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 Liberation, 33405 Talence, France*
[b] *UMR 1349 IGEPP, BP 35327, 35653 Le Rheu, France*
[c] *Dalat University, Dalat, Lam Dong, Vietnam*

Abstract

In the previous century, deep learning has been introduced for the artificial intelligence program, but it is difficult to apply it in real cases because of the limitation of the resource at that time. In recent years, it has risen strongly because of improvements in 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, each elementary block is a sequence 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 3 parts of 293 beetles. During the experiments, the network has been tested in two ways: training from scratch and applying fine-tuning process. In the fine-tuning step, to lead the pre-training step, a large public dataset of key points on human faces has been

*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.

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 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, key points detection, morphometry analysis

---

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 may contain different number of nodes, called *neurons* which have been inspired from the biological neural system [2], and this is the main unit responsible for calculating at each layer. According, the neurons at a layer extract the representation of input data which comes from the previous layers, then it computes a new output to the next layer. 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 [1, 8, 9, 10]. All of them have exhibited impressive performance comparing to more classical methods, for example, CNN is a specific one for pre-processing data which have grid topology, such as 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 example, to detect human face [14], human pose [17] or to measure out the 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 the images of 3 parts of 293 beetles: pronotum, head, and elytra. 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 facial key points database and fine-tuned our parameters model. The biologists have asserted that 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 algorithms have been introduced as multiple layers of non-linear features by using statistical methods to select the best features to forward to the next layer. Additional, the back-propagation [18], which has been introduced in the early $1960s$ in a inefficient and incomplete form, did not use to train the network at that time, instead, they used layer by layer least squares fitting. Until 1989, LeCun [5] used convolutional networks in combination with backpropagation to classify handwritten digits (MNIST), and it was known as a first practical system 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, 19], speech recognition [3, 4], language translation [8, 9], natural language processing [1, 10, 20]. 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 such 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. [21].

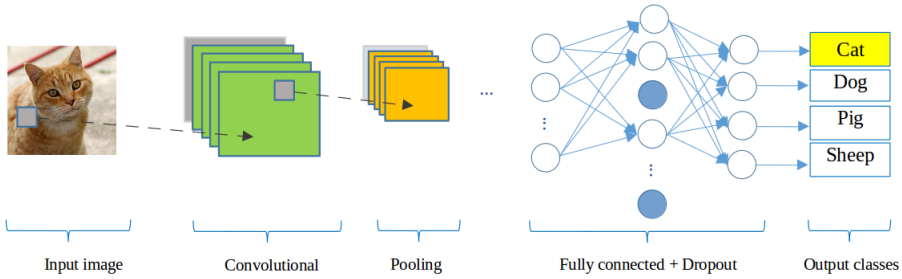


Figure 1: A CNN network for classification problem

2.2. State of the art 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 [22], GoogLeNet [7], VGGNet [23], or ResNet-50 [24]. The main difference between these networks is that their architectures became deeper and 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 in 2D 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

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

are concatenated from the coarse stream and provides accurate localization. Cintas et al. [16] have introduced an architecture which enabled to recognize 45 landmarks on human ears. Their model includes 3 times repeated of a structure consists of 2 convolutional layers, 1 pooling layer, and 1 dropout layer, to extract the features. These structures are 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 describing it, we will present the augmentation procedure that we have defined for our dataset.

3. Data augmentation

From AlexNet to ResNet-50, the obtained success stories [6, 24] 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 hundred 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 to improve 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

$\in [1, 255]$ 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.

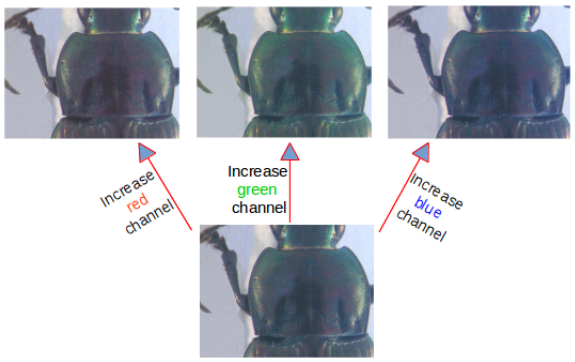


Figure 2: A constant $c = 10$ has been added to each channel of an original image

In the second procedure, each channel is considered separately and one grayscale 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).

