Landmarks prediction on Beetle anatomical by applying Convolutional Neural Network

LE Van Linh

May 30, 2018

Abstract

In morphometric studies, landmarks are regarded as one of important properties to analyze the object's shape. Especially in biology, collecting complete landmarks give us the information of the oganism. From that, the biologists can study the complex interactions between evolution of the oganism and environment factors. In the context of this study, we focus on one of the most common insect of North-Western France, Carabidae (Beetle). Landmarking manually will be a time-consuming process. In order to investigate the posibility of automatic identify the landmarks on Beetles, we propose a convolutional neural network (CNN) to predict the landmarks on the parts of Beetle: pronotum, head, and body. The proposed model will witnessed on the datasets includes 293 images (for each part) in different sizes. The experiments will be done in two directions: training the model from scratch and using fine-tuning. During the experiments, the coordinates of automatic landmarks will be evaluated by comparing with the manual coordinates, which are given by the biologists.

1 Introduction

Morphometric landmarks are important features in biological investigations. They are using to analyze the shape of the organisms. Depending on the organisms, the number of landmarks on their shapes is different. When we consider the position of the landmarks with the shape, most of landmarks are located on the edges, for example, the landmarks on wings of Drosophila fly [?]. Besides, we can also see the landmarks which stayed inside the anatomical part, i.e, landmarks on pinna of human ears [?]. Currently, the landmarks are set manually by the biologsits. However, landmarking manually will be a time-consuming process and difficult to pre-process. Consequently, a process that proposed automatically the coordinates of landmarks could be interested.

In image processing, segmentation is a most often step of the methods. In some cases, the object of interest is easy to extract and can be analyzed with the help of a lot of very well-known image analysis procedures. The result of segmentation step is very useful for many purposes. Depending on purpose of the applications, the object can be segmented or un-segmented before continuing the futher steps. Landmarks setting is no different. In a previous study [?], we have analyzed two parts of beetle: left and right mandibles. These parts are easy to segment. In that work, we have applied a set of algorithms based on segmentation, image alignment and SIFT [?] to detect the landmarks on mandibles.

Unfortunately, the images of other parts are not simply as the mandibles. Besides the main objects, we have also the different parts, i.e, we have a part of head and the legs in pronotum images. The image becomes very noisy. If we would like to segment the object as traditional processes, the process may have consumed a lot of time and difficult to choose a proper method.

This is the reason that we turn the landmarking problem on some parts of beetle (i.e, pronotum, head, and body) to a way of analyzing images without the segmentation step.

As the beetles have not been dissected, their anatomical parts have not been set apart. So image segmentation of each part, as they are still attached to the whole specimen, is problematic and has been given up. Coordinates of manual landmarks for each part have been provided and are considered as the ground truth to evaluate the predicted ones by our methods. Fig.1 shows the parts of beeltes and their manual landmarks what we are looking for.

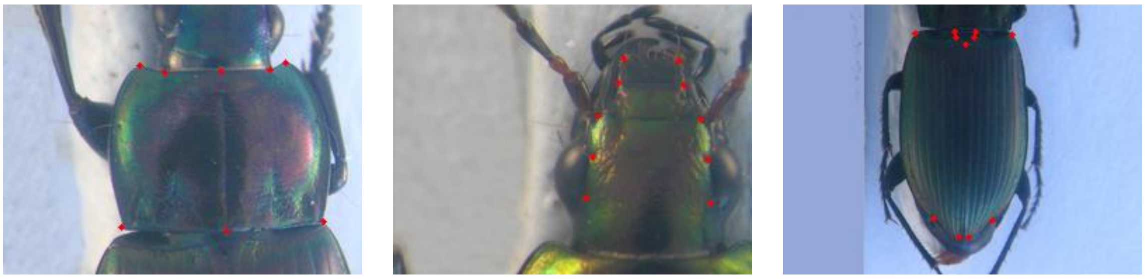


Figure 1: The dataset images with their manual landmarks.
From left to right: pronotum, head, and body

To achieve the landmarks prediction, we have proposed a CNN model [?] by using Lasagne library [?]. In the first evaluation, the proposed model has been trained from scratch on the dataset of each part. In the second step, the evaluation has been modified to use a fine-tuning [?] stage.

