Elementary Blocks Network to landmark anatomical images

Le Van Linh[a,c,*], Beurton-Aimar Marie[a,1], Zemmari Akka[a], Parisey Nicolas[b,1]

[a] *University of Bordeaux, 351, cours de la Libration, 33405 Talence, France*
[b] *UMR 1349 IGEPP, BP 35327, 35653 Le Rheu, France*
[c] *Dalat University, Dalat, Lamdong, Vietnam*

Abstract

Deep learning has been introduced in the previous century for artificial intelligence program, and in recent years, it has risen strongly because of improvements in the computation performance. It has been applied to solve problems in different domains such as computer vision, speech recognition, or languages translation. Among different types of deep learning architectures, Convolutional Neural Networks have been most often used in computer vision for image classification, object recognition, or key points detection and they have brought amazing achievements. In this work, we propose a new Convolutional Neural Network model based on composition of elementary blocks of layers to predict key points (landmarks) on 2D anatomical biological images. Our proposed model has been trained and evaluated on a dataset including the images of 5 parts of 293 beetles. During the experiments, the network has been tested in two ways: training from scratch and applying fine-tuning process. To lead the pre-training step, a large public dataset of keypoints on human faces has been used. The obtained parameters have been then inserted to re-train the model on beetle's images. The quality of predicted landmarks is evaluated by

[*] Corresponding author

Email addresses: `van-linh.le@labri.fr` (Le Van Linh), `beurton@labri.fr` (Beurton-Aimar Marie), `zemmari@labri.fr` (Zemmari Akka), `nicolas.parisey@inra.fr` (Parisey Nicolas)

[1] both authors contributed equally to this work.

comparing the coordinates distance between predicted landmarks and manual ones which have been set by biologists. The final results have been delivered to biologists and they have confirmed that the quality of predicted landmarks is statistically good enough to replace the manual landmarks for most of the different morphometry analyses.

Keywords: Deep learning, CNN, fine-tuning, landmarks

---

1. Introduction

In recent years, deep learning which is a part of machine learning is known as a solution for difficult tasks in different domains[1]. Computational model of deep learning is composed of multiple layers to learn data representation. Each layer extracts the representation of input data which comes from the previous layers, then it will compute a new output to the next layer. In a deep learning model, each layer may contain different number of nodes, called *neurons* which have been inspired from the biological neural system [2]. Currently, deep learning has many kinds of variant architectures and each of them has found success such as: Deep Neural Network (DNN) to solve classification or data analysis problems[3, 4]; Convolutional Neural Network (CNN) in computer vision [5, 6, 7]; Recurrent Neural Network (RNN) on time sequences analysis [8, 9, 1, 10]. All of them have exhibited impressive performance comparing to more classical methods. Considering deep learning architectures, CNN is a specific one for pre-processing data which have grid topology, for examples, time series (1-D), 2D and 3D images, or video. From the first architecture [5] until now, many CNN models have been proposed and have succeeded in different tasks of computer vision such as image classification [5, 6, 7], object recognition [7, 11, 12], and key points detection [13, 14, 15, 16].

In computer vision, key points detection is an important field. In this field, algorithms try to find the key points , called Points of Interest (PoI) or landmarks through images. The landmarks are considered as the points in the image that are invariant when the scene changes e.g. by some perspective projections. In biology, the landmarks are most often manually annotated on digital images by biologists. Depending on the objective of work and the studied object, the number of landmarks may be different and their positions can be defined along the outline of the object or inside the object. From landmarks coordinates, it is possible to extract object features and to apply measure, for examples, to detect human face [14], human pose [17] or measures of an organism anatomy.

In this work, we propose a new composition of layers for a CNN architecture, Elementary Blocks Network (EB-Net), to predict the landmarks on biological species images. This model has been trained on a dataset of 5 parts of 293 beetles images: pronotum, head, elytra, left and right mandible. We have also designed a specific procedure to augment our dataset because several hundred images are usually considered as a modest number to apply deep learning methods. Finally in order to boost the prediction we have applied transfer-learning procedure with the help of a public human face database and fine-tuned our parameters model. The biologists have asserted that at the end of this process, the predicted landmarks were statistically speaking good enough, in most of the cases, to replace the manual landmarks.

This paper is organized as followed: Section 2 presents the related works about deep learning and setting of landmarks on 2D images. Section 3 describes the method to augment our dataset. Section 4 explains the design of new network model. The first experiments of the network on each dataset are presented in Section 5, and the last section delivers all the final results including improvements provided by the fine-tuning procedure.

2. Related works

In the middle of the previous century, deep learning has been introduced by Lecun [1] as a method for artificial intelligence applications. However, several problems appeared in order to take into account real-world cases because of the limitation of the memory size or computing power. Nowadays, huge improvements of computing capacities, both in memory size and in computing time with GPU programming, have opened a new perspective for deep learning. In recent years, deep learning architectures have achieved remarkable accomplishments in many domains such as computer vision [5, 6, 7, 11, 12, 18], speech recognition [4, 3], language translation [8, 9], natural language processing [1, 10, 19]. Especially in computer vision, deep learning, specifically with CNN, has been used to achieve difficult tasks in image analysis such as image classification, objects or key points detection.

2.1. Overview of Convolutional Neural Network

A CNN is a feedforward network which takes the information following from the inputs to the outputs. Currently, CNNs have many variations, but it generally consists of several types of layers, as, convolutional, pooling or fully connected layer. Fig. 1 shows an overview of CNN network which inputs directly an image to several stages of convolutional and pooling layers. Then, the representation is feed into three fully connected layers. A dropout layer is inserted after the second fully connected layer to drop some nodes during the training process (blue nodes). Finally, the last fully connected layer gives output as the category label for the initial input image. This architecture could be seen as the most popular one. Reader interested in order details could read the review of Gu et al. [20].

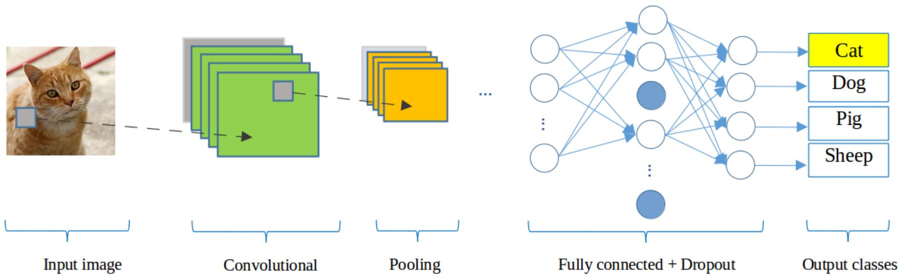


Figure 1: A CNN network for classification problem

2.2. State of the arts in deep learning and key points detection

LeNet [5] model is considered as the first architecture of CNN. LeCun et al. [5] have used it to classify the handwritten digits in cheques. LeNet exhibits a standard architecture of a CNN which consists of 2 convolutional layers, pooling layers, followed by two fully connected layers. But to be applied to realistic problems, this model requires huge computation capacities and large amount of training data which were hardly available in the early 2000s. In the last ten years, the computing capabilities have drastically improved while, in the same time, a huge amount of data became available, new models of neural networks appeared well adapted to this new environment. One of the first ones is AlexNet [6], which is similar to LeNet [5] but with a deeper structure: LeNet has 2 convolutional layers and 1 fully connected layer while AlexNet has 5 and 3, respectively. Furthermore, in AlexNet the activation functions have been changed and dropout layers have been added to prevent the over-fitting. AlexNet won the famous ImageNet Challenge[2] in 2012. From the success of AlexNet, a lot of different models have been proposed to improve the performance of CNN, one can cite ZFNet [21], GoogLeNet [7], VGGNet [22], or ResNet-50 [23]. The main difference between these networks is that their architectures became deeper and

[2]This is a challenge where evaluates algorithms for object detection and image classification.

deeper by adding more layers, e.g. ResNet-50, which won the champion of ILSVRC 2015, is deeper than AlexNet around 20 times.