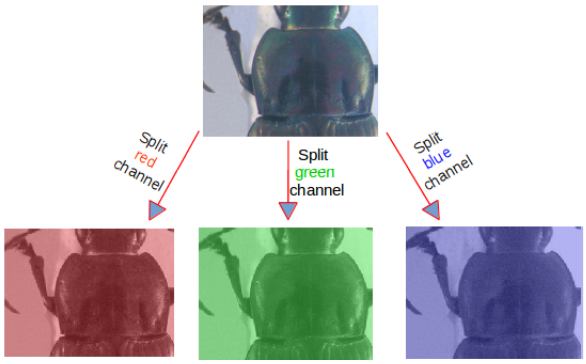


Figure 3: Three channels (red, green, blue) are separated from original image

4. Network architectures designing

As we have presented previously, several CNN architectures are available from literature and tool 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, e.g. convolutional, maximum pooling, dropout, and fully-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 CONV layers is usually done. The POOL layers progressively reduce the spatial size of the representation, reduce the number of parameters, computation in the network, and also 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 fully-connected (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).

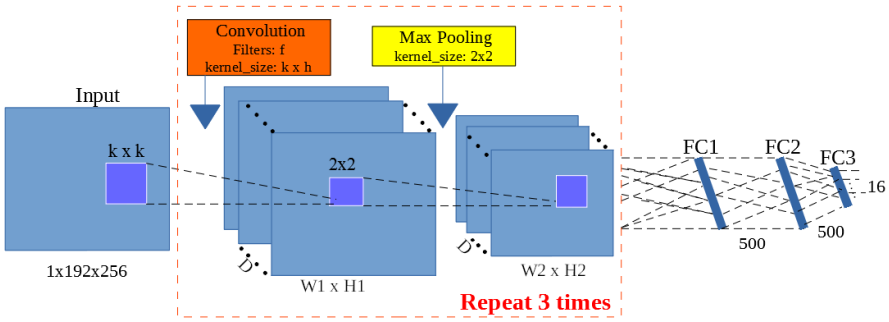


Figure 4: The architecture of the first model

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 over-fitting by adding a step to remove some nodes. This has significantly reduced overfitting and over performed the other regularization methods [25].

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

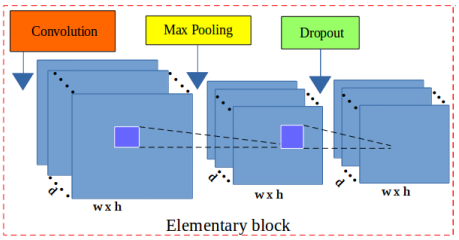


Figure 5: The layers in an elementary block. It includes a CONV layer (orange), a maximum POOL layer (yellow) and a DROP layer (green).

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.

The core of CNN is training over iterations. In each iteration, the features of images are computed in two phases: forward and backward. In the forward phase, the features are computed following the order of the layer in the network. In the backward phase, the values of learnable parameters are computed and

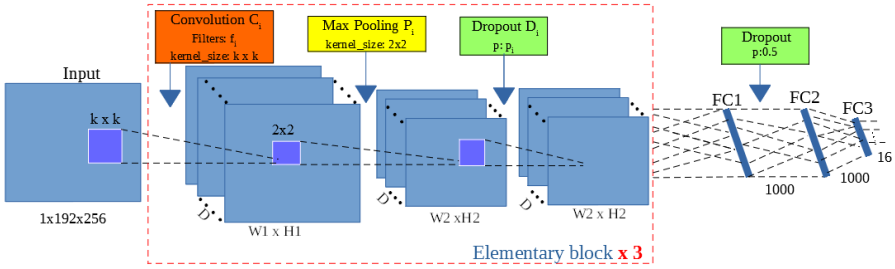


Figure 6: The architecture of EB-Net

updated to increase the accuracy of the network by using an optimizer. There are many ways to optimize the learning algorithm, but gradient descent [26] 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 was designed to use a learning rate initialize at 0.03 and to stop at 0.00001, while the momentum rateS was updated from 0.9 to 0.9999. The values were 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 [27] 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

Fig. 7 presents the 5 different beetle parts belonging to our dataset. The two first ones (from the left side) have been studied in previous work based on image processing techniques to work with segmentable images [28]. The choice to turn to deep learning methods for the three remained ones have been

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

motivated by the high difficulty to segment them, as we can observed in Fig. 7. Segmentation is most often a requirement to apply traditonal image processing methods and can be a bootleneck to achieve treatments. Using convolutional networks does not require this operation and provide solution to overcome this problem. Pronotum was the first part we analyzed with deep learning and the presented results mainly concern these images.

