A convolutional neural network on Beetles dataset

LE Van Linh and BEURTON-AIMAR Marie

October, 2017

Abstract

In this study, we present a convolutional neural network (CNN) which is used to predict the landmarks on beetle's images. The model is designed as a pipeline of the layers. It is evaluated on five datasets of beetle: *left mandible, right mandible, pronotum, body, and head*. The models which used to predict the landmarks for each beetle's part have the same structures but the output at the last layer is modified to suitable with the number of landmarks that it should be predicted. For each dataset, a number of 260 images are used to train and validate, the remaining images are used to test the output model. The evaluation is the correlation coefficient between the manual coordinates (which given by the biologist) and the predict coordinates. Besides, a statistic based on the distances between the manual landmarks and predicted landmarks are also calculated. The model is implemented by Python on Lassagne framework[1].

1 Convolutional neural network

1.1 Architecture

The network includes three convolutional(CONV) layers followed by three maximum pooling(POOL) layers, four dropouts(DROP) layers, and three full connected(FC) layers (Fig.1). The input of the network is the gray-scale image with the size of 256 × 192. The depth of network can be expressed by increasing of the deep at each convolutional layer. They are increased from $32, 64$, and 128 from the first CONV layer to the third CONV layer with different filter sizes. While, the filter sizes are kept in the same size for every POOL layers. The dropout ratios of the DROP layers increase from the first to the end: $0.1, 0.2, 0.3$, and 0.5. At the end of the network, three full connected are set up to predict the landmarks. The first two FC layers have the same outputs(1000) while the output at the last FC has been change to correspond with the number of landmarks. The detail parameters at each layer are presented in Appendix. The model is implemented by Lassagne framework[1].

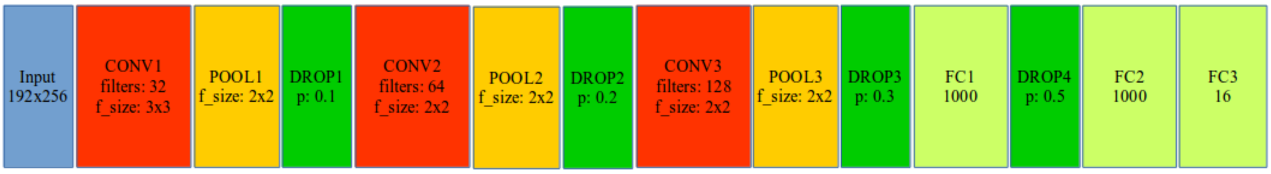


Figure 1: The illustration of the convolutional neural network

1.2 Parameters

The model is trained with 5000 `epochs` and `batch size` of 128. For each epoch, the dataset is randomly split into training set and validation set with the ratio of 0.6 : 0.4. The `learning rate` and `momentum` are initialized to 0.03 and 0.9, respectively. During training, they are re-calculated to adjust with the remaining epochs. All the initial parameters are shown in the Table 1.

Parameter	Initial value	End value
Epochs	5000	
Training batch size	128	
Testing batch size	128	
Learning rate	0.03	0.0001
Momentum	0.9	0.9999
Training data	0.6	
Validation data	0.4	

Table 1: The network parameters in proposed model

2 Data

The beetle dataset includes the images of five parts: *left mandible, right mandible, pronotum, body, and head.* For each part, a collection of **293** images are collected. However, the number of the images in each part are changed after checking to suppress the not-working images (i.e empty image, broken object). Table 2 shows the number of available images in each part and the number of the images in each process.