Besides classification or recognition of objects, CNNs have been also used to detect key points inside images. Liu et al. [13] have presented a method to predict the positions of functional key points on fashion items such as the corners of neckline, hemline and cuff. Yi Sun et al. [14] have proposed a CNNs cascade to predict the facial points belonging to the human face. Their model contains several CNNs which are linked together in a list as a cascade. Three levels of the cascade are set to recognize the human face from the global to local view with the objective to increase the accuracy of predicted key points. In the same topic, Zhanpeng Zhang et al. [15] have proposed a *Tasks-Constrained Deep Convolutional Network* to join facial landmarks detection problem with a set of related tasks, e.g. head pose estimation, gender classification, age prediction, or facial attribute inference. In their method, the input features have been extracted by 4 convolutional layers, 3 pooling layers and 1 fully connected layer which is shared by multiple tasks in the estimation step. Shaoli Huang et al. [17] have introduced a coarse-fine network to locate key points and to estimate human poses. Their framework consists of the base convolutional layers shared by two streams of key point detectors: the first stream, named coarse stream, includes 3 detector branches (3 stacks of Inception modules [7]) which are used to focus on capturing local features and modeling spatial dependencies between human parts. The second one, named fine stream, receives the features which are concatenated from the coarse stream and provides accurate localization. Cintas et al. [16] have introduced an architecture which is enable to recognize 45 landmarks on human ears. Their model includes a structure with 2 convolutional layers, 1 pooling layers, and 1 dropout layer to extract the features. This structure is repeated 3 times and is followed by 3 fully connected layers. In the same context of key point detection, we have developed a CNN to automatize landmarks prediction on beetle's anatomies but before to describe it, we will present the augmentation procedure that we have defined for our dataset.

3. Data augmentation

From AlexNet period to ResNet-50, the obtained success stories [6, 23] have proved that CNN models produce better results on a large dataset but to use this technique, the size of dataset remains a bottleneck and our own some hundreds of images are considered as modest for these models. So, it is important to be able to provide a large dataset in order to learn more cases and improves the learning ability of the network. Unfortunately, in some application domains as this work in biology, providing a large dataset is too costly. For this reason, one way to solve this problem is to create misshapen data from real data and to add them to the training set. Most often in image processing, dataset augmentation uses operations like translation, rotation or scaling which are well known to be efficient to generate new version of existing images. However, this kind of operations are not useful in our case because the analysis of images by CNN (convoluted) are usually invariant to translation or rotation. So, we prefered to rely on method changing color space values to obtain misshapen images.

Our image set is in RGB color map, the first procedure consists of changing the value of one color channel of the three channels in the original image to generate a new image. A constant value is sampled in an uniform distribution $\in [1, N]$ to obtain a new value caped at 255. For example, Fig. 2 shows the three images which are generated when a constant $c = 10$ is added to each channel of an original image. Following this way, we can generate three new versions of only one image.

In the second procedure, each channel is considered separately and one gray

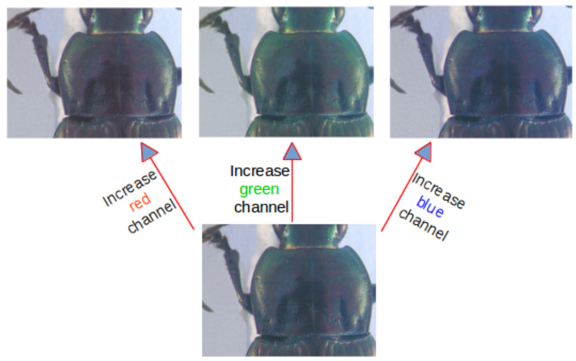


Figure 2: A constant $c = 10$ has been added to each channel of an original image

image is generated for it (Fig. 3). Consequently, we obtain 3 new images (single channel) from an original one. At the end of the process, 6 versions of an original image are made. In total, the new data set contains $293 \times 7 = 2051$ images for each anatomical part of beetle (an original image and six misshapen ones).

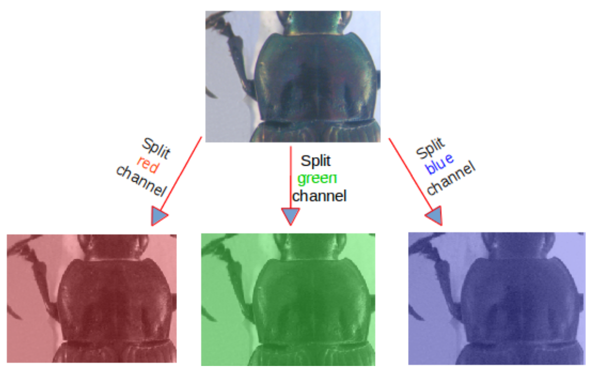


Figure 3: Three channels (red, green, blue) are separated from original image

4. Network architectures designing

As we have presented previously, several CNN architectures are available from literature and tools libraries. It is always possible to adapt them to a specific application by changing the parameters values or by modifying the arrangement of layers. Indeed, several trials have been achieved before to obtain a satisfying model dedicated to landmarks estimation. In this section, we present three versions of the model that we have designed to solve this task. As usual, we have combined the classical layer types to build the model, i.e., convolutional, maximum pooling, dropout, and full-connected layers.

The first architecture has been a very classical one (Fig. 4). It receives an input image with the size of $1 \times 192 \times 256$, then it is composed by 3 repeated structures of a convolutional (CONV) layer followed by a maximum pooling (POOL) layer. In most of CNNs, the parameters of CONV layers have been set to increase the depth of the feature maps from the first to the last layer. This is done by setting the number of filters at each CONV layer. In this first model, the depths of the CONV layers increase from $32, 64$, to 128 and with different size of the kernels: 3×3, 2×2 and 2×2, respectively. Inserting POOL layers after a convolutional layers is usually done. The POOL layer progressively reduce the spatial size of the representation, reduce the number of parameters, computation in the network, and also to prevent over-fitting. The operations of POOL layers are independent for each depth slice of their inputs. In our model, we have used the most common form for one POOL layer: a filter with size of 2×2 and a stride equal to the size of filter have been applied. At the end of the model, 3 FC layers have been added to extract the global relationship between the features and to proceed the outputs. The first two FC layers have been applied the activation functions to make sure these nodes interact well and to take into account all possible dependencies at the feature level. The outputs of the FC layers are $500, 500$ and 16. The output of the last FC layer corresponds to the coordinates (x and y) of 8 landmarks which we would like to predict on pronotum part. Nevertheless, the obtained results with this architecture has not been considered good enough to continue to use it. One of the main problems is the presence of over-fitting during the training process (Detailed results will be discussed in Section 5).

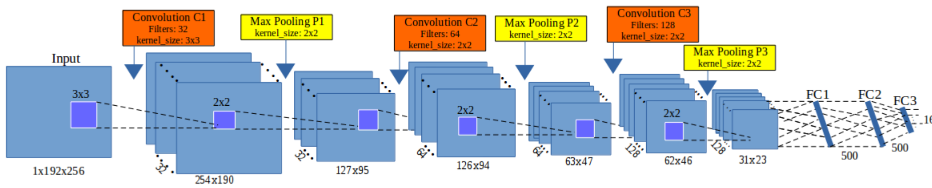


Figure 4: The architecture of the first model

The second model has kept the same architecture of the first model but the number of output of the two FC layers has been increased to 1000. Increasing the value at FC layers could allow to get more features from CONV layer without requirements of computing resources. However, the obtained results remained not satisfying, it will be discussed also in the result section (Section 5).

To build the third architecture, we have defined the *Elementary Block* (EB). An EB is defined as a sequence of 1 CONV (C_i), 1 maximum POOL (P_i) and 1 dropout (D_i) layers (Fig. 5). The dropout layer has been added to prevent overfitting by adding a step to remove some nodes. This has significantly reduced overfitting and over performed the other regularization methods [24].

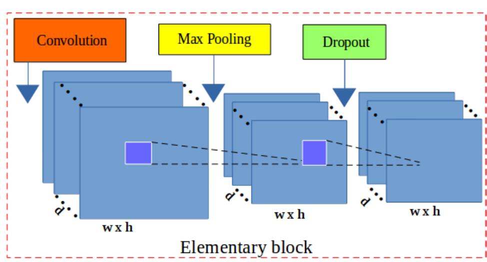


Figure 5: The layers in an elementary block. It includes a CONV layer (red), a maximum POOL layer (yellow) and a DROP layer (green).

Fig. 6 illustrates the structure of the third architecture. For our purpose, we have assembled **3 elementary blocks** which are the main components of **EB-Net**. The parameters for each layer in each elementary block are described as below, the list of values follows the order of elementary blocks ($i = [1..3]$):

- CONV layers:
  - Number of filters: $32, 64$, and 128
  - Kernel filter sizes: $(3 \times 3), (2 \times 2)$, and (2×2)
  - Stride values: $1, 1$, and 1
- POOL layers:
  - Kernel filter sizes: $(2 \times 2), (2 \times 2)$, and (2×2)
  - Stride values: $2, 2$, and 2
- DROP layers:
  - Probabilites: $0.1, 0.2$, and 0.3

For the FC layers, FC1 and FC2 have 1000 outputs, the last FC layer (FC3) has 16 outputs. As usual, a dropout layer is inserted between FC1 and FC2 with a probability equal to 0.5.

