Automated Morphometrics using Deep Neural Networks : Case Study on a Beneficial Insect Species

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

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

Abstract

Aims: ... *Methods:* ... *Conclusions:* ... Landmark is one of the important concepts in morphometry analysis. Setting landmarks is not only used to measure the shape of the object but is also applied to analyze the inter-organisms variations. Currently, the landmarks are mostly determined manually by the biologists. In this work, we propose a method to automatically predict the landmarks on biological images: Deep Learning, more specifically Convolutional Neural Network (CNN). We proposed a CNN architecture which was built from the module, so-called "elementary blocks". Each block was made up of some popular layers of CNN. This work has been also introduced a procedure to augment number of images in our dataset which can see a little bit small in our case to apply deep learning methods. The network then trained and tested on a dataset includes three parts of beetle (pronotum, head and elytra). In the experiments, we applied two strategies to evaluate the network and to improve the obtained results: training from scratch and applying a fine-tuning step. The predicted landmarks from the network have been compared with the manual landmarks which provided by the biologists. The obtained results have proved that the predicted landmarks were considered to be statistically good enough to replace the manual landmarks.

Keywords: Landmarks, morphometry, deep learning, CNN

1. Introduction

In the context of ecosystem services, there is an interest in studying complex interactions between evolution of insect populations and environmental factors affecting their functions. In order to assess specifically pest-regulating services and in line with studies pointing to shape traducing function [1], there are more and more research about beneficial insect morphometrics [2, 3]. In such morphometric studies, it is common to analyze subject's shape independently of their poses and sizes [4]. Since the late 20^{th} century [5], rooted in a strong statistical background, geometric morphometrics addresses the study of such biological shapes [6]. It is an effective set of methods with several specialised softwares readily available [7, 8]. Classical geometric morphometrics uses a set of landmarks to describe shape, a landmark being a two-dimensional anatomically-relevant point. In order to investigate the possibility of automated morphometric geometrics on beneficial insects, we chose to focus on one of the most common and ubiquitous beneficial insect of north-western France, *Poecilus cupreus* (Carabidae). It is considered

*Corresponding author
Email address: van-linh.le@labri.fr (Le Van Linh)
[1] both authors contributed equally to this work.

a polyphagous predator [9] beneficial to agriculture, being able to consume a large variety of agricultural pests including weed seeds, slugs and aphids [10]. As a Coleoptera, its morphological variability is usually measured on exoskeleton structures such as the head, pronotum and elytra [11].

Of course, the first step in any morphometric geometrics study is the digital imaging of the biological specimens with controlled illumination and contrasting background. As such, morphometric landmark detection and positioning can be though as a particular problem of features detection and solved using robust digital image processing [12]. In the recent years, the term "deep learning" emerged describing class of computational models composed of multiple processing layers learning representations of data with multiple levels of abstraction [13]. Each layer extracts the representation of the input data from the previous layer and computes a new representation for the next layer. In the hierarchy of model, the lower layers take care of the primary features whereas the higher layers care for the abstract features to enlarge the aspect of input for the computational task (classification, regression, ...) and to suppress irrelevant variations. Deep learning algorithms have proved to be very efficient in a wide variety of domains, notably computer vision [14, 15, 16, 17, 18], speech recognition [19, 20, 21], question answering [22] and language translation [23, 24]. Within deep learning, Convolutional Neural Networks (CNNs) are well known for their success in many computer vision tasks such as image classification [14, 15] and objets recognition [17, 18]. Recent success of this algorithm in human biometry [25] lead us to believe in its potential for insect morphometrics.

1.1. Related works

Landmark-based geometric morphometrics has been applied to a variety of research questions and applications in biology. The applications can be ranged from fossil human/dinosaurs [26, 27] to butterfly/fly wings [28], zebrafish skeletogenesis [29], flower shapes [30]. They have been also concerned on medical imaging, e.g., cephalometry aims at analyzing the human cranium for orthodontic diagnosis and treatment planning [31, 32].

Geometric morphometry analysis based on landmarks is mainly beginning by positioning the landmarks in two-dimensional images, which is typically achieved manually. Landmarks are then compared by employing various statistical methods to distinguish landmark variations or the changing of shape in large populations, e.g., Procrustes analysis. Depending on applications, the number of landmarks varies, it could be ranged from several to dozens of landmarks, for example, 15 landmarks have been defined in a study on drosophila wing [33] or 25 landmarks have been used in a research on zebrafish [29, 34]. Manual setting landmarks is time-consuming and difficult to reproduce. A solution that can automatically provide landmarks could be useful in these studies.

Recently, automatic prediction of landmarks has appeared in many applications of various domains: In computer vision, landmark localization is usually studied on human faces where we identify some points corresponding to significant parts on face, e.g., nose, eyes, mouth region [35, 36]. In biology, the landmark identification problem has been appeared in the studies to analyse shape and size on the organisms [33, 37], e.g., analyzing the corolla shape variation. In biomedical field, the problem of automatic landmark positioning has been addressed in cephalometry [32, 38]. The familiar methods in these domains are based on the combination of template matching and prior knowledge information after a step of the segmentation of interesting objects. Lowe et al. [39] have proposed SIFT method to find the corresponding keypoints between two images. Palaniswamy et al. [33] have proposed a method based on probabilistic Hough Transform to automatically identify the landmarks in digital images of Drosophila

wings. In previous work [40], we have proposed a method which was extended from Palaniswamy's method, to determine landmarks on beetle images. The experiments have been done on two sets of mandibles images which have an ordinary shape and are easy to segment. The obtained results were satisfying when comparing to the landmark's coordinates of the manual setting. Unfortunately, this method can not be applied to other parts of beetles as pronotum, head and elytra. As these pictures have been done before dissection, shape segmentation has been a trap for our method.

More recently, several studies have succeed to solve this task by using machine learning algorithms followed by global structure refinement [41, 42, 34]. In recent years, deep learning has been widely used in computer vision. Using convolutional neural network to determine the landmarks on 2D images has achieved good results. As was common, the CNN inputs raw pixels of the image, then it analyzes the relations between the pixels to predict the coordinates of landmarks. These operations are performed by a sequence of layers. It is worth to note we do not recognize the appearance of the segmentation step in the process. Thus, CNN has offered an effective solution to face images that have difficulty in segmentation. In the landmarking context, Yi Sun et al. [43] have proposed cascaded convolutional neural networks (three-levels) to predict the facial points of interest on the human face. Each level considers the face from global to local regions to determine the landmarks. Zhanpeng Zhang et al. [44] proposed a *Tasks-Constrained Deep Convolutional Network* to optimize facial landmarks detection. The model determines the facial landmarks with a set of related tasks such as head pose estimation, gender classification, age estimation, face recognition, or facial attribute inference. Cintas et al. [25] has introduced a network to predict the landmarks on human ears. After training, the network has the ability to predict 45 landmarks on human ears. Based on our knowledge, CNNs have been widely used in biological applications but not to provide the landmarks. In this work, we proposed a CNN architecture to predict the landmarks on biological images, specific beetle's images.

1.2. Contributions

In this article, we detailed a CNN architecture that we have designed to automatically set landmarks on beetles images, so-called Elementary Block Network (EB-Net). The prediction has been evaluated by comparing to the ground truth manually provided by biologists. We describe how we have applied data augmentation to remedy the problem of using machine learning algorithms on the small dataset. We will also describe how performance can be improved by using transfer learning from another dataset like human facial points.