Part	Total available images	Training + Validation	Testing
Left mandible	286	260	26
Right mandible	290	260	30
Body	293	260	33
Head	293	260	33
Pronotum	293	260	33

Table 2: The number of available images and the number of the images which used to train (and validate) and test

Because the number of the images are limited (just 260 color images), it does not enough to use for training process. Additional, the models are worked on gray-scale images. So, we applied some rules to enlarge the dataset for each part. The first rule is adding a constant value to a channel of RGB image, we will have a new RGB image. For example, from an original RGB image, if we add 10 to red channel, we will have a new image $(R + 10)GB$. Then, we apply the same rule with blue and green channel, we will obtain two new images: $R(G + 10)B$ and $RG(B + 10)$. By that way, from an RGB image, we can generate three RGB images by adding a constant to each channel(each time just change to a channel). The second rule is splitting the channels of RGB image (because the models work on gray-scale). It means that we can generate six versions from an original image. At the end, the number of the image in the training data is $260 \times 7 = 1820$ images (six versions and original). Before giving to the models, the images are down-sampled with the size of 256×192. The number of the images in training set and validation set are splitted automatically by the model's parameter.

3 Experiments

In practical, convergence is usually faster if the average of each input variable over the training set is close to zero. Because the values of the pixels and the coordinates of the landmarks are positive. If we consider that we stay at the a layer of the network, and the weights are updated by an amount proportional to δx(δ is the scalar error at the layer and x is the input vector). When the input vectors are positive, the updates of weights that feed into the layer will be the same sign($sign(\delta)$), it means that the weights can only all decrease or all increase together for a given input. That, if the weight vector change direction, it can only do by zigzagging which is inefficient and thus slow down learning. Therefore, it is good to shift the inputs so that the average over the training set is close to zero. Moreover, when the input is set closed with zero, it will more suitable with the sigmoid activation function[2]. So, *the brightness of the image is normalized to* $[0, 1]$*, instead of* $[0, 255]$*. And, the coordinates of the landmarks are normalized to* $[-1, 1]$*, instead of* $[0, 256]$ *and* $[0, 192]$ *before giving to the network.*

For each part, the network is training and validation with many times (called round). For each time, the training dataset is changed following the way to choose the test dataset (i.e circular). At the end, we can obtain the predicted landmarks of all images in the dataset by combining all the testing images corresponding with the training model.

The predicted landmarks are evaluated by two ways: calculating correlation coefficient and computing the statistic based on the number of landmarks has good prediction. The correlation coefficient is calculated by using three methods: Pearson[3], Spearman[4], and Kendall[5]. The good prediction statistic is computed based on the distance between manual and predicted landmarks. Firstly, the distance between manual and prediction one is calculated. Then, the average distance of this landmark on all images has been computed. A predicted landmark is considered as well prediction if the distance between it and the corresponding manual is less than average value.

This section presents the experimental on all parts of Beetle. For each part, we describe the losses (training and validation), the correlation coefficient and the statistic of good prediction.

3.1 Left mandible part

Table 3 shows the information during training and validation on left mandible.

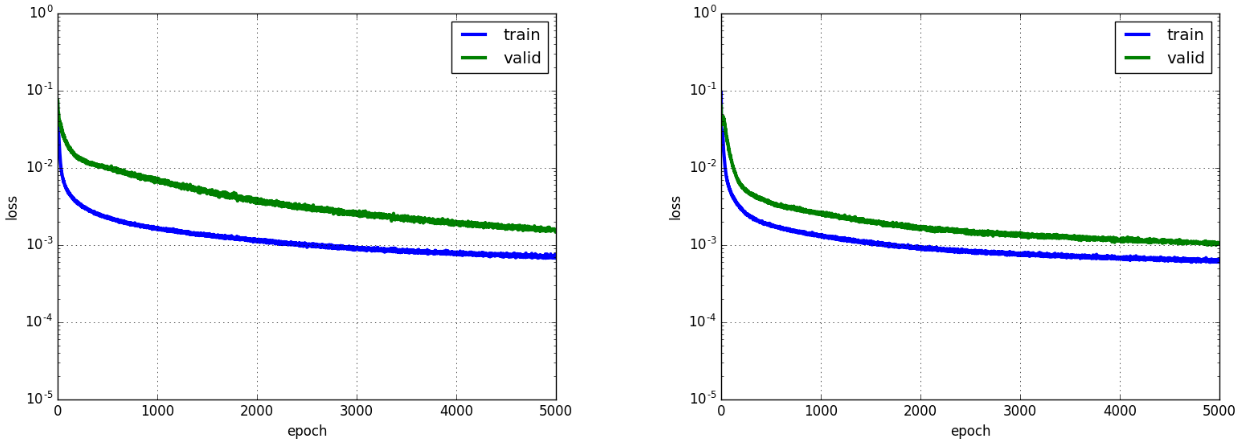
Round	Total images	Testing index (from-to)	Training index (from-to)	Training loss	Validation loss
r1	286	1-26	27-286	0.00073	0.00148
r2	286	27-52	remaining	0.00074	0.00149
r3	286	53-78	remaining	0.00074	0.00177
r4	286	79-104	remaining	0.00068	0.00141
r5	286	105-130	remaining	0.00077	0.00231
r6	286	131-156	remaining	0.00070	0.00180
r7	286	157-182	remaining	0.00063	0.00125
r8	286	183-208	remaining	0.00062	0.00104
r9	286	209-234	remaining	0.00067	0.00173
r10	286	235-260	remaining	0.00067	0.00145
r11	286	261-286	remaining	0.00072	0.00188