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

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 each pronotum image, a set of 8 manual landmarks is available. They have been set by biologists and are considered as the ground truth for the evaluation. In deep learning, many kinds of loss functions can be considered depending on the class of problem solving by the network, we have considered Root Mean Square Error (RMSE) because it is usually used for regression problems where the outputs are not discrete values as in the case of landmarks coordinates.

In order to predict landmarks for 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 and the 260 remaining images have been used to train and to validate the model. So, 9 rounds will be necessary to predict all landmarks. Of course, this dataset has been augmented as described in Section 3 to provide 1820 images for these 2 steps.

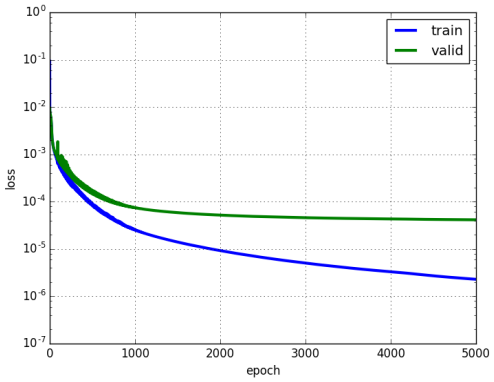


Figure 8: The losses (training and validation) of the 1^{st} model. The blue curve presents the RMSE errors of training process while green curve is the validation errors.

As it has been mentioned in Section 4, the first and the second model exhibit over-fitting behavior. Fig. 8 shows the different curves of the losses during training and validation step in the first architecture model. The blue curve presents the RMSE error of training process while green curve is the validation error. Clearly, over-fitting has appeared in the first model, e.g. training loss is able to decrease but validation loss is stable. In the second one, no concrete change appears in the curves even the parameters of fully-connected layers were modified.

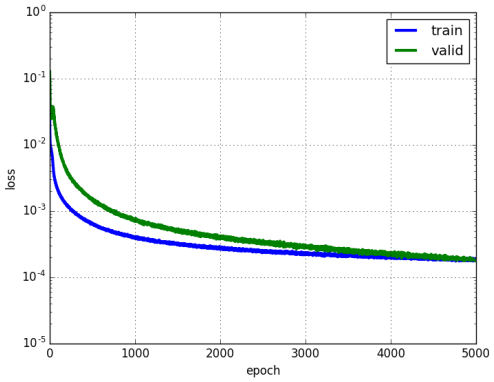


Figure 9: The losses (training and validation) of EB-Net

Fig. 9 illustrates the losses during the training of the third model, EB-Net. One can note that after several epochs, the two-loss values become close and the over-fitting disappears, we can assume that the addition of Dropout sequence inside elementary block works well to prevent over-fitting and improves the accuracy of the model greatly.

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

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

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, variance, minimum and maximum of correlation scores are given in the next rows. One can note that the correlation is strongly positive in each case with a very small variance, proving that each individual coordinate is well predicted.

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

	X-dimension	Y-dimension
Mean	0.8116	0.9438
Variance	0.0020	0.0006
Min	0.7474	0.9063
Max	0.8577	0.9638

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

dataset. With the images resolution 256×192, we can consider that an error around 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.

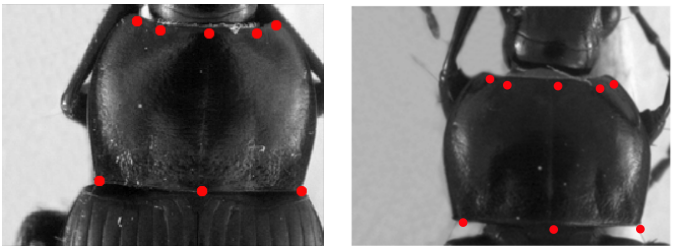
Landmark	**Distance** (in pixels)
1	**4.002**
2	4.4831
3	4.2959
4	4.3865
5	4.2925
6	**5.3631**
7	4.636
8	4.9363

Table 3: The average distances on all images per landmark on pronotum images.