Our contributions in this study are as follows: In the next section, we present the related works about automatic estimation landmarks on 2D images. The architecture of proposed network will be presented at section ??. In section ??, we describe the process to augment the dataset. All the experiments of our model will be shown in section ??. It includes the results to evaluate the model and comparison between two working strategy on neural networks.

2 Related works

In geometric morphometry, landmarks (or points of interest) are important features to describe a shape. Depending on the difficulty to segment the objects inside the images, setting automatic landmarks can rely on different methods. When segmentation can be applied, Lowe et al. [?] have proposed a method to identify the key points in the 2D image. From the detected key points, the method is able to match two images. Palaniswamy et al. [?] have applied probabilistic Hough Transform to automatically estimate the landmarks in images of Drosophila wings. In a previous study [?], we have extended Palaniswamys method to detect landmarks automatically on beetles mandibles with good results. Unfortunately, when the segmentation is not precise, we have observed that the results are getting worse. This is why we have turned our work on Deep Learning algorithms in order to find a suitable solution to predict the landmarks without any segmentation step.

Deep Learning models are coming from machine learning theory. They have been introduced in the middle of previous century for artificial intelligence applications but they encounter several problems to take real-world cases. More recently, the improvement of computing capacities, both in memory size and computing time with GPU programming has opened new perspective for Deep Learning. Many deep learning architectures have been proposed to solve the problems of classification [?], image recognition [?], speech recognition [?] and language translation [?]. To implement the algorithms, many frameworks have been built such as Caffe [?], Theano [?], Tensorflow [?],.... These frameworks help the users to design their application by re-using already proposed network architectures. In image analysis domain, Deep Learning, specifically with CNN, can be used to predict the key points in an image. Yi Sun et al. [?] have proposed a cascaded convolutional network to predict the key points on the human face. Zhang et al. [?] optimize facial landmarks detection with a set of related tasks such as head pose estimation, age estimation, ...Cintas et al. [?] have introduced a network to predict the landmarks on human ear images to characterize ear shape for biometric analysis. In the same way, we have applied CNN computing to predict the landmarks on beetles anatomical parts. The predicted landmarks will be found in two strategies: training the model from scratch and applying a fine-tuning process.

3 Dataset

The data includes images in three sets of the beetle: pronotum, body, and head (Fig.??). Each dataset includes 293 color images which are taken with the same camera in the same condition with a resolution of 3264 × 2448. Each image has the manual landmarks setting by biologists, i.e, pronotum has 8 manual landmarks. For each dataset, the images are randomly divided into two subsets: training and validation (called training set) include 260 images, and the testing set has 33 images.

One of the main characteristics of CNN is that it must use a huge number of data and one can consider that only several hundreds of images is not enough to feed a CNN. Moreover, working with small dataset can push us again to the popular problem of overfitting. A way to enlarge the dataset size has to be considered. In image processing, we usually apply transform procedures (translation, rotation) to generate a new image. Unluckily the methods to compute features through a CNN most often are translation and rotation independent. So, we have used another method to augment dataset from the down-sampling images.