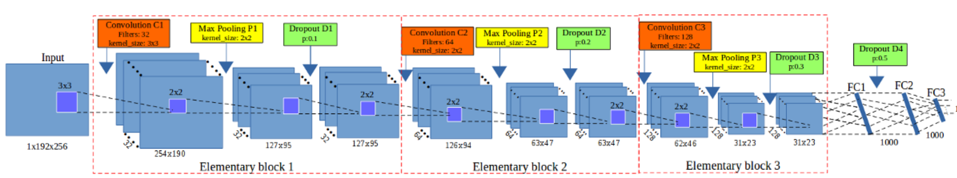


Figure 6: The architecture of EB-Net

The core of CNN is training over iteration. There are many ways to optimize the learning algorithm, but gradient descent [25] is currently a good choice to reduce the loss in neural network. The core idea is to follow the gradient until reaching a minimum of the cost function. So, we have chosen gradient descent in the backward phase to update the values of learnable parameters. EB-Net is designed to use a learning rate initialize at 0.03 and to stop at 0.00001, while the momentum is updated from 0.9 to 0.9999. The values are updated over training time to fit with the number of epochs [3] by applying parameters adjustment during the training. The architecture implementation has been written in Lasagne framework [26] in Python code. More information about the model can be obtained from the repository on GitHub: https://github.com/linhlevandlu/CNN_Beetles_Landmarks

5. Experiments and results

This work is a part of a project about automatized morphology measurements. Fig. 7 presents an example of each part of our dataset. In these parts, the left and right mandible are considered as the segmentable parts while other parts are high difficulty to segment. In the previous work, we have proposed a method based on image processing techniques to work with segmentable images [27]. The choice to turn to deep learning process has been motivated by the high difficulty to segment some parts of the beetle images and consequently to apply classical image processing methods. The pronotum was the first part we have analyzed with deep learning. images5parts

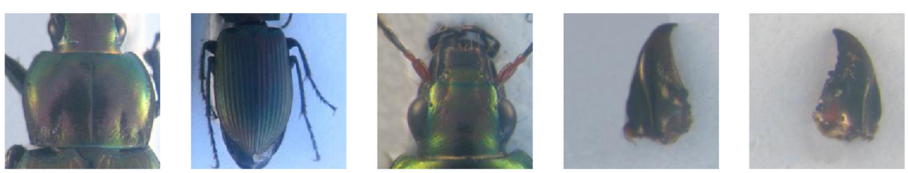


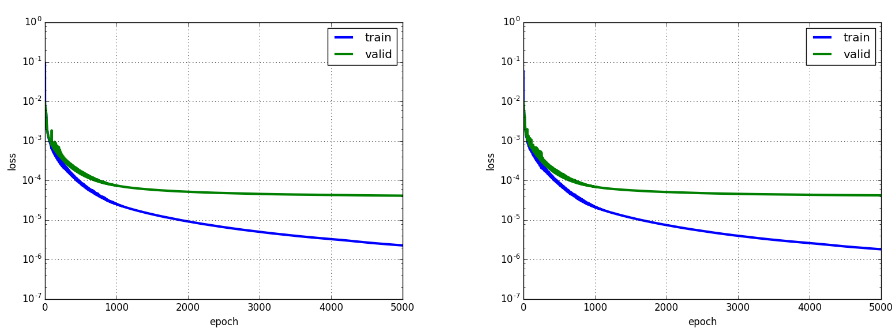
Figure 7: The image in each part of beetle. From the left to right: pronotum, elytra, head, left and right mandible.

The networks have been trained in 5,000 epochs on Linux OS by using NVIDIA TITAN X cards. During the training, the images are chosen randomly from the dataset with a ratio of 60% for training and 40% for validation. For

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

each image, a set of 8 manual landmarks are available. They have been set by biologists and are considered as the ground truth for the evaluation. In deep learning, many kinds of loss expressions can be considered depending on the class of problem solving by the network, for example, Root Mean Square Error (RMSE) is usually used for regression problems where the outputs are not discrete values. In our work, landmark prediction can be seen as a regression problem because the coordinates of landmarks do not belong to discrete classes.

In order to extract predicted landmarks from all pronotum images, we have applied *cross-validation* procedure to choose the test images, we call it *round*. For each round, we have decided to choose 33 images for testing step. In order to predict landmarks for all available images, we will do 9 rounds. The 260 remaining images are used as training and validation images. Of course, this dataset will be augmented as described before to provide 1820 images for these 2 steps.



(a) The first model

(b) The second model

Figure 8: The losses (training and validation) of the models. The blue curve presents the RMSE errors of training process while green curve is the validation errors.

Fig. 8a shows the training errors and the validation errors during training phase of the first architecture. The blue curve presents the RMSE errors of training process while green curve is the validation errors. Clearly, over-fitting has appeared in the first model, i.e., training losses are able to decrease but validation losses are stable. In the second model (Section 4), the parameters of full-connected layers have been modified to prevent the over-fitting but it seems that this solution is still not satisfying, the results are very similar to the previous ones (over-fitting still appears).

Fig. 9 illustrates the losses during the training of EB-Net, one can note that after several epochs, the two-loss values become close and the over-fitting disappears. The sequence of Dropout addition inside elementary block works well to prevent over-fitting and improve the accuracy of the model greatly. This third model has been selected to compute automatically landmarks.

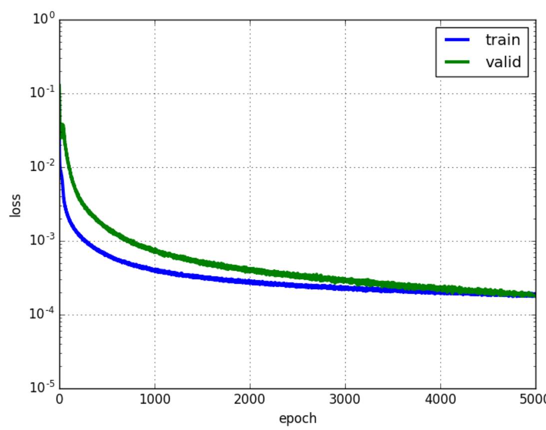


Figure 9: The losses (training and validation) of EB-Net

Table. 1 resumes the losses of 9 rounds when we trained EB-Net on pronotum images. Clearly, the training/validation loss among rounds are tiny and stable.

To evaluate the coordinates of predicted landmarks, the Pearson correlation metric between predicted and manual landmarks has been calculated for each dimension (x and y). Table 2 shows the obtained results. The average value of the coordinates correlation (both x and y) is in the first row, standard error, minimun and maximum of correlation scores are given in the next raws. One

Round	Training loss	Validation loss
1	0.00018	0.00019
2	0.00019	0.00021
3	0.00019	0.00026
4	0.00021	0.00029
5	0.00021	0.00029
6	0.00019	0.00018
7	0.00018	0.00018
8	0.00018	0.00021
9	0.00020	0.00027

Table 1: The losses during training the third model on pronotum images

can note that the correlation is strongly positive in each case with a very small standard error, proving that each individual coordinate is well predicted.

	X-dimension	Y-dimension
Mean	0.8116	0.9438
Std. Err.	0.002	0.0006
Min	0.7474	0.9063
Max	0.8577	0.9638

Table 2: Statistical indicators on Pearson correlation between manual and predicted landmarks

Standing on the side of the users, biologists would like to obtain an acceptable position of the landmark when they look at the images. So, the distances (in pixels) between manual coordinates and predicted ones have been calculated for all images. Then, the average distance for each landmark has been computed. Table. 3 shows the average distances by landmarks on all images of pronotum dataset. With the images resolution 256×192, we can consider that an error of 1% (corresponding to 2 pixels) could be an acceptable error. Unhappily, our results exhibit average distance of 4 pixels in the best case, landmark 1 and more than 5 pixels in the worse case, landmark 6.

Fig. 10 shows the distribution of the distances on the first landmark of all images. The accuracy based on the distance in each image can be separated

Landmark	**Distance** (in pixels)
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 distances on all images per landmark on pronotum images.

into three spaces: best results, the images having distance less than the average value (4 pixels): 56.66%; acceptable results, the images having the distance from the average value to 7 pixels (standard deviation error): 31.40%; and the images which are clearly in error with the distance greater than 7 pixels: 11.94%.