2. Material and Methods

In this section, we first present the dataset that we have used in this study, as well as the strategies to pre-process the data. Then, we describe the designed network architecture to predict the landmarks in the beetle's images.

2.1. Dataset and preprocessing

In order to provide the experimental data, we have selected the Brittany lands (North-West of France) to collect the samples. After collecting in three months, a collection of 293 beetles has been established (147 males and 146 females/ 155 organic and 138 conventional) (Figure 1). As usual, images of beetles have been chosen to be studied instead of using real objects for practical reasons. For each beetle, five images corresponding to five parts of beetles

have been taken into account: elytra, pronotum, head, left and right mandibles. The pictures of each body parts were captured under a trinocular magnifier at ≈ 300 pixels/mm for elytra, ≈ 600 pixels/mm for pronotum and head, 1500 pixels/mm for mandibles. One can note that the head, pronotum, and elytra parts have been captured before dissection. The left and right mandibles have been separated from the beetle's body before taking the photos. All the images have been taken with the same camera under same conditions to release in the RGB color mode with a size of 3264×2448 pixels.

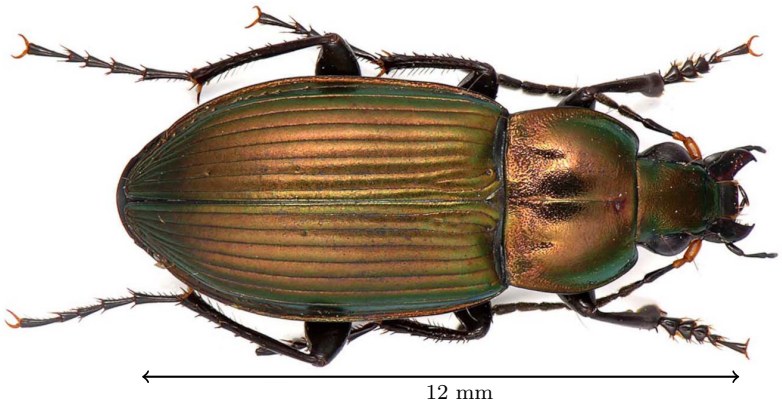
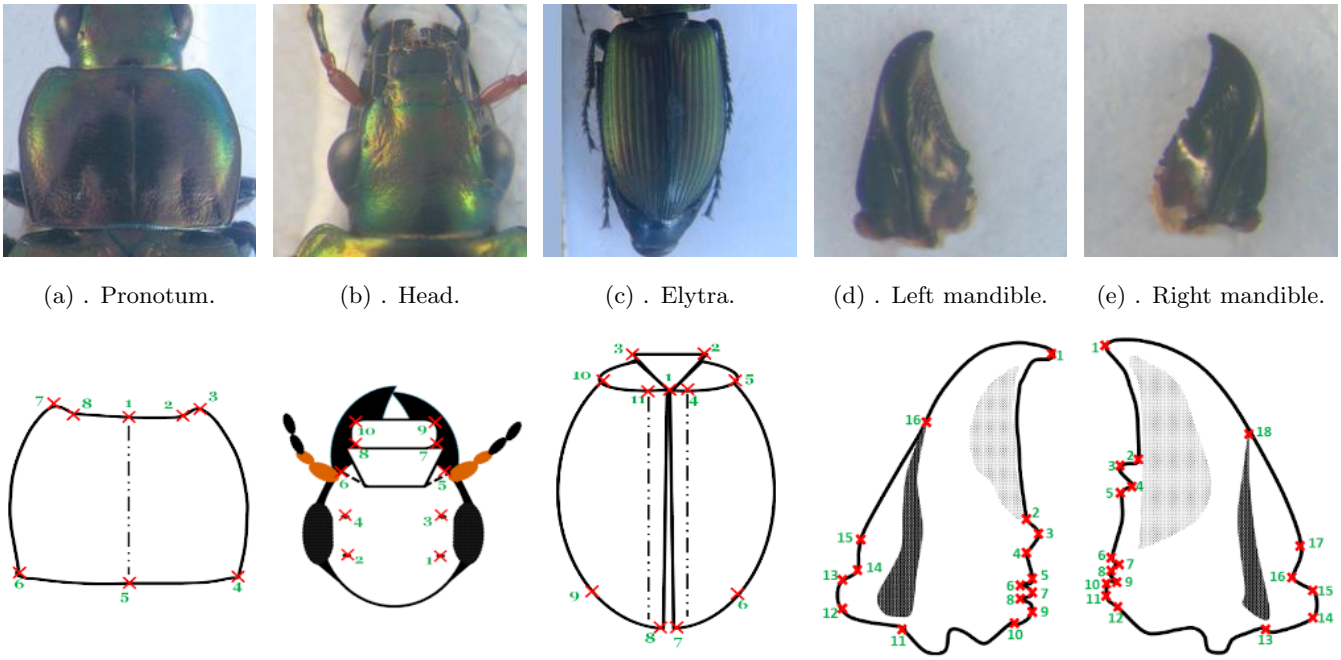


Figure 1: An illustration of the beetle.

In the next step, morphological landmarks were first set manually on the dorsal views of each body part of the beetles (head, pronotum, elytra, right and left mandibles). The morphology of each body part was processed and analyzed separately in order to limit variation resulting from their relative positions due to articulation. Landmarks were chosen according to the ease and the precision of their location on each specimen (Figure 2). Replicability analyses were performed to confirm the accuracy of landmarks positioning. They were positioned on each picture with TPSDig2 software (version 2.17) (Rohlf, 2013a). In some individuals, mandibles could not be processed because they were lacking or broken. For each specific part, a set of number of landmarks has been provided, for example, 8 *landmarks for pronotum,* 10 *landmarks for head,* 11 *landmarks for elytra,* 16 *and* 18 *landmarks for left and right mandibles, respectively* (Figure 2). In the context of this study, these manual landmarks have been used as ground truth to evaluate the output of our method.

The success stories [14, 15, 16] have proved that CNN models have to be trained on a large dataset with an enormous number of data samples before using the trained model to perform on testing data. Training the model with a big dataset can help the model able to learn more different cases and to improve the learning ability of the network. Unfortunately, providing a large dataset is too costly in several domains, e.g., in biology, medicine. A solution to deal with this problem is to create the misshapen data from real data and to add them to the dataset. In our case, we have only 293 images for each part of the beetles. This number is large from the point of view of manual operations, but it is not enough to apply deep learning methods. So, we have applied data augmentation process to face this problem.