To illustrate this point, Fig. 10 shows the predicted landmarks on two test images(chosen randomly). One can note that even some predicted landmarks (Fig. 10a) are close to the manual ones, in some case (Fig. 10b), the predicted are far from the expected results. So, the next step has been dedicated to the improvement of these results.

6. Improving results by fine-tuning

The results that we have been just discussed, have been obtained by training EB-Net from scratch but training a network from scratch is not the only



(a) Image with well-predicted landmarks

(b) Image with inaccuracy landmarks

Figure 10: The predicted landmarks, in red, on the images in test set.

way to work in Deep learning. It is possible to initialize parameter values by extracted values from another experiment with another dataset. This technique is called transfer learning [29]. In transfer learning, the obtained parameters values of a model, which have been used to solve a problem, are reused for other datasets [30] 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 [31], a well-known dataset with more than $100,000$ images, has been used to train many famous CNN architectures such as AlexNet [6] or VGG-16 [23] 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 as it has been described in by Lin et al. [32], ImageNet features mainly concern the detection of global shape of the objects whereas landmarks can be considered as local features. Luckily, searching for landmarks is well defined in face recognition and facial key points detection, and we can consider that this application presents similarities with our problem. So, we have decided to

train EB-Net with a facial key points dataset and then to transfer the trained parameters to fine-tune on beetle's images.

6.1. Pre-train EB-Net on Facial Keypoints dataset

A **Facial Keypoints dataset** has been published for a competition in the 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. 11 shows four face images in the dataset and the landmarks on each face.

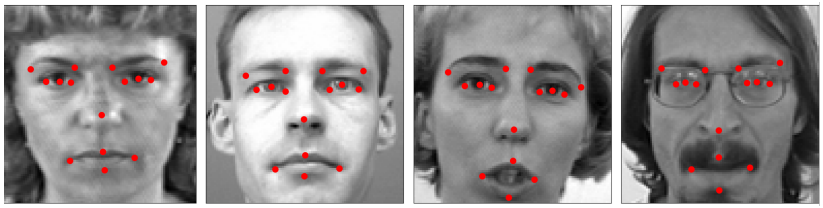


Figure 11: Four face images in the dataset and ground truth position of the landmarks.

For the pre-training step, EB-Net was trained with this dataset, the first objective of this task was to evaluate and to compare the effectiveness of EB-Net with other published results in the Kaggle challenge. Basically, the layer's parameters are the same than for tranining from scratch, we have just adapted the image size in EB-Net to match to 96×96 (Kaggle images size) and the output number of the last FC layer to correspond to the waited number of landmarks (15 landmarks). Considering the EB-Net hyper-parameters, the learning rate and momentum remained the same but the number of epochs has been increased to $10,000$ to improve the parameters learning. After training, the obtained RMSE score is 1.1464. This score is better than top 3 on the leader board of this competition.

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

6.2. Fine-tuning on beetle parts

The fine-tuning stage was processed by transfering the layer parameters of the pre-training step and by continuing the backpropagation. One can note that, the images in Facial Keypoints dataset are squared, in order to respect this we have reduced the size of beetle's images to 192×192 by cropping a background band. To declare the difference between the Kaggle's images and the beetle's ones, the stride property of the first convolutional layer has been modified from 1 to 2.

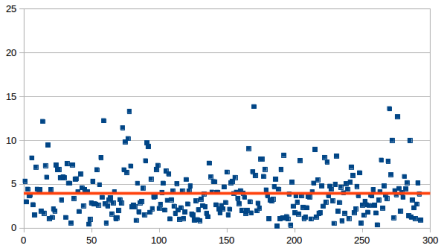
After finishing the fine-tuning process, EB-Net was used to predict the landmarks on test images. To evaluate the accuracy of the model's outputs, the distances (in pixels) between predicted and corresponding manual landmarks have been calculated again as their average distances. Tables 4 resumes the results on pronotum images: **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. The green and red values are respectively the best and the worst average distances in the two cases. The same procedure has been applied to the two others parts: elytra and head, the obtained results can be seen in Appendix A. First of all, the whole results have been clearly improved with the help of transfer learning for each landmark, but to go deeply in the analysis, it is worth to note that average computing can hide different situations. So, the distribution of the distances has been taken into account.