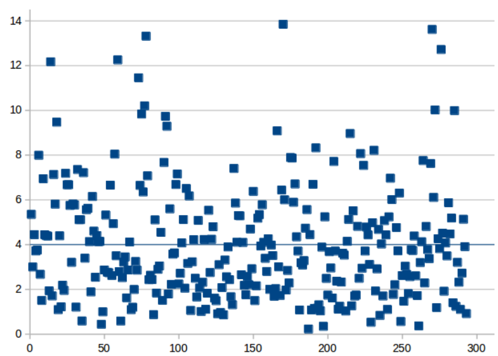
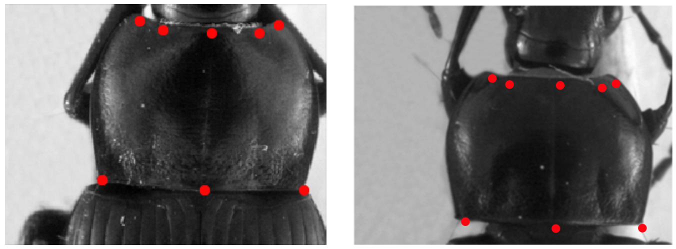


Figure 10: The distribution of the distances on the first landmark. The blue line is the average value of all distances.

To illustrate this purpose, Fig. 11 shows the predicted landmarks on two test images. One can note that even some predicted landmarks (Fig. 11a) are close to the manual ones, in some case (Fig. 11b) the predicted ones are far from the expected results. So, the next step has been dedicated to the improvement of these results.



(a) Image with well-predicted landmarks

(b) Image with inaccuracy landmarks

Figure 11: The predicted landmarks, in red, on the images in test set.

6. Improving results by fine-tuning

EB-Net has been trained from scratch on five datasets of beetles (left mandible, right mandible, pronotum, elytra, and head). At this step, the network was able to predict the landmarks on the images. But as we have discussed before, even if the strength of the correlation validates the results, when we display on the images, the predicted coordinates are not exactly close to the manual ones, and the average distances are a little bit high.

Training a network from scratch is not the only way to work in Deep learning. It is possible to initialize parameter values by extracted values from another experiment on another dataset. It is called transfer learning [28]. In tranfer learning, the obtained parameters values of model which have been used to solve a problem, are reused on other datasets [29] and potentially to solve another task. The name of this procedure is currently called **fine-tuning**.

Fine-tuning does not only replace and retrain the last layer of the model on the new dataset but also tunes the weights of a trained model by continuing the backpropagation. In this context, ImageNet [30] is a well-known dataset with more than $100,000$ images. It has been used to train many famous CNN architectures such as AlexNet [6] or VGG-16 [22] with success. The pre-trained models on ImageNet have been then shared in deep learning community as a source to re-use features of ImageNet. Unfortunately, some preliminary tests have shown that re-using ImageNet features is not relevant for our application because landmarks dectection has a distinct difference from general object recognition, as mentioned in [31]. However, we noticed that our problem is related to face recognition and facial key points detection. So, we have decided to train EB-Net on a facial key points dataset and then, the trained parameters have been transfered to fine-tune on beetle's images.

6.1. Pre-train EB-Net on facial key points dataset

Facial Keypoints dataset has been published for a competition on Kaggle community [4]. It includes $2,140$ human face images with the size of 96×96. Each image contains 15 landmarks on the face: 6 landmarks for eyes, 4 landmarks for eyebrows, 4 landmarks for mouth, and 1 landmarks for nose tip. Fig. 12 shows four face images in the dataset and the landmarks on each face.

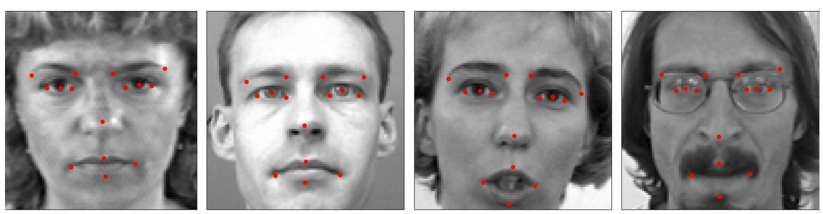


Figure 12: Four face images in the dataset and ground truth position of the landmarks.

For the pre-training step, EB-Net has been trained on facial key point dataset. The objective of this task is to evaluate and to compare the effectiveness of EB-Net and other published results in the challenge. Basically, the layer's parameters are the same when training from scratch. We have just changed the image size of network model to 96×96 and the output number of the last FC layer to correspond to the number of landmarks (15 landmarks).

[4] https://www.kaggle.com/c/facial-key points-detection

For hyper-parameters of model, the learning rate and momentum remained the same but the number of epochs has been changed to $10,000$ instead of $5,000$ to achieve better learning on the parameters. After training, the obtained RMSE score is 1.1464. This score is better than top 3 on the leader board of the challenge.

6.2. Fine-tuning on each beetle's part

EB-Net has been pre-trained on facial key points dataset with 30 outputs (15 landmarks $\times$ 2 coordinates). Then, the parameters have been transfered to be fine-tuned for the images of each beetle's part. The fine-tuning stage has been done by continuing the backpropagation to update the layers parameters. The images in Facial Keypoints dataset are squared, in order to respect that we have reduced the size of beetle's images to 192×192 by cropping a background band. The stride property of the first convolutional layer has been also modified from 1 to 2.

After finishing the fine-tuning process, EB-Net is used to predict the landmarks on test images. To evaluate the accuracy of the model's output, the distances (in pixels) between predicted and corresponding manual landmarks have been calculated. Then, the average distances for each landmark has been taked into account. Tables 4, 5, 6, 7, and 8 show the average distances by landmarks on each beetle's part of the two processes: training from scratch and fine-tuning process. **From scratch** columns remind the previously average distances when EB-Net was trained from scratch. **Fine-tune** columns present the new average distances after applying fine-tuning on each part. The green and red values represent the best and the worst average distances on each part. From these tables, one can note a difference in average distances between the two processes, the average distances of each landmark have decreased in the case of fine-tuning. It is clearly proved that the quality of predicted landmarks with the help of fine-tuning is better than training from scratch.

#LM	From scratch	Fine-tune
1	**4.00**	2.99
2	4.48	3.41
3	4.30	2.98
4	4.39	3.54
5	4.29	3.37
6	**5.36**	**4.06**
7	4.64	**2.93**
8	4.94	3.64

Table 4: Average distances comparison between training from scratch and fine-tuning on pronotum images

In order to get a better view of the results, we have considered that average values can hidde different case of errors. So, we have calculated other statistical indicators such as median, standard error, minimum value and maximum values to appreciate of dispersion of points. All these statistical values are presented in Appendix A. From these tables, we can see the minimum and the maximum distances have a large range of values. However, the median values, which separate the set into two parts, are very close with minimum values and so far from maximum values, even smaller than the mean distances. It confirms that almost all distances stay around the median values and the predicted landmarks are good enough to replace the manual ones. Besides, the distribution of the distances on each landmark of each part have been taked into account in Appendix B. We can observe that most of distances are close to the mean and median values, only some exceptional cases are really far. Fig. 13 shows the predicted landmarks of fine-tuning process in one case of each part.