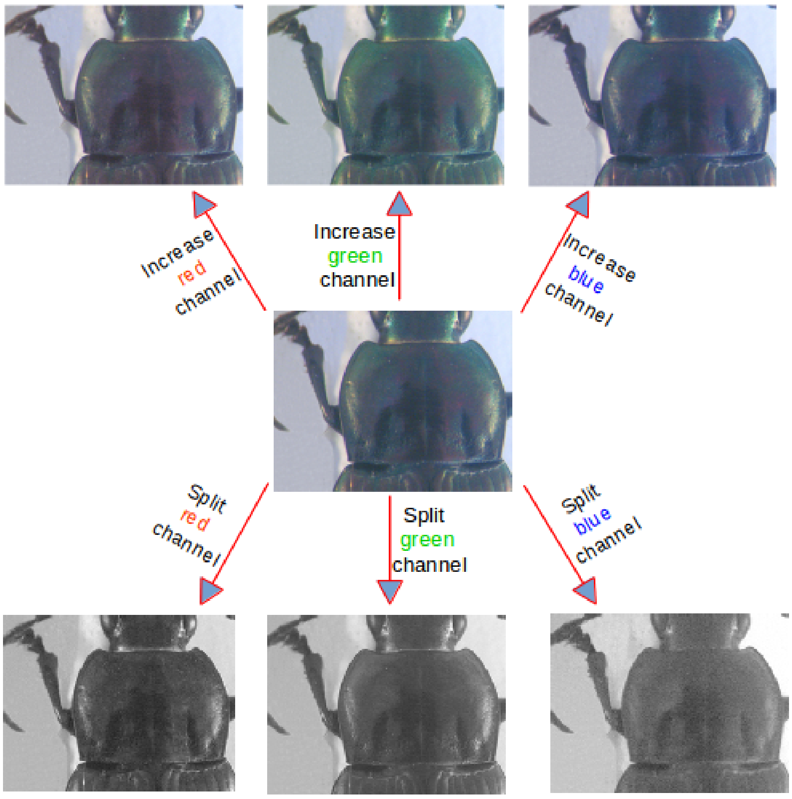


Figure 2: An example of augmentation data. From an original image, six augmented version have generated.

A first procedure has been applied to change the value of each color channel in the original image. According to that, a constant is added to one of the RGB channels each time it is used for training. Each constant is sampled in a uniform distribution $\in [1, N]$ to obtain a new value caped at 255. For example, we can add a constant $c = 10$ to the red channel of all images in order to generate new images. This operation can be done for the three color channels. The second procedure separates the channels of RGB into three gray-scale images. As the network works on single channel images we are able to generate six versions of the same image, the total number of images used to train and to validate is $260 \times 7 = 1820$ images (six versions and original image). This has been an efficient way to proceed to the dataset expansion. Fig.2 presents two strategies that we used to augment the data.

To have obtained the datasets in different sizes of images, we have applied different methods to down-sampling from the original images (see section **??**). In the context of this study, we have considered the images in two different resolutions: 256×192 and 96×96. Of course, the coordinates of manual landmarks have been also scaled to fit with the new resolutions. In the first considered resolutions (256×192), we have simplied down-sampling from the original resolution. While, in the second considered resolution (96×96), the original images have been cropped from the left to obtain a new resolution of 2448×2448. Then, the cropped images have been down-sampled to the target resolution.

4 The model

The model, what we used to predict the landmarks, receives an image of $1 \times 256 \times 192$ as the input. The network was constructed from 3 *"elementary block"* following by 3 full-connected layers. An elementary block is defined as a sequence of convolution (C_i), pooling (P_i) and dropout (D_i) layers. The parameters for each layers are as below, the list of values follows the order of elementary blocks:

- CONV layres:
  - Number of filters: 32, 64 and 128,
  - Kernel filters size: $(3 \times 3), (2 \times 2)$, and (2×2)
  - Stride values: $1, 1, 1$
  - No padding is used for CONV layers
- POOL layers:
  - Kernel filters size: $(2 \times 2), (2 \times 2)$, and (2×2)
  - Stride values: $2, 2, 2$
  - No padding is used for CONV layers
- DROP layers:
  - Propabilities: $0.1, 0.2$ and 0.3.

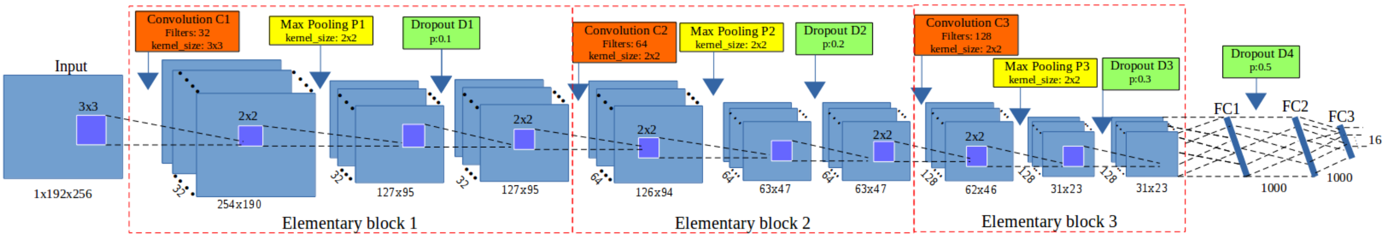


Figure 3: Network architecture using 3 *elementary blocks*. Convolution layer in red, pooling in yellow and dropout in green color.

In the last full-connected layers (FC), the parameters are: FC1 output: 1000, FC2 output:

1000, FC3 output: 16. As usual, a dropout layer is inserted between FC1 ond FC2 with a probability equal to 0.5. Fig.3 illustrate the order of the layers in the network.

The parameters of CNN are shown in Table.1.