Most often in deep learning applications, dataset augmentation uses operations such as translation, rotation, or scaling, which are well-known efficient to generate new versions of existing images [14, 45]. In order to select the

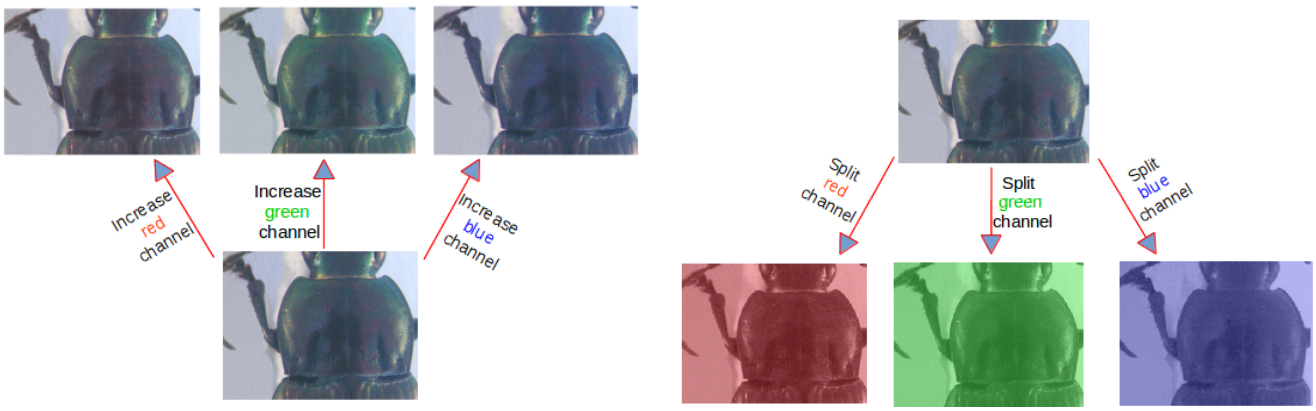


(a) . Pronotum. (b) . Head. (c) . Elytra. (d) . Left mandible. (e) . Right mandible.

(f) . Pronotum. (g) . Head. (h) . Elytra. (i) . Left mandible. (j) . Right mandible.

Figure 2: The sample images in our dataset (top) and manual landmarks on each part defined by biologists (bottom).

right method for our application, we have done some tests by moving the object in the picture. In each time, we have quickly gone to the over-fitting in the training step. Consequently, we have preferred other ways to produce misshapen images by operating on the image's color channels. We have proposed two strategies to augment the number of images in our dataset.



(a) . Add a constant to each channel. (b) . Split the channels of image.

Figure 3: The two strategies to augment the number of images in pronotum set.

The first strategy was applied to change the value of each channel in the original image. According to this, a constant have been added to a channel of RGB image for each time. For example, if we add a constant $c = 10$ to the red channel from an original RGB image, we will obtain a new image with the values at red channel by greater than the red channel of original image a value of 10. By this way, we can generate three new RGB images from a

RGB image.

The second procedure was to split the channels of RGB images in order to create three gray-scale images. This work seems promising because the network model works on single-channel images. At the end, we have generated six versions from an image. In total, we have obtained $293 \times 7 = 2051$ images for each set of images. Figure 3 illustrates the two described strategies.

To perform the objective, we have observed the input size of the several CNN models [14, 15, 25, 43] and noticed that most often their input sizes were limited to 256 pixels. One can note that our images were released with the size of 3264×2448, as mentioned in Section 2.1. This size is a bit heavy for training the network. Consequently, we have down-sampled our images to a new size of 256×192 to respect the ratio between width and height. Of course, coordinates of landmarks have been down-sampled to the new size of images. Practically, convergence is usually faster if the average of each input variable over the training set is close to zero [46]. So, the brightness of the image is normalized to $[0, 1]$.

2.2. Network architecture

Our initial trials were inspired by AlexNet architecture [14]. These models have been designed by combining in sequence the classical layers, e.g., convolutional (CONV), max pooling (POOL) and fully-connected (FC) layers. Unfortunately, over-fitting effects appearance very quickly. In deep learning context, it exists another type of layer, Dropout (DROP) layer, which is well-known to prevent over-fitting. After experiments, we have defined a concept of Elementary Block (EB). This EB concept is the core of our model architecture.

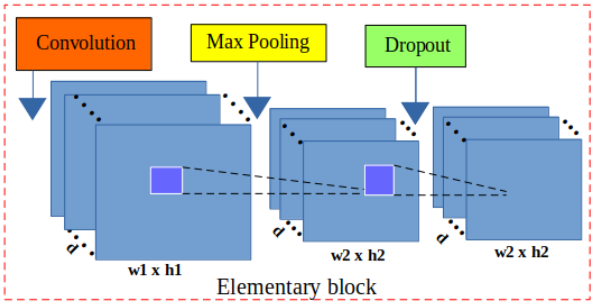


Figure 4: The components of an elementary block

Figure 4 illustrates the order of the classes in an EB. It is defined as a sequence of a CONV layer, a maximum POOL layer, and a Dropout layer. In an Elementary Block, the convolution layer is used to extract the high-level features by applying different filters on the input. Setting POOL layer after the CONV layer towards to reduce the spatial size of the representation to decrease the number of parameters as well as to use less the computing time. The DROP layer is usually inserted between the FC layers to prevent over-fitting [13, 14], but we have included them in the extracting feature blocks to produce some kinds of image noise augmentation. The dropout layer randomly drops some connections during the training process. It makes the network thinner than the original one (fewer parameters), and training a network with dropout layer is equivalent to train a set of thinner networks. As presented in [47], DROP layer has significantly reduced over-fitting and given more considerable improvements than other regularization methods.

We have assembled three Elementary Blocks to create whole network architecture, called Elementary Blocks Network (EB-Net). The used parameters of layers in EBs have been set as follows: the depths of CONV layers are set to increase $(32, 64, 128)$ with a small kernel $(2\times2,\ 2\times2)$ from the first to the third block, respectively. The POOL layers in all three blocks have been designed with a filter of 2×2 and a stride of 2 pixels. With these parameter values, the spatial size of the image will be halved after every block. The probabilities of the dropout layers are set increasing: $0.1, 0.2$, and 0.3, respectively. In order to extract the global relationship between the features and to provide the prediction, three fully-connected layers have been added after the combination of EBs. The first FC layer takes all features from the last block as the input for computing. The last FC layer outputs the coordinates prediction of landmarks. The number of outputs at each FC layers is set to $1000, 1000$, and X, respectively. The number X equals to two times the number of landmarks that we want to predict $(x, y$ coordinates). For example, to predict 8 landmarks on pronotum, we have set 16 as the number of output of the last FC layer. Additionally, a dropout layer with the probability equals to 0.5 has been inserted between the first and the second fully-connected layers as mentioned in [48]. Figure 5 resumes the EB-Net architecture.

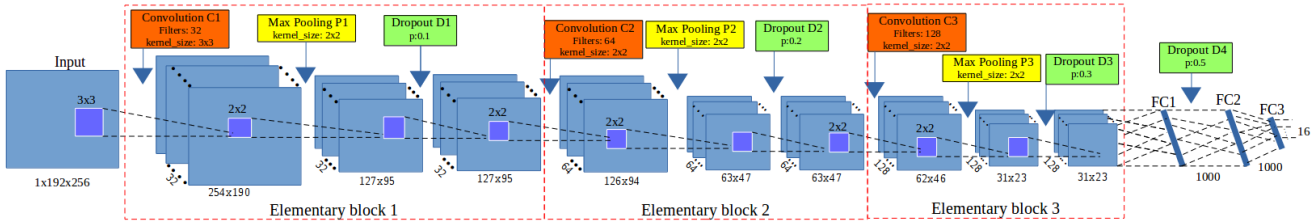


Figure 5: Elementary Blocks Network (EB-Net) architecture

In a CNN model, besides the model-specific parameters which involve the structure of the network (e.g., the number of layers or parameter values of each layer), the optimized hyper-parameters are also important variables in the designing of a CNN model. These variables are related to the training of the network model, for example, the loss function, the number of epochs, the batch size, or the initialized value of learning rate. Practically, these values are discovered through empirical observation depending on the task and the dataset. In our application, we have chosen the most common optimization algorithm, Stochastic Gradient Descent (SGD), like usual studies [14, 25]. In order to apply SGD, the two relevant parameters need to be provided: learning rate and momentum. In our application, the learning rate begins from 0.03 and stops at 0.00001; whereas the momentum rate is updated from 0.9 to 0.9999. During the training, learning rate and momentum are adjusted to fit with the number of epochs. For example, the learning rate and momentum at the first/the third epoch are 0.03/0.029994 and 0.9/0.90001, respectively. Besides, the Root Mean Square Error (RMSE) has been chosen as loss function because it is employed for regression problems where outputs represent quantitative values as the case of coordinates of landmarks. The EB-Net has been implemented by using Lasagne framework [49], and trained in 5000 epochs on Linux system by using a NVIDIA GPU (Titan X) card.