The fine-tuning process has improved the results of the proposed architecture on both 5 datasets: left, right mandible, pronotum, elytra and head. All the average distances have significantly decreased: $\approx$ 41.35% on left mandible, $\approx$ 46.51% on right mandible, $\approx$ 25.98% on pronotum, $\approx$ 15.8% on elytra, and

#LM	From scratch	Fine-tune
1	**5.53**	**4.82**
2	5.16	4.21
3	5.38	4.73
4	5.03	4.11
5	**4.18**	**2.76**
6	4.45	3.50
7	4.79	3.92
8	4.53	3.40
9	5.14	4.17
10	5.06	3.94

Table 5: Average distances comparison between training from scratch and fine-tuning on head images

#LM	From scratch	Fine-tune
1	**3.87**	3.21
2	3.97	3.28
3	3.92	**3.20**
4	**3.87**	3.22
5	4.02	3.31
6	4.84	4.21
7	5.21	4.54
8	**5.47**	**4.76**
9	5.27	4.55
10	4.07	3.39
11	3.99	3.29

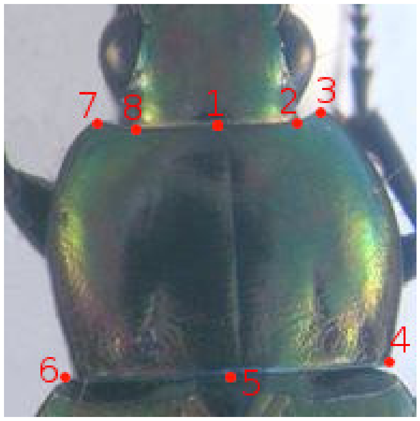
Table 6: Average distances comparison between training from scratch and fine-tuning on elytra images

#LM	From scratch	Fine-tune
1	**9.13**	5.28
2	**6.72**	4.05
3	6.87	4.01
4	6.77	4.02
5	7.13	3.92
6	6.94	**3.88**
7	7.32	4.01
8	7.41	4.16
9	7.58	4.35
10	7.63	4.46
11	7.69	4.72
12	8.42	5.08
13	7.99	4.50
14	7.49	4.26
15	7.79	4.62
16	8.52	**6.04**

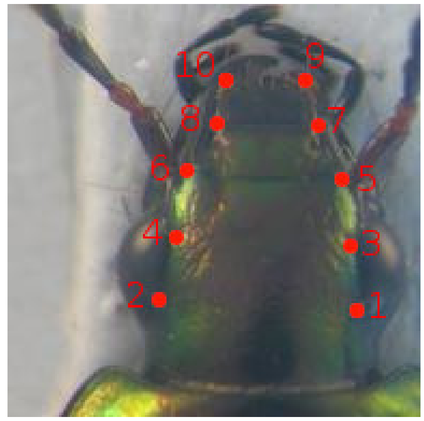
Table 7: Average distances comparison between training from scratch and fine-tuning on left mandible images

#LM	From scratch	Fine-tune
1	**9.50**	4.86
2	7.17	4.06
3	7.24	3.97
4	**7.04**	3.87
5	7.16	4.05
6	7.57	3.82
7	7.43	**3.77**
8	7.66	3.87
9	7.79	3.96
10	8.02	3.97
11	8.31	4.27
12	8.16	4.42
13	8.89	4.87
14	9.18	4.93
15	8.79	4.46
16	8.31	4.17
17	8.29	4.57
18	8.89	**5.89**

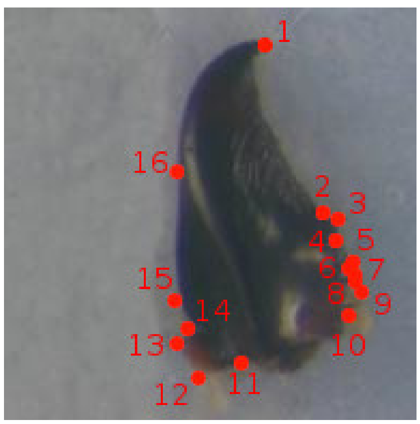
Table 8: Average distances comparison between training from scratch and fine-tuning on left mandible images



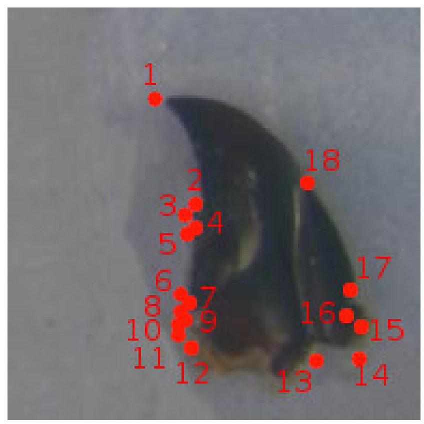
(a) Pronotum



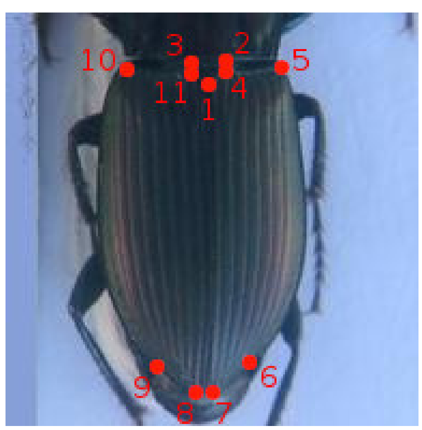
(b) Head



(c) Left mandible



(d) Right mandible



(e) Elytra

Figure 13: The location of predicted landmarks (red points) in one case of each part

$\approx 18.10\%$ on head part. Besides, if we consider a predicted landmark, which has the distance (from manual ones) less than mean value plus standard deviation, is acceptable, the accuracy of method on each part is **87.07%** on pronotum, **87.92%** on head, **91.78%** on elytra, **93.58%** on left mandible and **88.31%** on right mandible.

We also have a comparison between the results of deep learning and early methods where we have applied image processing techniques to predict the landmarks [27]. Clearly, the result with fine-tuning has improved the location of estimated landmarks. Even the average distances obtained from scratch training are still higher but they are more stable than the results from the early method: most of the average distance(or landmarks) of left mandibles are less than the results of the early method, while the average distances are very close in the case of right mandibles.

7. Conclusion

In this work, we have presented how to apply CNN to predict the landmark on 2D anatomical images of beetles. After going through many trial models, we have presented EB-Net for automatic detection landmarks on anatomical images of beetles which includes the repetition of some elementary blocks, followed by fully connected layers. We have applied transfer learning to fine-tune the model. A public facial keypoints dataset has been used to pre-train EB-Net.

In our case, the size of dataset is limitting. Therefore, we have applied image processing techniques to augment dataset. The predicted landmarks have been evaluated by calculating the distance between manual landmarks and corresponding predicted landmarks. Then, the average of distance errors on each landmarks has been considered.

The results have shown that using the convolutional network to predict the landmarks on biological images leads to satisfying results without need for time consuming human intervention (i.e. semi-automatic methods) or lots of pre- and post- processing of the digital images that will be only add hyperparameters and/or more errors to the predicted landmarks. The best set of estimated landmarks has been obtained after a step of fine-tuning using the whole set of images that we have for the project, i.e. about all beetle parts. The quality of prediction allows using automatic landmarking to replace the manual ones.

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[2] M. A. Arbib, Brains, machines, and mathematics, Springer Science & Business Media, 2012.

[3] G. Hinton, et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal Processing Magazine 29 (6) (2012) 82–97.

[4] T. Mikolov, et al., Strategies for training large scale neural network language models, in: Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on, IEEE, 2011, pp. 196–201.

[5] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[6] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[7] C. Szegedy, et al., Going deeper with convolutions, Cvpr, 2015.

[8] S. Jean, K. Cho, R. Memisevic, Y. Bengio, On using very large target vocabulary for neural machine translation, arXiv preprint arXiv:1412.2007.

[9] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, 2014, pp. 3104–3112.

[10] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, Journal of Machine Learning Research 12 (Aug) (2011) 2493–2537.

[11] C. Farabet, C. Couprie, L. Najman, Y. LeCun, Learning hierarchical features for scene labeling, IEEE transactions on pattern analysis and machine intelligence 35 (8) (2013) 1915–1929.

[12] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5325–5334.

[13] Z. Liu, S. Yan, P. Luo, X. Wang, X. Tang, Fashion landmark detection in the wild, in: European Conference on Computer Vision, Springer, 2016, pp. 229–245.

[14] Y. Sun, X. Wang, X. Tang, Deep convolutional network cascade for facial point detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 3476–3483.

[15] Z. Zhang, et al., Facial landmark detection by deep multi-task learning, in: European Conference on Computer Vision, Springer, 2014, pp. 94–108.

[16] C. Cintas, et al., Automatic ear detection and feature extraction using geometric morphometrics and convolutional neural networks, IET Biometrics 6 (3) (2016) 211–223.

[17] S. Huang, M. Gong, D. Tao, A coarse-fine network for keypoint localization, in: The IEEE International Conference on Computer Vision (ICCV), Vol. 2, 2017.

[18] M.-T. Vu, M. Beurton-Aimar, V.-L. Le, Heritage image classification by convolution neural networks, in: 2018 1st International Conference on Multimedia Analysis and Pattern Recognition (MAPR), IEEE, 2018, pp. 1–6.

[19] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th international conference on Machine learning, ACM, 2008, pp. 160–167.

[20] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, Pattern Recognition 77 (2018) 354–377.

[21] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer, 2014, pp. 818–833.

[22] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[23] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[24] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting., Journal of machine learning research 15 (1) (2014) 1929–1958.

[25] Y. A. LeCun, et al., Efficient backprop, in: Neural networks: Tricks of the trade, Springer, 2012, pp. 9–48.