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

5 Experiments

The model has been experimented on two resolution datasets: 256×192 and 96×96. For each resolution dataset, the model was evaluated in two strategies: training from scratch and fine-tuning. In each evaluation, cross-validation technique to select the training and the testing data. From that, we able to have the predicted landmarks on all images.

The dataset was trained on the model with 10000 epochs[1]. During training, 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.

To evaluate the prediction, the distance between predicted and manual landmarks have been computed in all images. Then, the average distance and standard deviation (SD) have been calculated on each landmark. Following sub-sections describe the average distance and SD of each dataset in each part of beetle.

5.1 Experiment on the first resolution dataset (256×192)

Landmarks	From scratch		With fine-tuning	
	Average	SD	Average	SD
LM1	**4.002**	**2.5732**	**2.486**	**1.5448**
LM2	4.4831	2.7583	2.7198	1.7822
LM3	4.2959	2.7067	2.6523	1.8386
LM4	4.3865	3.0563	2.7709	1.9483
LM5	4.2925	2.9086	2.4872	1.6235
LM6	**5.3631**	**3.4234**	**3.0492**	**1.991**
LM7	4.636	2.8426	2.6836	1.7781
LM8	4.9363	3.0801	2.8709	1.9662

Table 2: The average distance and standard deviation on pronotum images

Table. **??**, **??**, and **??** show the comparation on average distances and standard deviations on protum, body, and head part, respectively. The results show that the average distance have been improved with using fine-tuning. Besides, we do not see the significant difference of the fine-tuning processes (with and without freezing some layers).

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

Landmarks	From scratch		With fine-tuning	
	Average	SD	Average	SD
LM1	**3.87**	**2.5732**	**2.34**	**3.11**
LM2	3.97	2.7583	2.27	3.15
LM3	3.92	2.7067	2.27	2.96
LM4	3.87	3.0563	2.25	3.26
LM5	4.02	2.9086	2.27	3.28
LM6	**4.84**	**3.4234**	**3.14**	**3.47**
LM7	5.21	2.8426	3.14	3.40
LM8	5.47	3.0801	3.29	3.36
LM9	5.27	3.0801	3.42	3.09
LM10	4.07	3.0801	2.49	3.05
LM11	3.99	3.0801	2.30	3.06

Table 3: The average distance and standard deviation on body (elytra) images

Landmarks	From scratch		With fine-tuning	
	Average	SD	Average	SD
LM1	**5.53**	**2.5732**	**3.03**	**1.89**
LM2	5.16	2.7583	2.94	1.97
LM3	5.38	2.7067	2.96	1.81
LM4	5.03	3.0563	2.88	1.92
LM5	4.84	2.9086	2.76	1.75
LM6	**4.45**	**3.4234**	**2.67**	**2.02**
LM7	4.79	2.8426	2.29	1.64
LM8	4.53	3.0801	2.20	1.54
LM9	5.14	3.0801	2.57	1.60
LM10	5.06	3.0801	2.44	1.53

Table 4: The average distance and standard deviation on head images

5.2 Experiment on the second resolution dataset (96×96)

5.2.1 Pronotum part

Landmarks	From scratch		With fine-tuning	
	Average	SD	Average	SD
LM1	**4.002**	**2.5732**	**2.486**	**1.5448**
LM2	4.4831	2.7583	2.7198	1.7822
LM3	4.2959	2.7067	2.6523	1.8386
LM4	4.3865	3.0563	2.7709	1.9483
LM5	4.2925	2.9086	2.4872	1.6235
LM6	**5.3631**	**3.4234**	**3.0492**	**1.991**
LM7	4.636	2.8426	2.6836	1.7781
LM8	4.9363	3.0801	2.8709	1.9662

5.2.2 Body part

Landmarks	From scratch		With fine-tuning	
	Average	SD	Average	SD
LM1	**4.002**	**2.5732**	**2.486**	**1.5448**
LM2	4.4831	2.7583	2.7198	1.7822
LM3	4.2959	2.7067	2.6523	1.8386
LM4	4.3865	3.0563	2.7709	1.9483
LM5	4.2925	2.9086	2.4872	1.6235
LM6	**5.3631**	**3.4234**	**3.0492**	**1.991**
LM7	4.636	2.8426	2.6836	1.7781
LM8	4.9363	3.0801	2.8709	1.9662