2.3. Setting and training EB-Net

In order to provide the predicted landmarks for all images, we have applied cross-validation technique to select the test images, we will call a selection step is a round. For each round, we take 33 images and keep them for

testing, the 260 remaining images will be used to train and to validate the network model. It is worth to note that the set of 260 images has been augmented by the strategies described in Section 2.1, to provide 1820 (260×7) images for training and validationsteps. To achieve the cross-validation steps, we have to do 9 rounds in total.

During the training and validation step, 1820 images are randomly divided into training and validation sets with a ratio of 60% : 40%. In each training step, the pair of *image* and *its manual landmarks* is inputted to train the network model. In the testing step, we input the image only to the trained model to predict landmarks. In our cases, the manual landmarks have been given by the biologists. So, they can be used as ground truth to train the network, as well as to evaluate the predicted ones.

3. Results

3.1. The first evaluation

EB-Net has been firstly trained and tested on the pronotum images. Table 1 shows the losses of 9 rounds when we train EB-Net on pronotum images. We can observe the losses are tiny in each round, and the differences among rounds are not various even we have altered images in each round. Based on the success of EB-Net on pronotum images, we have employed it on the two sets of head and elytra images. We have got the similar situation as training on pronotum. The obtained losses are tiny, and not too much difference between the losses of rounds.

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

Table 1: The losses during the training of EB-Net on pronotum images

Figure 6 shows the losses of training and validation processes of one round on pronotum images. The blue curve is training loss, and the green curve is validation loss. Learning is effective and over-fitting does not appear. We can assume that EBs have properly worked to prevent over-fitting during training and validation steps.

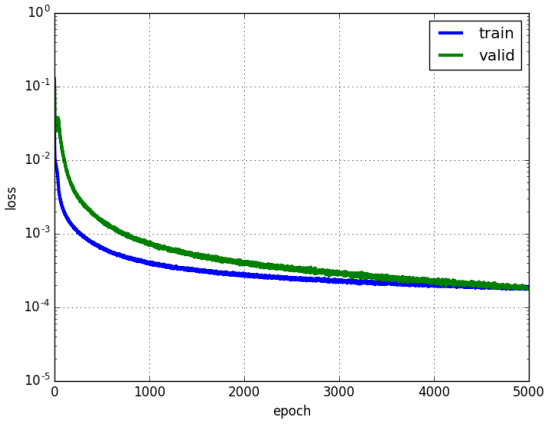


Figure 6: The losses during training EB-Net on pronotum images.

The trained model of each round has been then used to predict the landmarks on the corresponding testing images. The coordinates of outputted landmarks are evaluated by comparing with the manual ones. We have calculated the distances (in pixels) between the manual landmarks and corresponding predicted ones. Then, the average distance on each position has been considered. Table 2 shows the average distance on each position of all three parts: pronotum, head and elytra. With the image's size of 256×192, we can consider that an error around 1% (≈ 2 pixels) coud be an acceptable error. Unfortunately, our results exhibit the average distance around 4 pixels for the best cases, and more than 5 pixels in the worst cases.

Landmark	1	2	3	4	5	6	7	8	9	10	11
Pronotum	4.00	4.48	4.3	4.39	4.29	5.36	4.64	4.94	-	-	-
Head	5.53	5.16	5.38	5.03	4.84	4.45	4.79	4.53	5.14	5.06	-
Elytra	3.87	3.97	3.92	3.87	4.02	4.84	5.21	5.47	5.27	4.07	3.99

Table 2: The average distances per landmark on images of each set.

It is worth to note that an average value could reflect two different cases: the values closed together (small dispersion) or two sets of values very far (large dispersion). In order to see we are in which situation, Figure 7 shows the distribution of distances between the manual and predicted landmarks for the best and the worst cases in each set of images. Each point presents the distance between the points (landmarks) of an image. The lines (blue/red) illustrate the average distance in each case. In both of two cases (the best and the worst), the distances are most often stay in the region from 0 to the average value. However, it exhibits a small dispersion, some points are still far away the mean value.

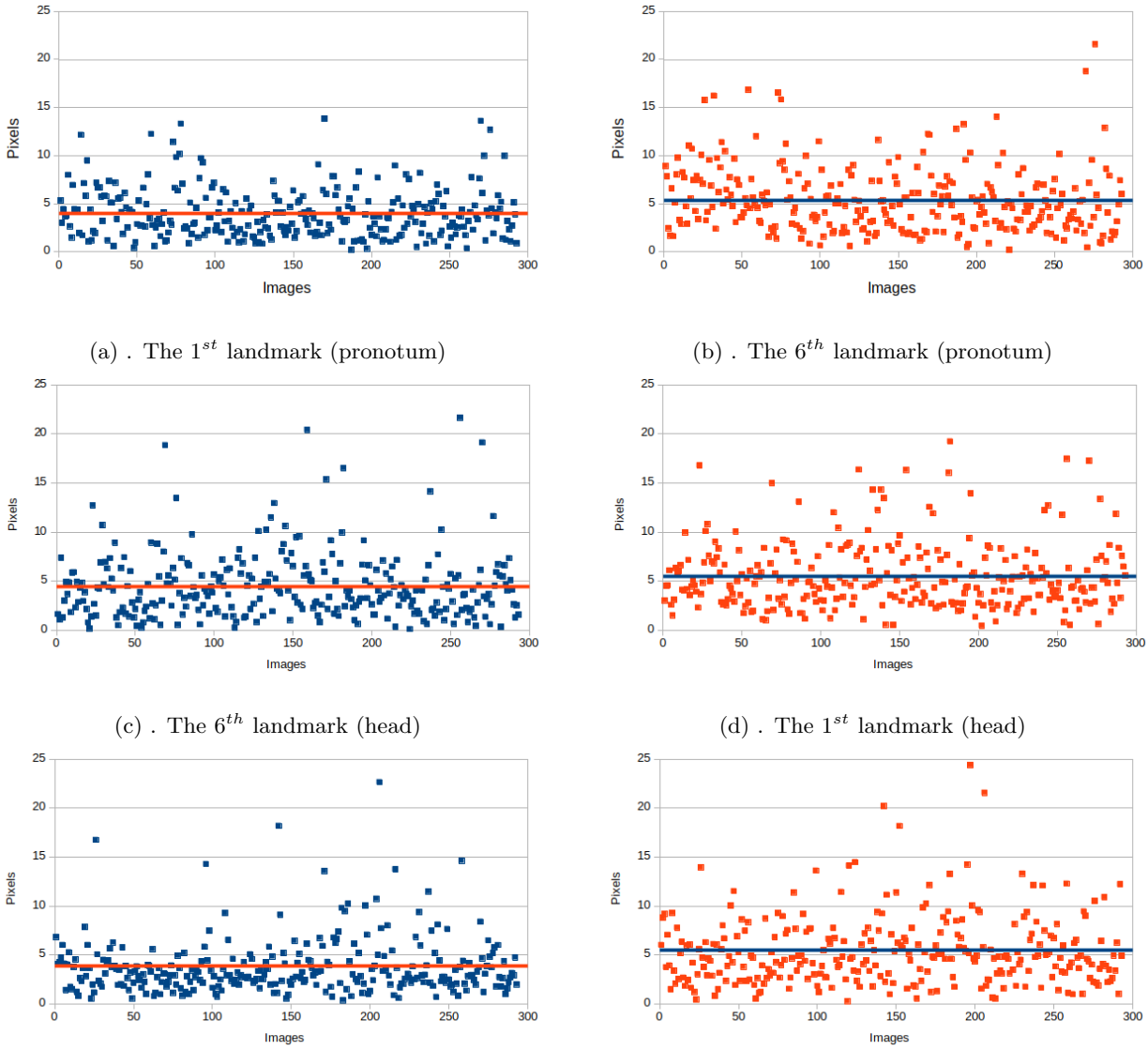
To illustrate the points, Figure 8 shows the landmarks on the three parts of beetles. The red/yellow points present the predicted/manual landmarks. One can note that even some predicted landmarks are close to the manual ones, we have also some predicted coordinates that are far from the expected results.

