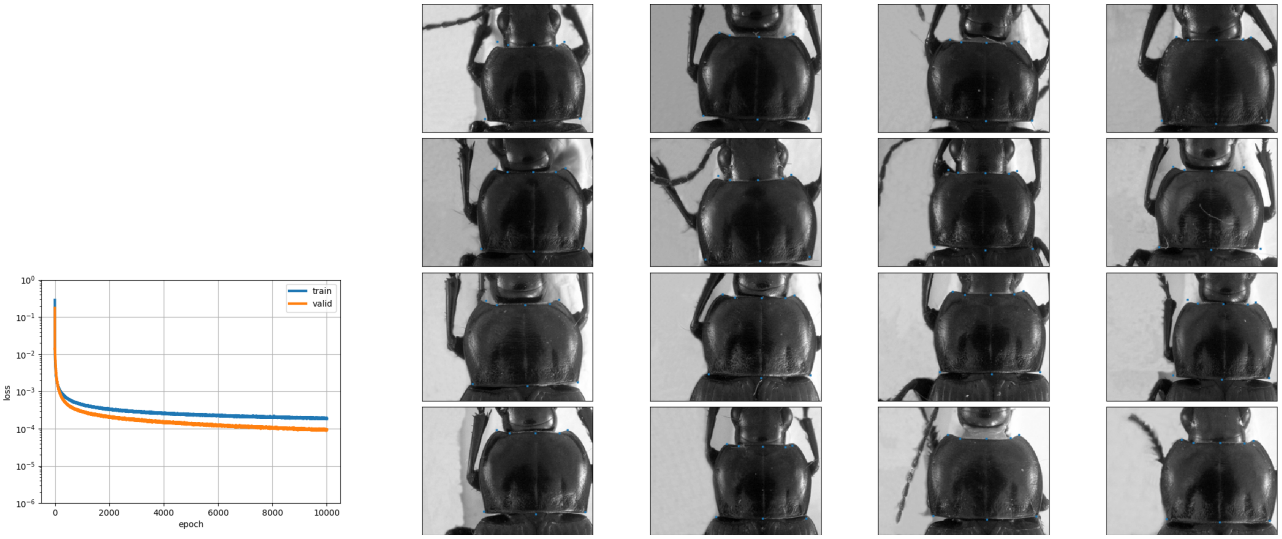
3.2 Fine-tuning on pronotum dataset

The trained model have been continued to fine-tune [2] with pronotum dataset. To get all predicted landmarks for the pronotum images, a scenario to choose the test images is executed. For each round, we have chosen 33 images for the test set, the remaining images have been put to training test. Table.2 shows the losses during fine-tuning on different dataset of pronotum images.

Round	Training loss	Validation loss
1	0.00019	0.00009
2	0.00018	0.00010
3	0.00018	0.00010
4	0.00019	0.00008
5	-	-
6	-	-
7	-	-
8	-	-
9	-	-

Table 2: The losses during fine-tuning model

Fig.5 shows an example of the losses during fine-tuning and corresponding predicted landmarks on the test set.



(a) Training losses and validation loss

(b) A pronotum with predicted landmarks

Figure 5: An result example when fine-tuning the trained model on pronotum dataset

After fine-tuning, the predicted landmarks of all images are provided. To evaluate the effects of fine-tuning, we calculated the distance between the predicted landmarks and corresponding manual landmarks. A statistic on the distance of each landmarks is also computed.

Table.3 shows the average error distance given by each landmark. The values in **Distance 1** and **Distance 2** columns present for the average distance of all landmark when the pronotum images were trained from scratch and fine-tuning, respectively.

#Landmark	Distance 1	Distance 2
1	4.002	-
2	4.4831	-
3	4.2959	-
4	4.3865	-
5	4.2925	-
6	5.3631	-
7	4.636	-
8	4.9363	-

Table 3: The average distance per landmark.

4 Conclusions

A CNN model has been trained on a dataset that includes the images of three parts of beetle. The trained model then has been fine-tuned with the pronotum dataset. Comparing the losses when we trained the pronotum from scratch, the losses during fine-tuning has been improved 40% on validation test. Besides, the coordinates of predicted landmarks are also more accuracy than the last result (training from scratch) (Table.3). From the result, we can see that fine-tuning has affected to the results from CNN. However, the effects still limits in our case. The experiments of the techniques on fine-tuning need to do to reach to the result as we expect.

References

[1] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 253–256. IEEE, 2010.

[2] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.