Table 3: The training loss and validation loss at each training round of left mandible

Which:

- **Round**: indexing training round
- **Total images**: total images of left mandible
- **Testing index**: indexing of the images that chosen to test set.
- **Training index**: indexing of the images that chosen to train and valid set.
- **Training loss**: training loss at a round
- **Validation loss**: validation loss at a round

Fig.2 shows the curves of training and validation losses of two rounds of left mandible.



(a) Round 1

(b) Round 8

Figure 2: The losses curves of training and validation of two training rounds of left mandible

The correlation coefficient results are shown in Table 4.

Method	x correlation	y correlation
Pearson	0.9781574	0.9875064
Spearman	0.983688	0.9800946
Kendall	0.9136765	0.8932026

Table 4: The correlation between manual and predicted landmarks on left mandible images

Fig.3 shows the proportions of well predicted landmarks on left mandibles.

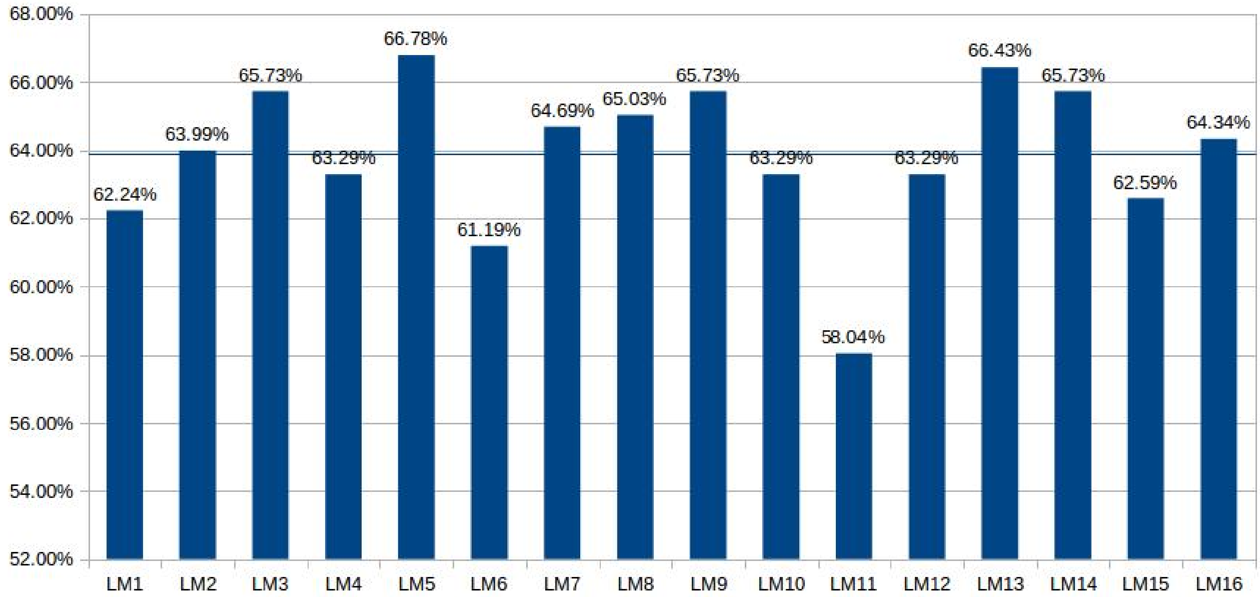


Figure 3: The proportion of well predicted landmarks on left mandibles

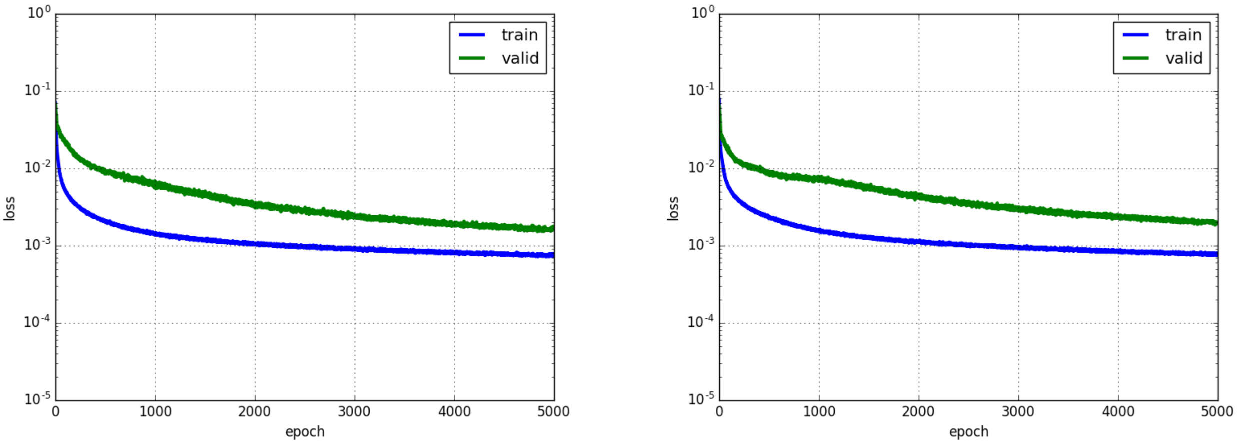
3.2 Right mandible part

The information of each training round on right mandible is shown in Table 5.