Fig. 12 describes the distribution of distances of two examples cases: 1^{st} and 6^{th} landmarks. Clearly, with the help of fine-tuning, the distances have decreased and more convergence under the mean value can be observed.

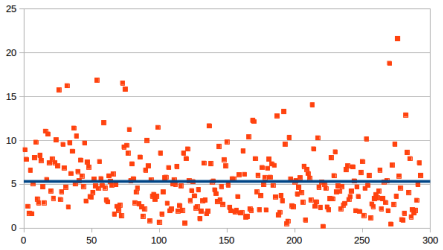
Another way to characterize the distribution of the values can be given by standard error, median, minimum and maximum values. These statistical values

#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

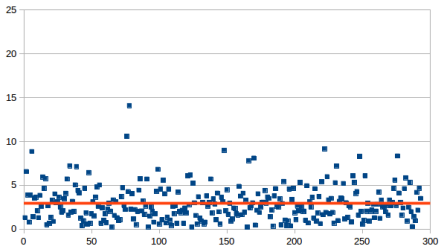
Table 4: Average distances comparison between training from scratch and fine-tuning on pronotum images



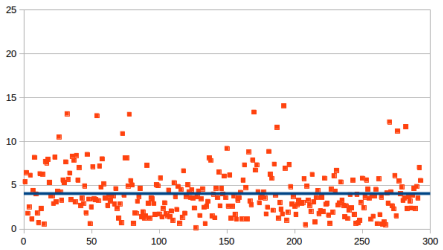
(a) 1^{st} landmark (from scratch)



(b) 6^{th} landmark (from scratch)



(c) 1^{st} landmark (fine-tuning)



(d) 6^{th} landmark (fine-tuning)

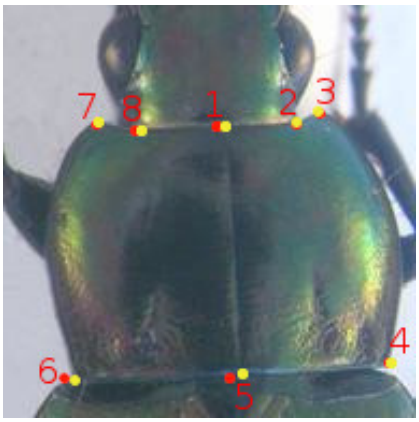
Figure 12: A comparison of distances distribution of the 1^{st} landmark and the worst case (6^{th} landmark) when applying two processes. The top/bottom line shows the distribution when applying training from scratch/fine-tuning. The lines in charts present for the average values.

are presented in Appendix B. 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 distances set into two parts, are smaller than the averages values and they are very close to the minimum values and so far from maximum values. It confirms that almost all distances stay around the median values and the predicted landmarks are good enough to replace the manual ones. The distribution of the distances on each landmark of each part are given in Appendix C.

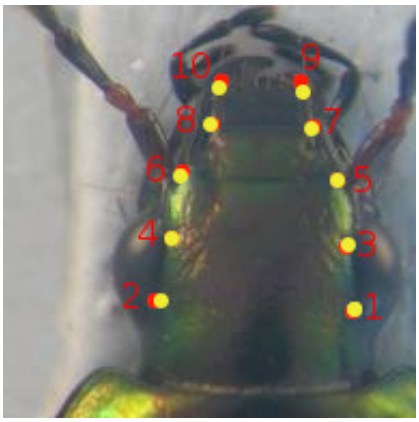
From these results, we can observe that most of distances are close to the mean and median values, only some exceptional cases are really far. To illustrate that the Fig. 13 shows both the predicted (in red) and the manual (in yellow) landmarks for the three beetle parts.

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: $\approx 25.98\%$ on pronotum, $\approx 15.8\%$ on elytra, and $\approx 18.10\%$ on head part. Additionally, 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, and **91.78%** on elytra.

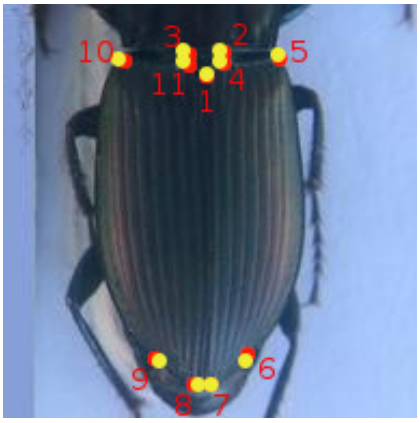
We have also a comparison between the results of deep learning and early methods where we have applied image processing techniques to predict the landmarks [28]. Clearly, the result with fine-tuning has improved the location of estimated landmarks. Even the average distances obtained from scratch training 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.