The EB-Net has achieved good outcomes in most cases, but it exists of several complex images in our dataset that the model meets a difficulty to recognize. It explains why some high distance values have appeared in Figure 7. The next step is to improve the predicted results.

3.2. Transfer learning process

Working with deep learning requires not only to design a good architecture but also to provide a large dataset to train and to test the model. Practically, this is a potential problem in some application domains as in biology. In section 2.1, we have augmented the number of images in our dataset and used it to train EB-Net. However, our number is far away several thousands images. In this case, knowledge transfer or transfer learning between tasks could be another solution to improve the prediction.

Transfer learning [50, 51] is a technique in deep learning to re-purpose a model, which has been designed for a specific task (called source task), on another related task (target task). Choosing which strategy of transfer learning to apply depending on the relationship between two tasks, as well as the size of database. Practically, transfer learning is mostly targeted on 2 strategies:



(a) . The 1^{st} landmark (pronotum)

(b) . The 6^{th} landmark (pronotum)

(c) . The 6^{th} landmark (head)

(d) . The 1^{st} landmark (head)

(e) . The 1^{st} landmark (elytra)

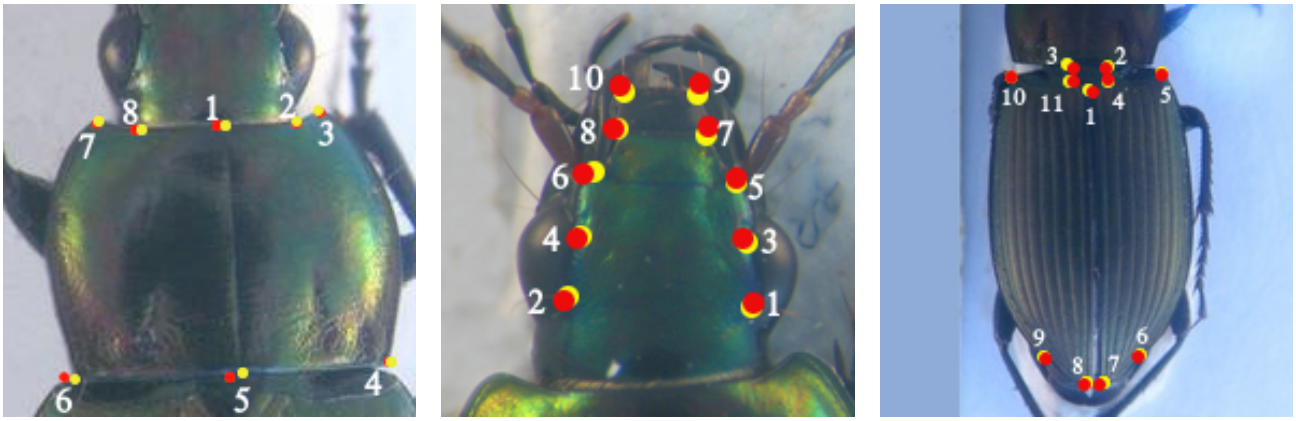
(f) . The 8^{th} landmark (elytra)

Figure 7: The distribution of distances for the best and the worst cases of the three parts

- **Use CNN as a fixed feature extractor**: Take a CNN pre-trained on a large dataset, then remove the last fully-connected and use the rest layers of CNN as a fixed extractor for the new dataset.

- **Fine-tuning a CNN**: This scenario is the same as the first strategy. However, it does not only replace and retrain the last layer but also fine-tunes the weights of the pre-trained model by extending the backpropagation. One can note that to reuse a pre-trained model, the parameters have been adapted between two tasks. These parameters could be the size of input images, the number of outputs, or the parameters of each layer. As usual, the parameter values at each layer, e.g., padding or stride values, are selected to change their values to declare the differences between the two tasks.

As a preliminary work, we have tested several well-known models [14, 52] that have been trained on ImageNet [53]. These pre-trained models have been shared in deep learning community as a source to re-use the features



(a) . Pronotum (b) . Head (c) . Elytra

Figure 8: The landmarks on three parts of beetle. The red/ yellow points present the predicted/ manual landmarks.

of ImageNet dataset. Unfortunately, the features from ImageNet seem that do not relevant for our application because the image features mainly concern the detection of global shape of the objects whereas landmarks can be considered as local features [54]. Fortunately, searching for landmarks is explicitly defined in other applications like face recognition, or facial key-points detection. Consequently, we have decided to continue with EB-Net: we have pre-trained EB-Net on a public facial keypoints dataset, then transferred the pre-trained parameters to fine-tune on beetle's images.

3.2.1. Pre-train EB-Net on facial keypoint dataset

In recent years, several competitions [2] have been organized for predicting facial keypoints on human face. In this context, dataset for training model has been freely published. As we have mentioned, our problem has a revelant to this kind of applications. So, we have decided to choose a facial keypoints dataset to train EB-Net, and then to transfer the parameters to fine-tune it for beetle's images.

The dataset that we have selected has been published for a challenge[3] in the Kaggle website. It includes 2140 images of human faces (96×96) pixels. For each image, 15 landmarks have been defined: 6 landmarks for eyes, 4 landmarks for eyebrows, 4 landmarks for the mouth, and 1 landmark for nose tip. Figure 9 shows 4 examples of face images and their landmark positions in the dataset.