[26] S. Dieleman, et al., Lasagne: First release. (Aug. 2015). `doi:10.5281/zenodo.27878`.
URL `http://dx.doi.org/10.5281/zenodo.27878`

[27] V. L. Le, M. Beurton-Aimar, A. Krahenbuhl, N. Parisey, MAELab: a framework to automatize landmark estimation, in: WSCG 2017, Plzen, Czech Republic, 2017.
URL `https://hal.archives-ouvertes.fr/hal-01571440`

[28] L. Torrey, J. Shavlik, Transfer learning, Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques 1 (2009) 242.

[29] J. Margeta, et al., Fine-tuned convolutional neural nets for cardiac mri acquisition plane recognition, Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization 5 (5) (2017) 339–349. `arXiv:https://doi.org/10.1080/21681163.2015.1061448`, `doi:10.1080/21681163.2015.1061448`.
URL `https://doi.org/10.1080/21681163.2015.1061448`

[30] J. Deng, et al., ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR09, 2009.

[31] S. Lin, Z. Zhao, F. Su, Homemade ts-net for automatic face recognition, in:

Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, ACM, 2016, pp. 135–142.

Appendix A. Statistic information on each beetle's part

Table A.9, A.10, A.11, A.12, and A.13 show the statistical values on each part. The green and red numbers represent the best and the worst values on each statistical indicators, respectively.

#LM	Mean	Standard Error	Median	Minimum	Maximum
LM1	2.9914	**0.1057**	2.7031	**0.23**	14.2496
LM2	3.4066	0.1306	2.9626	0.175	18.4053
LM3	2.9829	0.1205	2.5864	0.216	19.2092
LM4	3.5449	0.1422	3.117	0.1638	**22.8899**
LM5	3.3675	0.1327	2.9741	**0.101**	17.4586
LM6	**4.0611**	**0.1512**	**3.5733**	0.1733	**14.0745**
LM7	**2.9274**	0.1159	**2.5703**	0.2263	14.092
LM8	3.6448	0.145	3.0116	0.1647	15.4585

Table A.9: The statistical values on pronotum images

#LM	Mean	Standard Error	Median	Minimum	Maximum
LM1	**4.8185**	0.1709	4.2951	0.3732	21.1819
LM2	4.2098	**0.1715**	3.7484	0.2072	**23.9351**
LM3	4.7286	0.1705	**4.3991**	0.2719	**19.12**
LM4	4.1071	0.1701	3.6232	0.1942	21.6451
LM5	4.1769	0.1545	3.7967	0.2683	20.2307
LM6	3.4976	0.1657	2.9338	0.2384	22.6836
LM7	3.9168	**0.1477**	3.4284	0.2134	21.0319
LM8	**3.402**	0.1486	**2.7877**	**0.1478**	21.233
LM9	4.1703	0.1481	3.7181	**0.4441**	22.0267
LM10	3.9433	0.1574	3.4147	0.152	20.7223

Table A.10: The statistical values on head images

#LM	Mean	Standard Error	Median	Minimum	Maximum
LM1	3.2081	0.179	2.6311	0.1265	32.6688
LM2	3.2842	0.1872	2.5934	0.1607	33.9982
LM3	**3.1975**	**0.1755**	2.5412	0.0763	31.0928
LM4	3.225	0.1812	**2.479**	0.1485	33.1458
LM5	3.3062	0.1869	2.606	0.1187	**35.7959**
LM6	4.2069	0.1957	3.578	0.2149	35.3037
LM7	4.5445	**0.2049**	4.0792	0.3454	34.7368
LM8	**4.7596**	0.2018	**4.3057**	**0.4697**	32.1749
LM9	4.548	0.1916	3.9626	0.2711	**28.3484**
LM10	3.3918	0.1772	2.7726	0.1799	29.9211
LM11	3.2897	0.1764	2.7064	**0.0527**	32.3641

Table A.11: The statistical values on elytra images

#LM	Mean	Standard Error	Median	Minimum	Maximum
LM1	5.2804	0.2805	4.2294	**0.6754**	41.9898
LM2	4.0548	0.276	3.2748	0.2977	62.6295
LM3	4.013	0.2965	**3.0758**	0.0416	72.6524
LM4	4.0203	0.2915	3.2101	**0.0167**	70.5794
LM5	3.9157	0.318	3.1796	0.2025	82.6241
LM6	**3.8781**	0.3022	3.1983	0.2125	77.8756
LM7	4.0127	0.3306	3.126	0.2276	86.2835
LM8	4.1555	0.3251	3.2471	0.2322	84.0953
LM9	4.349	**0.3521**	3.3104	0.1464	**91.2018**
LM10	4.4575	0.3105	3.6117	0.0886	79.3924
LM11	4.7191	**0.1915**	4.0415	0.4054	**27.077**
LM12	5.0797	0.2816	4.1478	0.3743	58.941
LM13	4.4999	0.3194	3.5737	0.1282	77.467
LM14	4.2572	0.2776	3.4518	0.4414	66.049
LM15	4.618	0.3165	3.811	0.1256	77.1424
LM16	**6.042**	0.3312	**4.5958**	0.1927	62.5569

Table A.12: The statistical values on left mandible images

#LM	Mean	Standard Error	Median	Minimum	Maximum
LM1	4.8759	0.2462	3.721	0.133	26.9596
LM2	4.0644	**0.1737**	3.3734	0.1778	22.6007
LM3	3.9658	0.1923	3.2037	0.1583	23.8552
LM4	3.8721	0.1823	3.2363	**0.0428**	**21.6248**
LM5	4.0479	0.2011	3.1172	**0.3983**	24.7061
LM6	3.8179	0.1847	3.1692	0.1078	35.2811
LM7	**3.7662**	0.186	**3.0912**	0.1559	34.9122
LM8	3.8728	0.1891	3.1345	0.2351	36.0385
LM9	3.9616	0.1948	3.2576	0.1376	35.3078
LM10	3.9661	0.1876	3.3955	0.1709	34.7438
LM11	4.2698	0.1919	3.6016	0.2445	36.3356
LM12	4.4238	0.205	3.7387	0.341	**38.4304**
LM13	4.8663	0.1922	4.1789	0.3772	27.2213
LM14	4.9318	0.2134	4.0853	0.1473	31.3994
LM15	4.4636	0.1975	3.5378	0.0791	28.7507
LM16	4.1737	0.1838	3.3537	0.3285	25.8165
LM17	4.566	0.1933	3.8441	0.2639	27.9728
LM18	**5.8936**	**0.2812**	**4.7034**	0.1854	30.8248

Table A.13: The statistical values on right mandible images

Appendix B. The distribution of distances on each part

In this section, the distribution of distances on each landmark of each part is shown. The red and green lines are represented for the mean and median values, respectively.