Round	Total images	Testing index (from-to)	Training index (from-to)	Training loss	Validation loss
r1	290	1-30	31-290	0.00075	0.00162
r2	290	31-60	remaining	0.00081	0.00208
r3	290	61-90	remaining	0.00076	0.00158
r4	290	91-120	remaining	0.00075	0.00167
r5	290	121-150	remaining	0.00079	0.00206
r6	290	151-180	remaining	0.00080	0.00263
r7	290	181-210	remaining	0.00081	0.00245
r8	290	211-240	remaining	0.00080	0.00194
r9	290	241-270	remaining	0.00079	0.00157
r10	290	271-290	remaining	0.00082	0.00242

Table 5: The training loss and validation loss at each training round of right mandible

Fig.4 shows the curves of training and validation losses of two rounds on right mandible.



(a) Round 1

(b) Round 8

Figure 4: The losses curves of training and validation of two training rounds of right mandible

Table 6 shows the correlation coefficient between manual landmarks and predicted landmarks on right mandibles.

Method	x correlation	y correlation
Pearson	0.9852194	0.9858498
Spearman	0.9863889	0.983251
Kendall	0.9104557	0.898321

Table 6: The correlation between manual and predicted landmarks on right mandible images

Fig.5 shows the proportions of well predicted landmarks on right mandibles.