In order to use EB-Net on this dataset, we have adapted the parameters of the input and the output layers to match with the face images size and the landmark number. The new parameter values are 96×96 for the input size, and 30 for the number of outputs of the last FC layer (corresponding to 15 landmarks). In hyper-parameters side, we have increased the number of epochs to 10000 but kept the same for other values as training from scratch. As the first step, we take into account the RMSE score to evaluate and to compare the effectiveness of EB-Net with other published scores in the challenge.

Table 3 shows the scores of top 3 on the challenge board and our one. It is worth to note that their scores

[2]Deepfake Detection Challenge/ Facial Keypoints Detection
[3]https://www.kaggle.com/c/facial-keypoints-detection

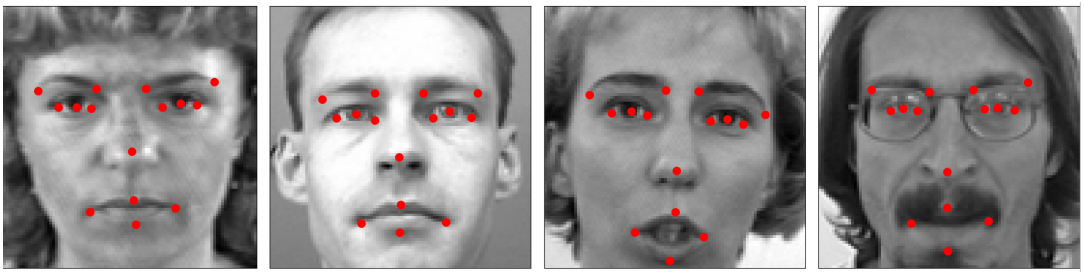


Figure 9: The samples (face and manual landmarks) in the dataset that we used to pre-train EB-Net.

Team	Olegra 1^{st}	Trump 2^{nd}	Enes 3^{rd}	EB-Net
RMSE score (in pixels)	1.2824	1.4004	1.4026	**1.497**

Table 3: RMSE comparison between our score and top three of challenge.

have been obtained by testing their models on a private set of images that we can not access. So, we have kept 100 images in the public dataset for testing process. Comparing with these scores, the three models present better results than us but we are very close. In our opinion, the RMSE score around the 1 pixels is not so far if we would like to display the landmarks on the images. Consequently, we have the base to believe that EB-Net is still good in any way, and we have decided to re-use the pre-trained parameter values to fine-tune the model for beetle images.

3.2.2. Fine-tuning on beetle images

As we have mentioned, fine-tuning is a strategy of transfer learning that could boost the efficiency of a model on the target task. Technically, the weights of a CNN model can be fine-tuned by continuing the backpropagation, and it exists two ways to perform fine-tuning process: *frozen* and *unfrozen*.

- **Frozen** scenario: the parameter of lower layers (close to the input layer) will be fixed, we fine-tune only the higher ones (close to the output layer).
- **Unfrozen** scenario: allows continuing to update the parameter values of all layers in the pre-trained model.

In order to fine-tune EB-Net on beetle images, we have gone with unfrozen process to continue updating the parameter values. One can note that the sizes of images in the two datasets are different: the beetle images have a size of 256×192 pixels; whereas the size of facial images is 96×96 pixels. Therefore, adjustments are needed to match the two tasks.

First of all, reducing the resolution of the beetle images to 96×96 could be lead to a loss of essential information. As our images contain a background band that is easy to suppress with a pre-processing operation, we have chosen to remove the background region instead of down-sampling our pictures. Moreover, removing the background pixels can limit the effect of un-useful image areas. So, the new beetle images are finally set to 192×192 pixels. The EB-Net parameters will be settled to take into account these values between the pre-training and fine-tuning steps. To declare the modification, we mention in a stride value of the first convolutional layer equals to 2 (as the usual way to do [51]).

3.2.3. Fine-tuning results

To evaluate fine-tuning process, the parameters of the pre-trained model have been transferred to separately fine-tune on three sets of images: pronotum, head, and elytra. We present all the obtained results in the same way as the previous section to provide an explicit comparison.

Foremost, the average distance at each landmark is provided by computing in the same way as the previous one: we calculate the distances (in pixels) between predicted and corresponding manual landmarks, then these distances are used to calculate the average value for each landmark position.

Tables 4, 5, 6 show the comparison at each position between the average distances provided by the two processes (training from scratch and fine-tuning) on pronotum, head, and elytra, respectively. The first row presents the landmark number; **From scratch** row reminds the previously average distances when EB-Net was trained from scratch; **Fine-tune** row presents the new average distances; the last row presents the improvement percentage between the two processes. The green and red values are respectively the best and the worst values in each process. All distances are given in pixel unity. From these tables, all the average distances have decreased from 1 to 1.5 pixels in both of three sets of images. Clearly, the fine-tuning process has improved the prediction of landmarks: we can see the change at the best-predicted position (green ones), but it exists a group of well-predicted landmarks in each set of images, such as:

- For pronotum: the 1^{st}, 3^{rd}, and 7^{th} landmark.
- For head: the 6^{th}, 7^{th}, 8^{th}, and 10^{th} landmark.
- For elytra: the $1^{st} - 5^{th}$, 10^{th}, and 11^{th} landmark.

At the opposite side, the worst cases remain the same positions as previously: the 6^{th}, 1^{st}, and 8^{th} landmark on pronotum, head, and elytra, respectively.

Landmark	1	2	3	4	5	6	7	8
From scratch	4.00	4.48	4.3	4.39	4.29	5.36	4.64	4.94
Fine-tune	2.99	3.41	2.98	3.54	3.37	4.06	2.93	3.64
% of impr.	25.25	24.01	30.56	19.18	21.55	24.28	36.85	26.16

Table 4: Average distances comparison on pronotum images

Landmark	1	2	3	4	5	6	7	8	9	10
From scratch	5.53	5.16	5.38	5.03	4.84	4.45	4.79	4.53	5.14	5.06
Fine-tune	4.82	4.21	4.73	4.11	4.18	3.5	3.92	3.4	4.17	3.94
% of impr.	12.83	18.43	12.15	18.42	13.69	21.43	18.29	24.94	18.88	22.01

Table 5: Average distances comparison on head images

Considering on each landmark position, the improvement is different depending on the difficulty of its location. But, all cases have been improved even they are the best or the worst cases. With the help of fine-tuning, the

Landmark	1	2	3	4	5	6	7	8	9	10	11
From scratch	3.87	3.97	3.92	3.87	4.02	4.84	5.21	5.47	5.27	4.07	3.99
Fine-tune	3.21	3.28	3.2	3.22	3.31	4.21	4.54	4.76	4.55	3.39	3.29
% of impr.	17.04	17.34	18.36	16.61	17.66	13.13	12.82	12.96	13.69	16.68	17.54

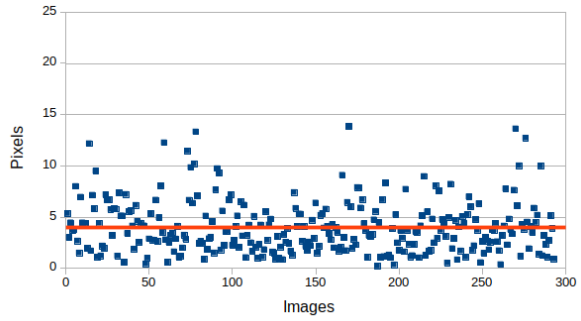
Table 6: Average distances comparison on elytra images

predictions have gained from 36.85%/24.94%/18.36% to 19.18%/12.15%/12.82% on pronotuom, head, and elytra, respectively.