5.2.3 Head

Landmarks	From scratch		With fine-tuning	
	Average	SD	Average	SD
LM1	**4.002**	**2.5732**	**2.486**	**1.5448**
LM2	4.4831	2.7583	2.7198	1.7822
LM3	4.2959	2.7067	2.6523	1.8386
LM4	4.3865	3.0563	2.7709	1.9483
LM5	4.2925	2.9086	2.4872	1.6235
LM6	**5.3631**	**3.4234**	**3.0492**	**1.991**
LM7	4.636	2.8426	2.6836	1.7781
LM8	4.9363	3.0801	2.8709	1.9662

5.3 Training on three parts of beetle

Fig.4 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

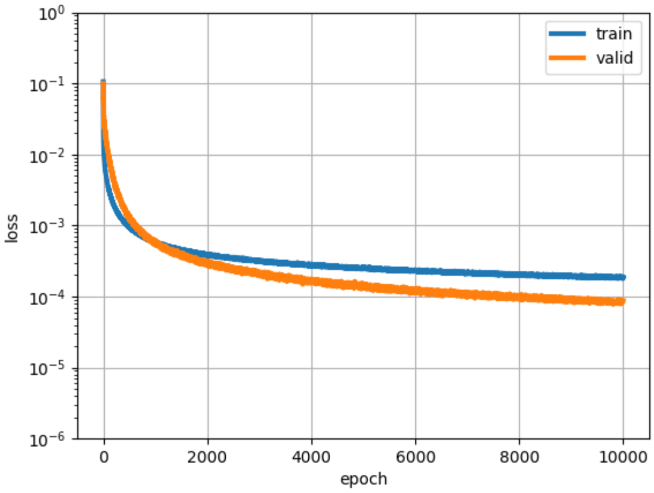


Figure 4: The losses during training on the images of three parts

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.

Fig.5 shows the predicted landmarks on some images in test set.