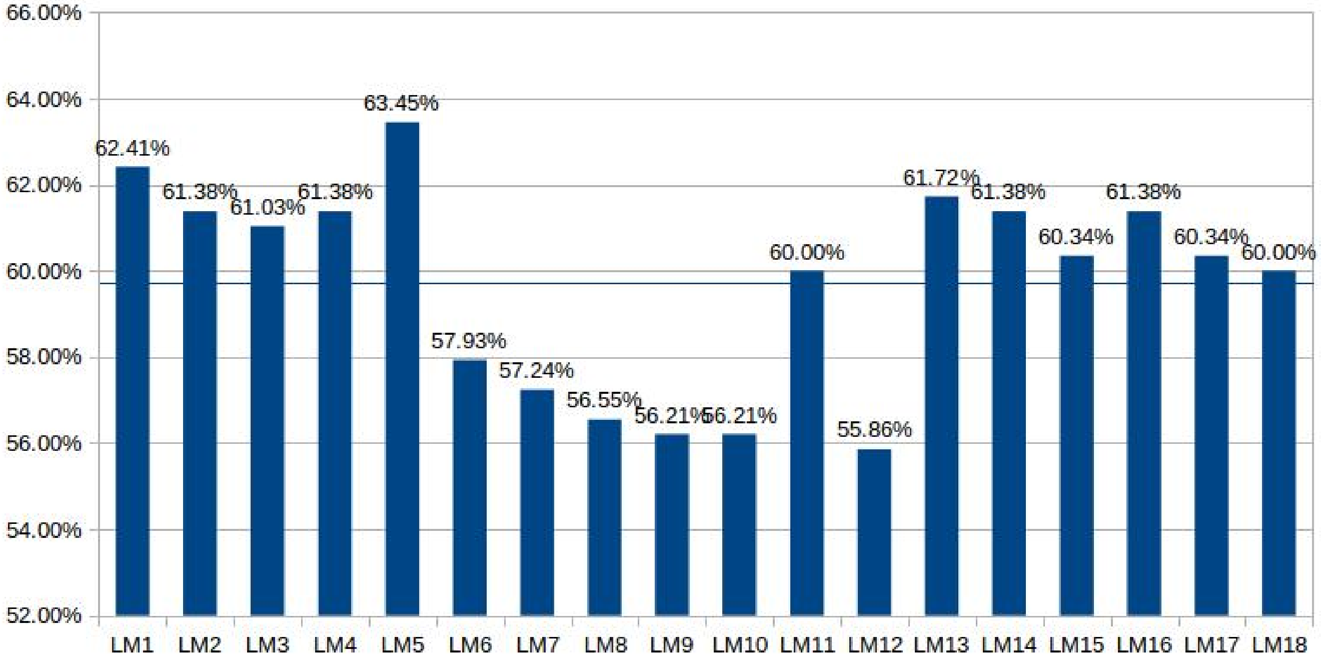


Figure 5: The proportion of well predicted landmarks on right mandibles

3.3 Pronotum part

The information of each training round on pronotum is shown in Table 7.

Round	Total images	Testing index (from-to)	Training index (from-to)	Training loss	Validation loss
r1	293	1-33	34-293	0.00075	0.00162
r2	293	34-66	remaining	0.00081	0.00208
r3	293	67-99	remaining	0.00076	0.00158
r4	293	100-132	remaining	0.00075	0.00167
r5	293	133-165	remaining	0.00079	0.00206
r6	293	166-198	remaining	0.00080	0.00263
r7	293	199-231	remaining	0.00081	0.00245
r8	293	2232-264	remaining	0.00080	0.00194
r9	293	265-293	remaining	0.00079	0.00157

Table 7: The training loss and validation loss at each training round of pronotum

Table 8 shows the correlation coefficient between manual landmarks and predicted landmarks on pronotum part.

Method	x correlation	y correlation
Pearson	0.0	0.0
Spearman	0.0	0.0
Kendall	0.0	0.0

Table 8: The correlation between manual and predicted landmarks on pronotum images