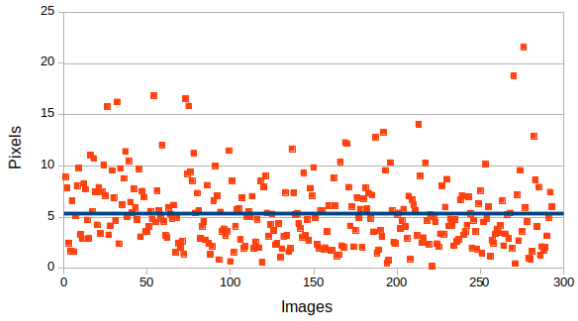
The average distance between manual and predicted landmarks could be not the only way to appreciate the obtained results. As we have mentioned, the mean value can hide different situations when deeply going to the analysis. Again, it could exist of two cases for an average value: all distance values are very close to the mean value, or they are widely widespread around the mean one. In this case, distribution of distances could be taken into account to characterize this situation.

Figure 10, 11, 12 show the distribution of distances on two samples in each set of images: pronotum, head, and elytra, respectively. In these charts, the x-axis and y-axis present the number of images and the distance (in pixel). Each point in the chart represents a distance between manual and predicted landmarks. The blue/red lines in charts present the average values. We can observe from the figures that the distance values are very different from the two processes (training from scratch and fine-tuning). The distances have been reduced with the help of fine-tuning process even they were the low or high values when training from scratch. We have gained more points (in the chart) in the range from zero to the average one, and the number of extraordinary values have been decrease too. For example, in the worst case of pronotum (the 6^{th} landmark), we do not see any point greater than 15 pixels with the fine-tuning process.

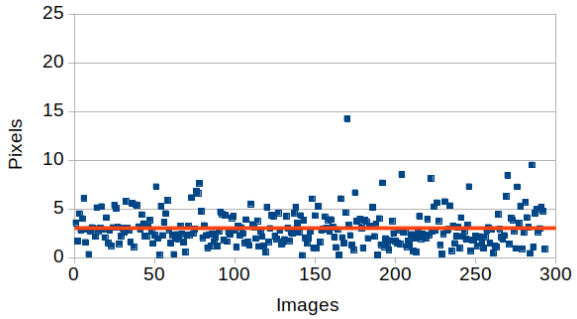
To illustrate the results, Figure 13 shows both predicted (in red) and manual (in yellow) landmarks on three random images from the three sets of images. Clearly, the predictions have been improved, they are more close to the manual ones. For example, we have obtained 7 well-predicted landmarks on the head images.



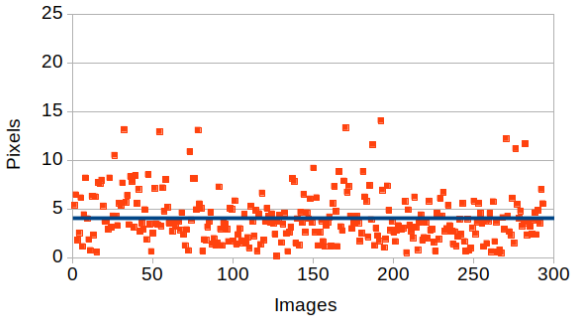
(a) . 1^{st} landmark (from scratch)



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

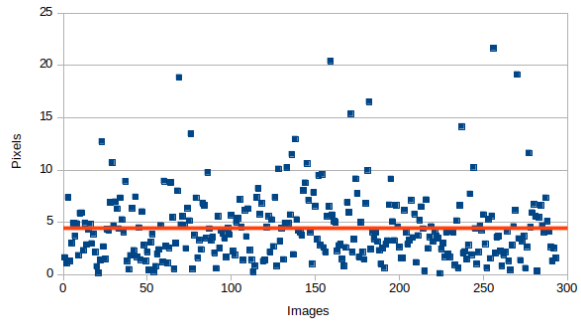


(c) . 1^{st} landmark (fine-tuning)

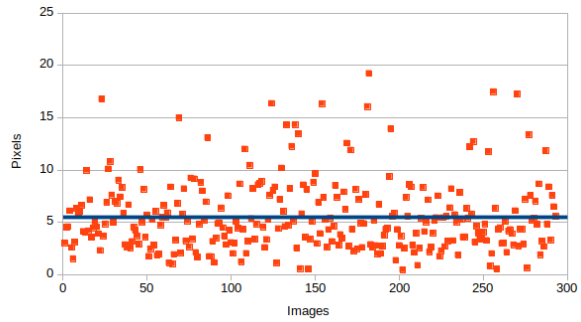


(d) . 6^{th} landmark (fine-tuning)

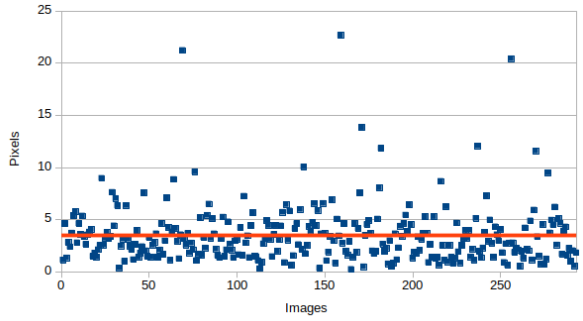
Figure 10: A comparison of distances distribution of the 1^{st} landmark and the worst case (6^{th} landmark) on pronotum images.



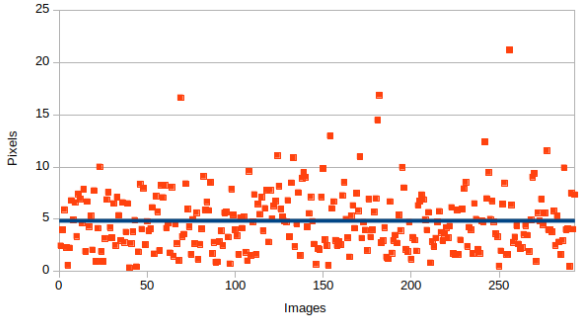
(a) . 6^{th} landmark (from scratch)



(b) . 1^{st} landmark (from scratch)