(a) Pronotum



(b) Head



(c) Elytra

Figure 13: The location of predicted landmarks in one case of each part. The red/yellow points represent the predicted/manual landmarks.

7. Conclusion

In this work, we have presented a new CNN model, EB-Net, to predict key points, landmarks, on 2D anatomical images of beetles. EB-Net model includes the repetition of 3 Elementary Blocks, followed by 3 fully connected layers. Each elementary block is a sequence of 1 CONV layer, 1 maximum POOL layer and 1 Dropout layer. In order to augment the dataset size, we have generated new images by modification of color channels of the original images. In the first

strategy of the training step, EB-Net has been used to train and to test on the images of each part of beetle. While in the second strategy, transfer learning has been applied to improve the results. EB-Net has been trained on a human facial key points dataset before transfering to fine tune and to test on beetle's images.

To evaluate the predicted landmarks, the distances between them and corresponding manual ones have been computed. Then, the statistical indicators have been considered, such as: average distance, median distance, standard error, minimum and maximum distance. These values have figured out that using the convolutional network to predict the landmarks on biological images leads to satifying results without human intervention and the procedures to pre-process (or post-process) the digital images.

In both case of training from scratch and fine-tuning, most of predicted landmarks are close to the manual 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, e.g. about all beetle parts. The quality of predicted coordinates 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] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural computation 1 (4) (1989) 541–551.

[19] 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.

[20] 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.

[21] 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.

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

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

[24] 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.

[25] 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.

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

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

[28] 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

[29] L. Torrey, J. Shavlik, Transfer learning, Handbook of Research on Machine

Learning Applications and Trends: Algorithms, Methods, and Techniques 1 (2009) 242.

[30] 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

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

[32] 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. Comparing the results between training from scratch and fine-tuning process

Table A.5 and A.6 show the comparison of the average distance between two processes: training from scratch and fine-tuning, on each landmark of head and elytra images, respectively. The green and red numbers represent the best and the worst distances in each case.

#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 A.5: The average distance of two processes: training from scratch and fine-tuning, on each landmark of 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 A.6: The average distance of two processes: training from scratch and fine-tuning, on each landmark of elytra images

Appendix B. Statistic information on each beetle's part

Table B.7, B.8, and B.9 display the statistical values on each part. The green and red numbers represent the best and the worst values on each statistical indicator, 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 B.7: The statistical indicator 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 B.8: The statistical indicator 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 B.9: The statistical indicator values on elytra images

Appendix C. The distribution of distances on each part

The Figure C.14, C.15, and C.16 illustrate the distribution of distances on each landmark of each part: pronotum, head, and elytra, respectively. The red line represents the mean value of the distance in each case.

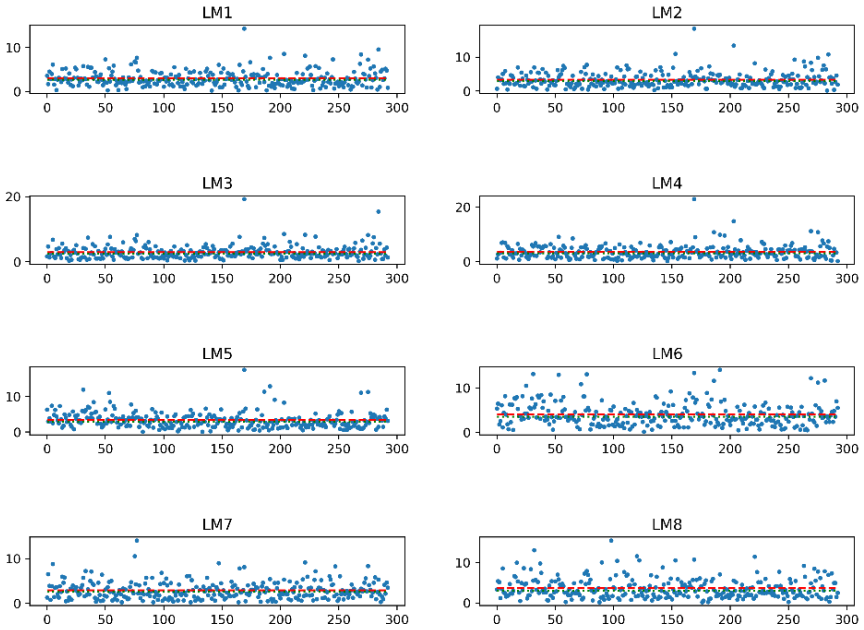


Figure C.14: The distribution of distances on each landmark of all pronotum images