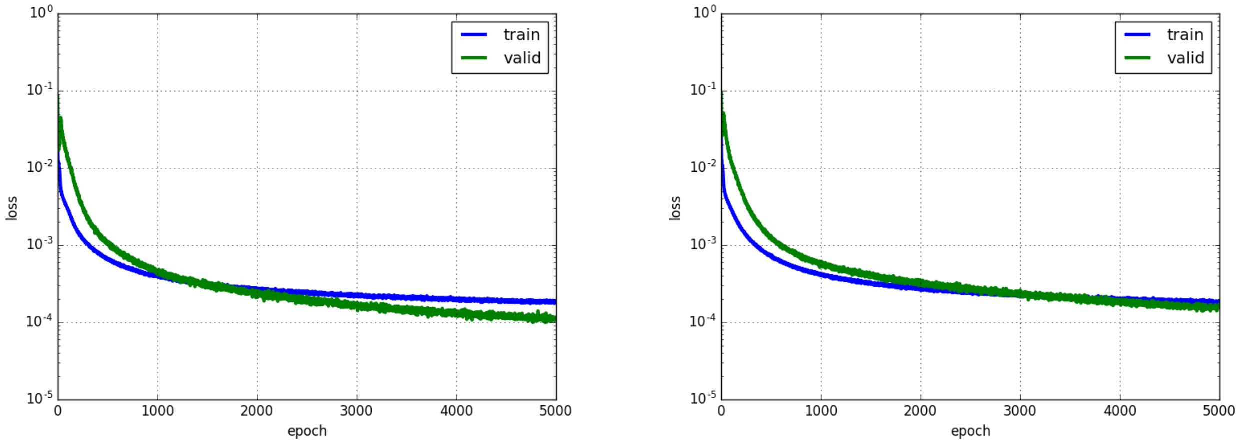
3.4 Body part

The information of each training round on body part is shown in Table 9.

Round	Total images	Testing index (from-to)	Training index (from-to)	Training loss	Validation loss
r1	293	1-33	34-293	0.00019	0.00012
r2	293	34-66	remaining	0.00020	0.00012
r3	293	67-99	remaining	0.00019	0.00012
r4	293	100-132	remaining	0.00020	0.00011
r5	293	133-165	remaining	0.00018	0.00010
r6	293	166-198	remaining	0.00019	0.00013
r7	293	199-231	remaining	0.00018	0.00013
r8	293	2232-264	remaining	0.00018	0.00017
r9	293	265-293	remaining	0.00019	0.00012

Table 9: The training loss and validation loss at each training round of body

Fig.6 shows the curves of training and validation losses of two rounds on body part.



(a) Round 1

(b) Round 8

Figure 6: The losses curves of training and validation of two training rounds of right mandible

Table 10 shows the correlation coefficient between manual landmarks and predicted landmarks on body part.

Method	x correlation	y correlation
Pearson	0.9818338	0.9986623
Spearman	0.9833374	0.980597
Kendall	0.9032424	0.8882938

Table 10: The correlation between manual and predicted landmarks on body images

Fig.7 shows the proportions of well predicted landmarks on body part.