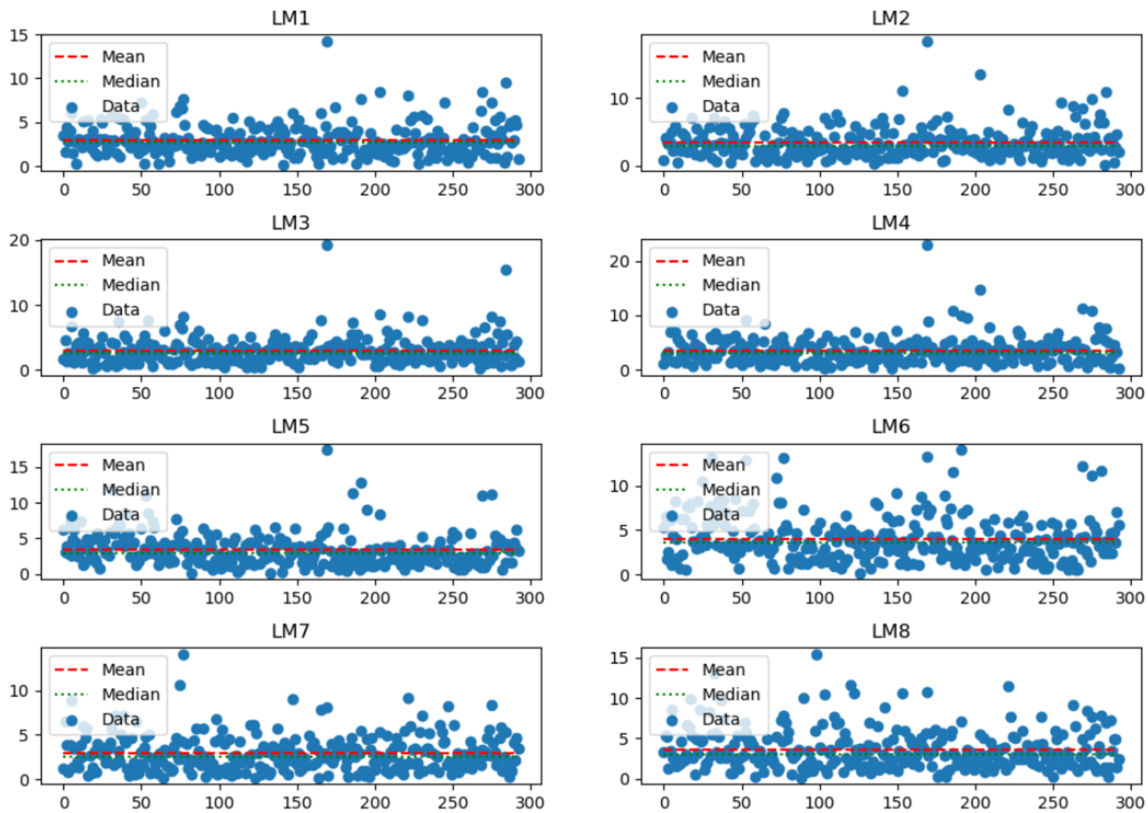


Figure B.14: The distribution of distances on each landmark on pronotum images

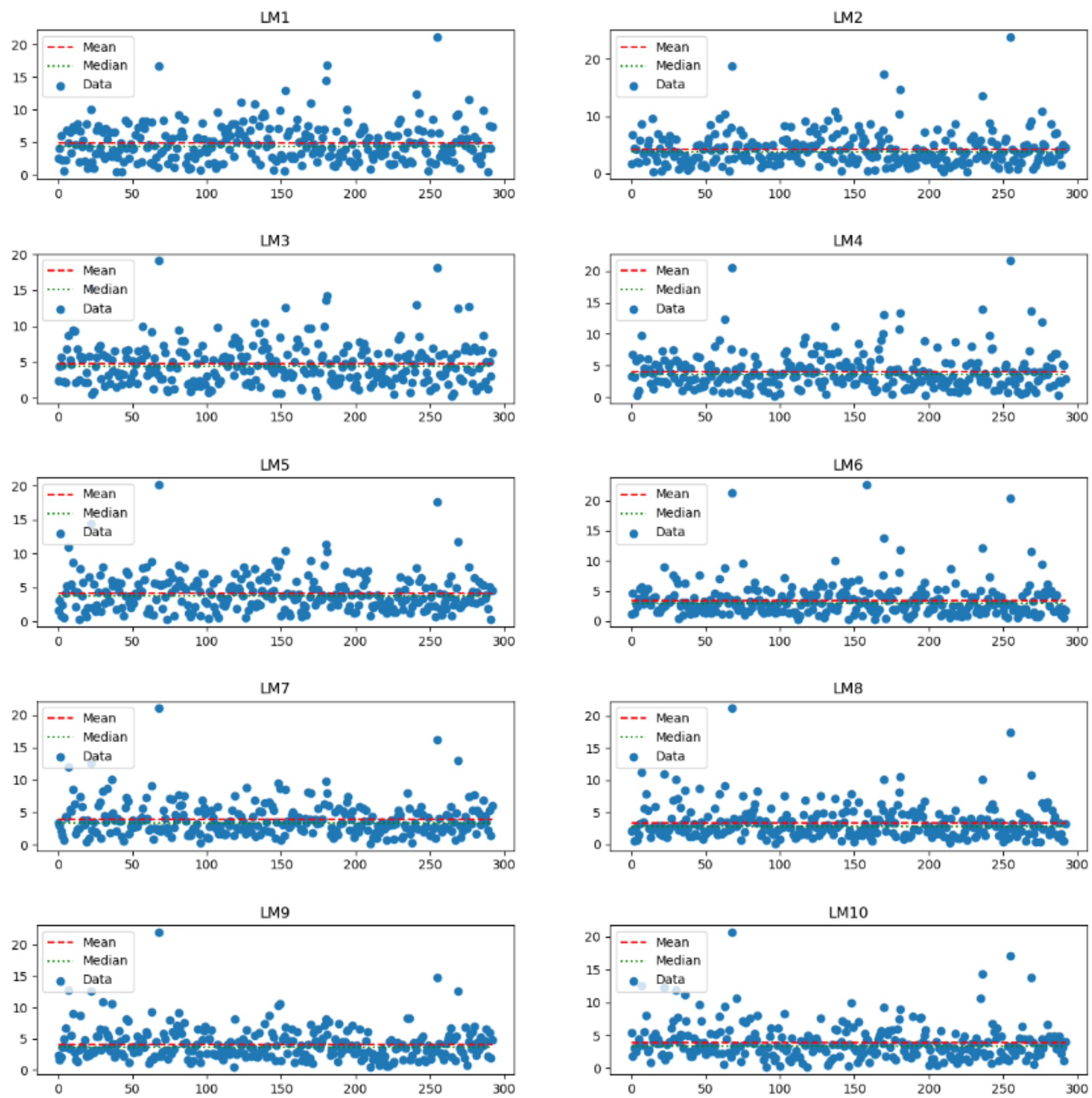


Figure B.15: The distribution of distances on each landmark on head images

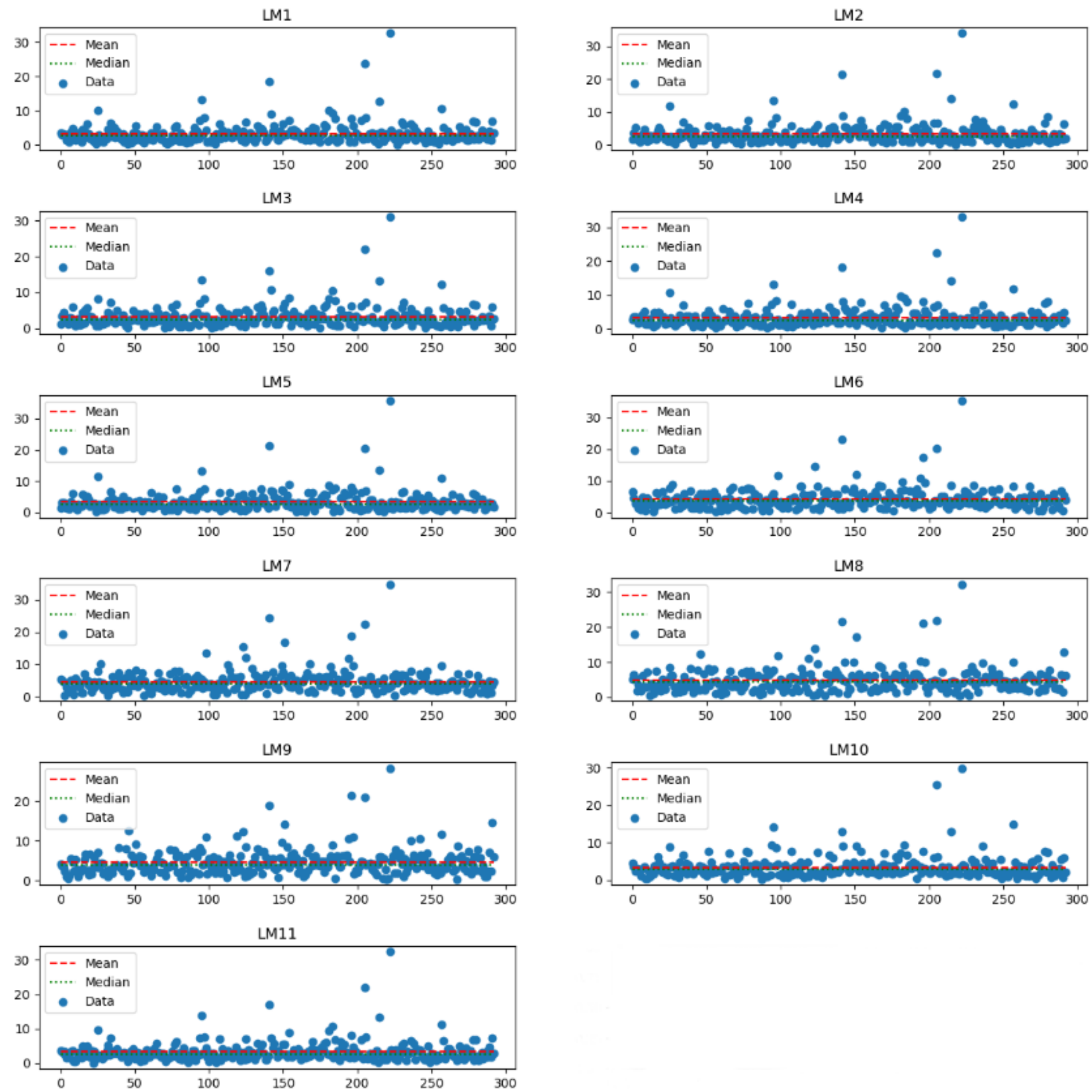