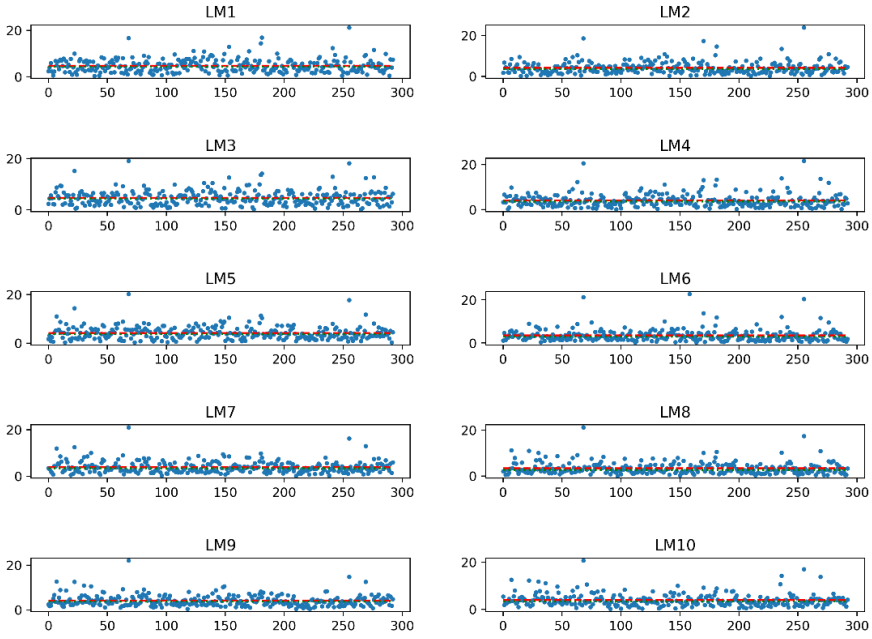


Figure C.15: The distribution of distances on each landmark of all head images

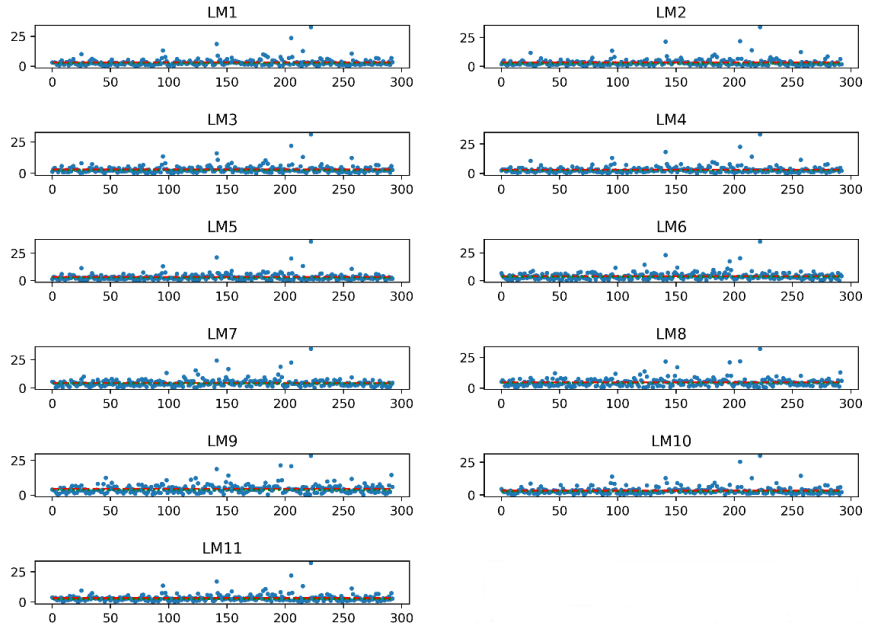


Figure C.16: The distribution of distances on each landmark of all elytra images