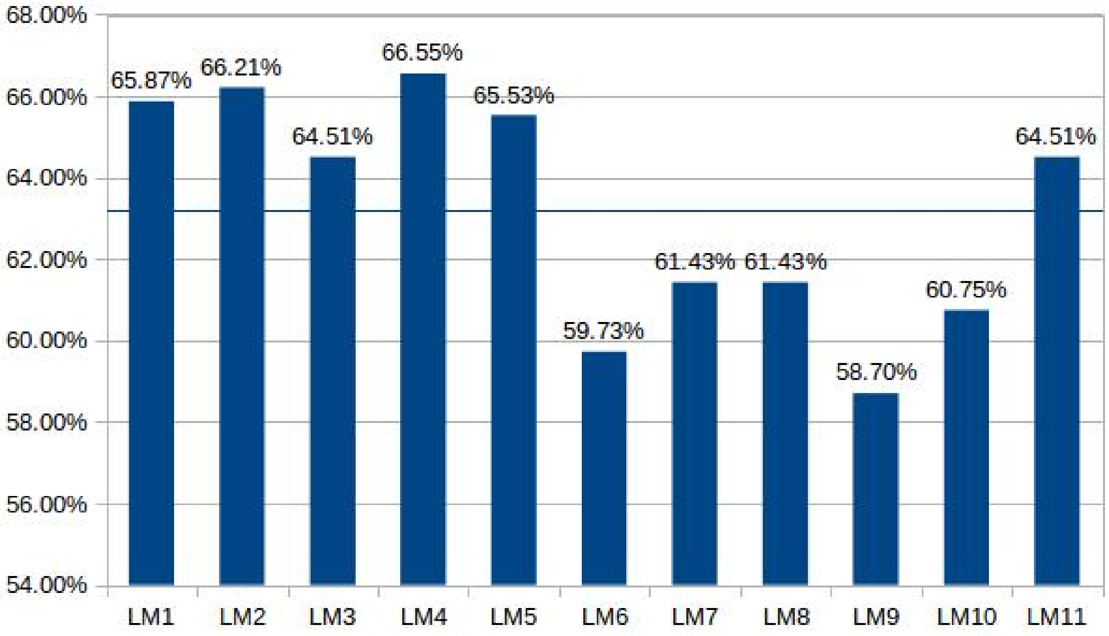


Figure 7: The proportion of well predicted landmarks on body part

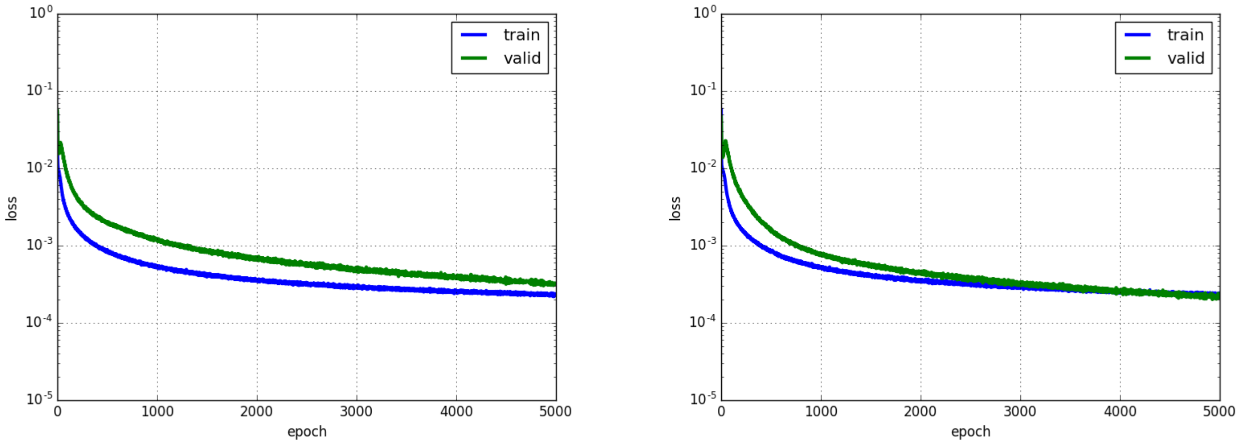
3.5 Head part

The information of each training round on head part is shown in Table 11.

Round	Total images	Testing index (from-to)	Training index (from-to)	Training loss	Validation loss
r1	293	1-33	34-293	0.00023	0.00032
r2	293	34-66	remaining	0.00027	0.00044
r3	293	67-99	remaining	0.00026	0.00051
r4	293	100-132	remaining	0.00026	0.00058
r5	293	133-165	remaining	0.00027	0.00072
r6	293	166-198	remaining	0.00025	0.00050
r7	293	199-231	remaining	0.00023	0.00019
r8	293	2232-264	remaining	0.00024	0.00021
r9	293	265-293	remaining	0.00025	0.00027

Table 11: The training loss and validation loss at each training round of head

Fig.8 shows the curves of training and validation losses of two rounds on right mandible.



(a) Round 1

(b) Round 8

Figure 8: The losses curves of training and validation of two training rounds of head part

Table 12 shows the correlation coefficient between manual landmarks and predicted landmarks on right mandibles.

Method	x correlation	y correlation
Pearson	0.9936695	0.9935629
Spearman	0.99080	0.9944676
Kendall	0.9237709	0.9387203

Table 12: The correlation between manual and predicted landmarks on head images

Fig.9 shows the proportions of well predicted landmarks on head part.