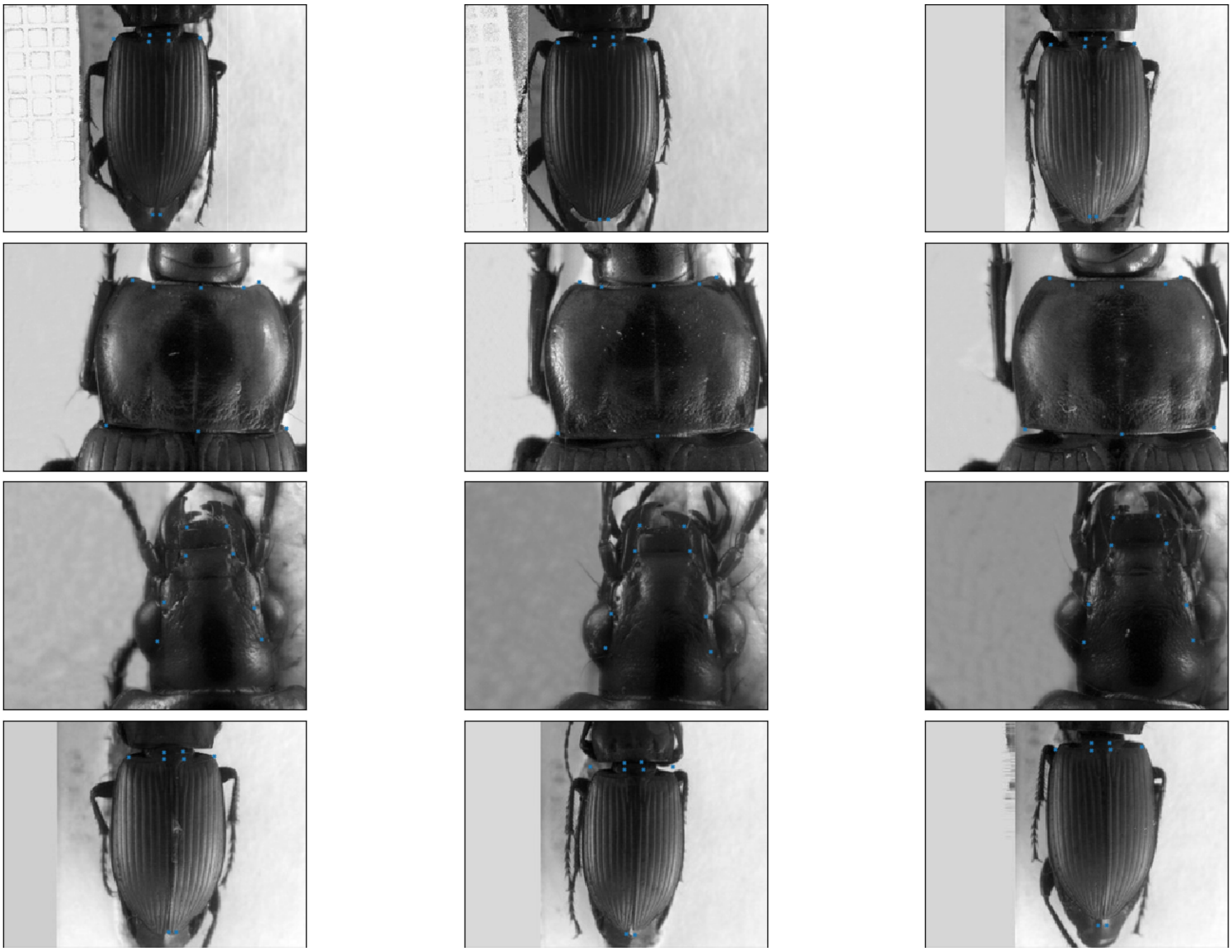


Figure 5: The blue points present for the predicted landmarks on the images in test set

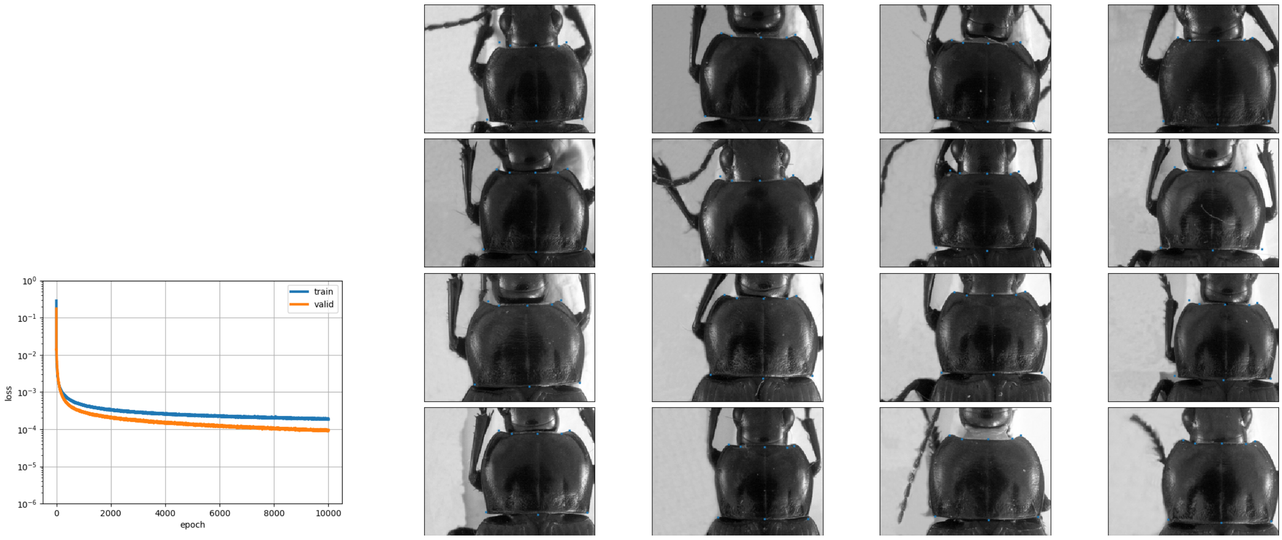
5.4 Fine-tuning on pronotum dataset

The trained model have been continued to fine-tune [1] 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.5 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	0.00019	0.00009
6	0.00018	0.00008
7	0.00019	0.00008
8	0.00018	0.00006
9	0.00018	0.00009

Table 5: The losses during fine-tuning model

Fig.6 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 6: 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.6 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. From the Table. 6, the result from fine-tuning is significantly improved ($\sim$ 38%)

#Landmark	Distance 1	Distance 2
1	4.002	2.486
2	4.4831	2.720
3	4.2959	2.652
4	4.3865	2.771
5	4.2925	2.487
6	5.3631	3.049
7	4.636	2.684
8	4.9363	2.871

Table 6: The average error distance per landmark.

6 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.6). 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] 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.