Figure B.16: The distribution of distances on each landmark on elytra images

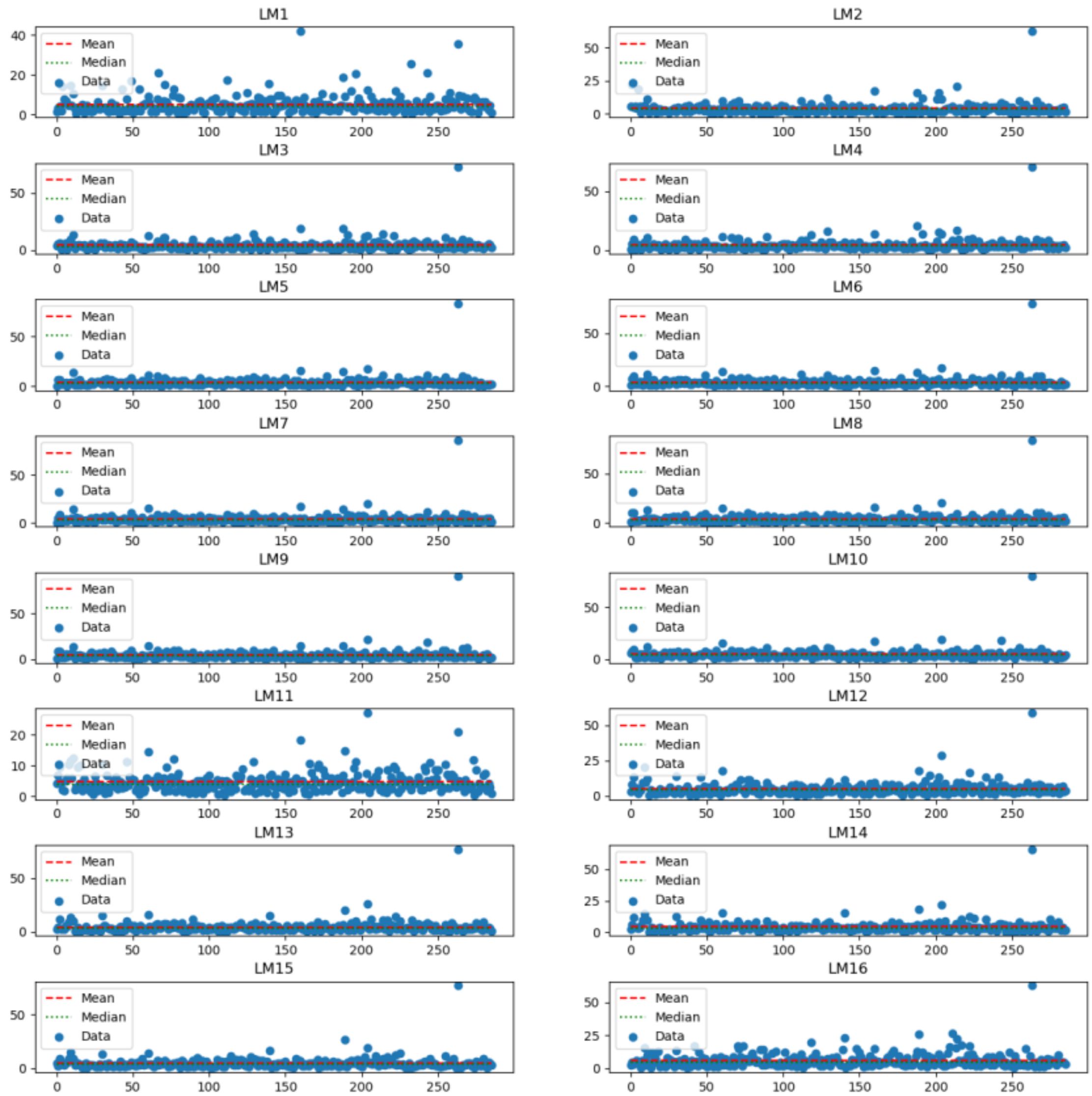


Figure B.17: The distribution of distances on each landmark on left mandible images

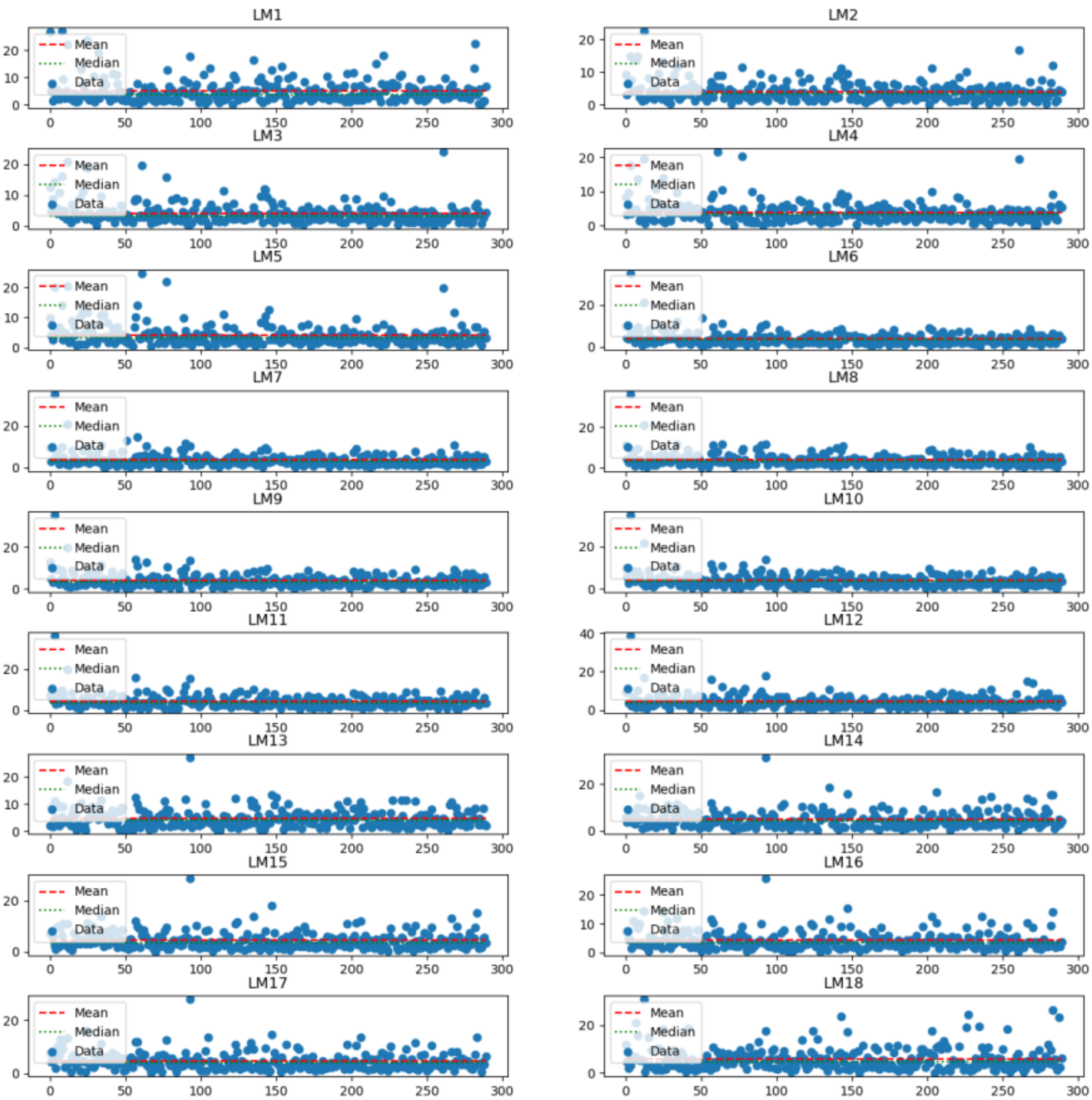


Figure B.18: The distribution of distances on each landmark on righ mandible images