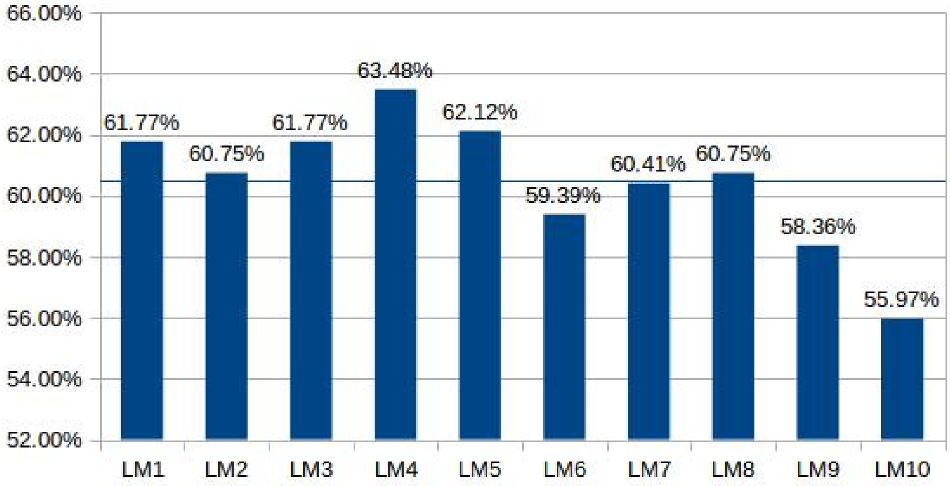


Figure 9: The proportion of well predicted landmarks on head part

4 Conclusions

In this study, we proposed a CNN to predict the landmarks on beetles images. The model is evaluated on five datasets corresponding five parts of the beetle: left mandible, right mandible, pronotum, body, and head. For each dataset, the model has been trained in several times with different images data. Then, the trained model is evaluated with the corresponding test set. At the end, the coordinates of the landmarks on all the images in each dataset have been predicted. Three correlation methods have been used to calculate the coefficient between manual landmarks and predicted landmarks. Besides, a statistic based on the distance between manual and predict landmarks is also calculated. The statistic accepts the predicted landmark that has the distance (corresponding manual and itself) less than the average value (of all images). From two evaluation ways, the coefficients are enough good to precise when we consider the statistic problem. But, when we stay on the side of the image, the results are not good as we expect.

References

[1] Sander Dieleman, Jan Schlter, Colin Raffel, Eben Olson, Sren Kaae Snderby, Daniel Nouri, et al. Lasagne: First release., August 2015.

[2] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[3] Julie Pallant. *SPSS survival manual.* McGraw-Hill Education (UK), 2013.

[4] Jerome L Myers, Arnold Well, and Robert Frederick Lorch. *Research design and statistical analysis.* Routledge, 2010.

[5] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

Appendix: A comparison on parameters of the networks

The number of layers in the model are shown in Table 13.

Model	N^o layers	Input size	N^o CONVs	N^o POOLs	N^o Dropout	N^o FC
model	13	$1 \times 256 \times 192$	6	3	4	3

Table 13: The number of layer types in each model

The detail parameters in each layer of the models are shown in Table 14.

layers	model
input	$1 \times 256 \times 192$
layer 1	CONV(32,3,1,0)
layer 2	POOL(2,2,0)
layer 3	**DROP(0.1)**
layer 4	CONV(64,2,1,0)
layer 5	POOL(2,2,0)
layer 6	**DROP(0.2)**
layer 7	CONV(128,2,1,0)
layer 8	POOL(2,2,0)
layer 9	**DROP(0.3)**
layer 10	FC(1000)
layer 11	**DROP(0.5)**
layer 12	FC(1000)
layer 13	FC(32/36/16/22/20)

Table 14: The parameters at each layer of the model

Which:

- CONV(x,y,z,t): convolutional layer with the parameters: x = *number of filters*, y = *size of filter matrix*, z = *stride value*, t = *padding value*
- POOL(y,z,t): maximum pooling layer with: y = *size of filter*, z = *stride value*, t = *padding value*
- DROP(p): dropout layer with p *is the dropout ratio*
- FC(x): full-connected layer with x *is the number of output*