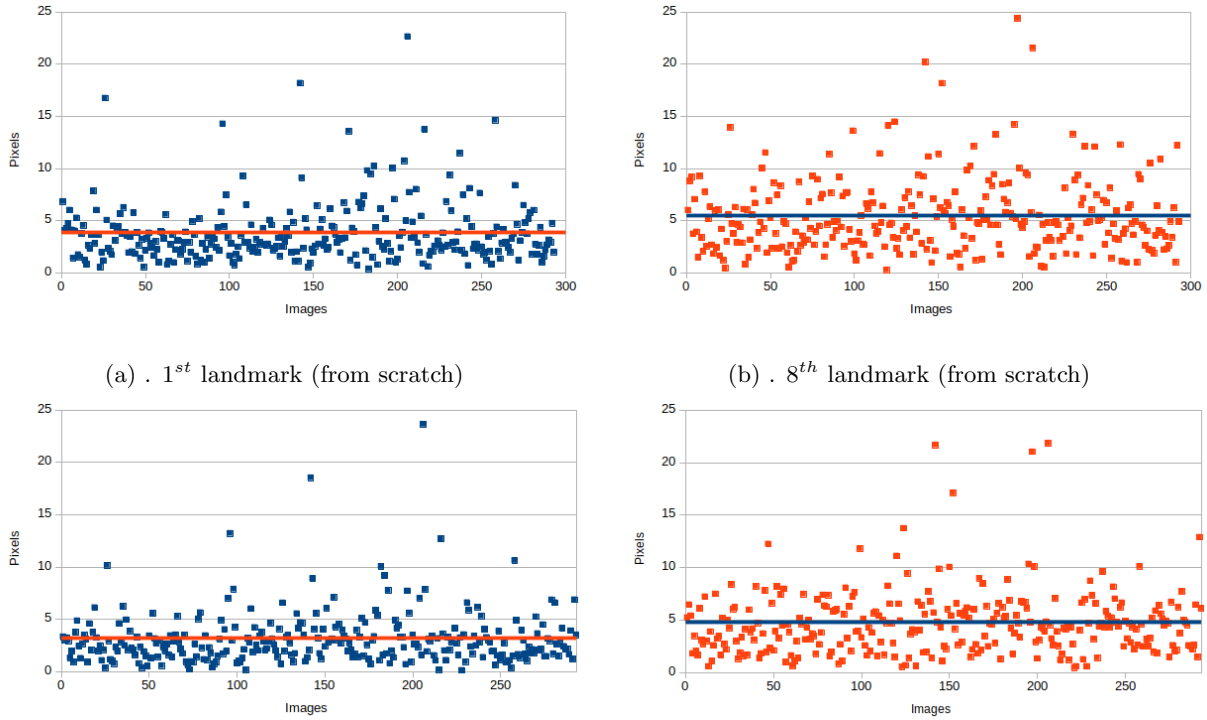


(c) . 6^{th} landmark (fine-tuning)



(d) . 1^{st} landmark (fine-tuning)

Figure 11: The distribution of distances of all head images on 1^{st} landmark and 6^{th} landmark.



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

(b) . 8^{th} landmark (from scratch)

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

(d) . 8^{th} landmark (fine-tuning)

Figure 12: The distribution of distances of all elytra images on 1^{st} landmark and 8^{th} landmark.

(a) . Pronotum

(b) . Head

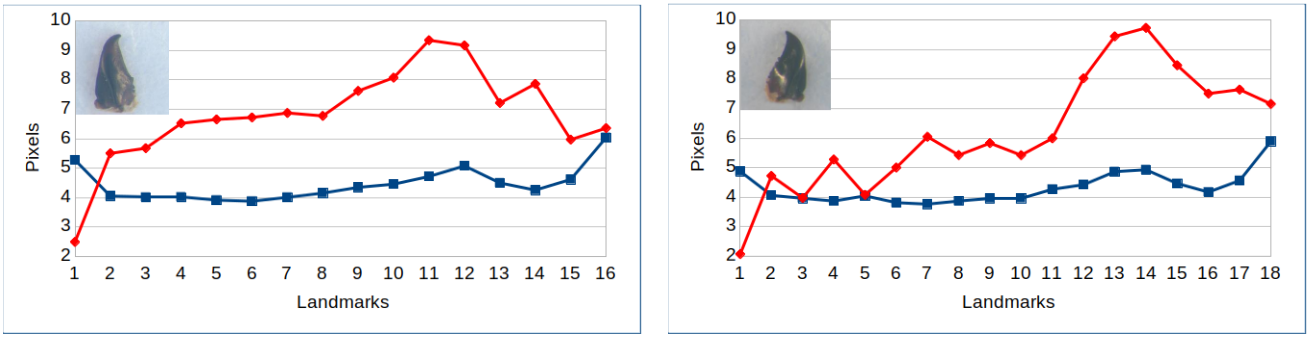
(c) . Elytra

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

3.2.4. Results on mandibles

In the five sets of images, the mandibles are considered as easy to extract by using the image processing algorithms. In [40], we have proposed a pipeline to estimate the landmarks in mandible images. In order to evaluate the effectivement of deep learning method and to compare with the obtained results, we have applied the fine-tuning process on mandible images in the same way as we have done on other parts. Figure 14 shows the comparison of average distance at each landmark position between the obtained results of two methods (deep

learning and image processing methods). The red curves illustrate the average distances which have been obtained from image processing technique while the blue curves present the results of fine-tuning process. To reach the comparison, the obtained values in [40] have been re-computed to match the new size of the images (192×192) by scaling these errors (distances) with the same ratio that we have used to down-sample images. Then, the average value has been computed for each landmarks position.



(a) . Left mandible

(b) . Right mandible

Figure 14: These charts show the average distance on each landmark of all mandibles images. The red, blue lines present the results from image processing and fine-tuning process, respectively.

For the left mandible (Figure 14a), it is worth to note that they have been more difficult to process than the right ones by using image processing techniques, as we have discussed in [40]. However, the results of the fine-tuning process are almost better than image processing one. The fine-tuning has improved the results even the cases that were difficult to predict with image processing (the $9^{th} - 12^{th}$ landmark).

For the right mandible (Figure 14b), with image processing techniques, some first landmarks (from 1^{st} to 6^{th}) have been well-predicted, which illustrated by small average values in the chart. However, these values begin to increase from the 7^{th} landmark. The reason is that the first group of events is mainly concentrated on the tip of the mandible where we can obtain the precise segmentation, while others are on the base where we can meet some difficulties to get the segment contours. For the fine-tuning process, the obtained values are more stable. Although some first values can be approximate or larger (from 2^{nd} to 7^{th} positions), other ones are better than the previous results (after 7^{th} position). Remarkably, the fine-tuning has a great improvement at the positions located at parts that are hard to extract the contours. More, these results are stable on every landmark position.

4. Discussion

In this work, we have proposed the elementary block as the based to build the CNN for predicting landmarks. It is worth to note that our block is a generic one. It is easy to add it, easy to test the model, even to remove a block from the model if the results are not satisfying. The users can choose suitable blocks for their applications. Choosing the number of blocks depends on the computing resources, as well as the expected results. For example, we have tried to add a new elementary block to EB-Net. The experiments have shown that the results have been improved, but we have spent more resources and computing times to reach this improvement. The average distances have been improved 0.5 pixels. However, this change is insignificant if we show it on the images. Consequently,

we need to note that it exists a balance between the depth of the model, the accuracy of outputs and the cost to compute.

TODO: on biological part.

5. Conclusion and future works

In this article, we have presented a convolutional neural network, EB-Net, for automatic detection landmarks on the pronotum, the head, and the elytra of beeltes after testing several models. It includes three times repeated structure which consists of a convolutional layer, a max pooling layer, and a dropout layer, followed by the connected layers. In the first step, the EB-Net has been separately trained on each part of beetles. In order to improve the results, the fine-tuning process has been applied by pre-training EB-Net on a facial keypoints dataset before transfering the parameter values to fine-tune on each set of images.

The results have been evaluated by calculating the distance between manual landmarks and predicted ones. These results on the three parts have shown that using CNN to predict the landmarks on biological images leads to satisfying results without need for segmentation step on studied object. The best set of predicted landmarks has been obtained after a step of fine-tuning step. These results have been delivered to biologists and they have confirmed that the quality of prediction allows using estimated landmarks to replace the manual ones.

In future, we plan to test our model on different datasets owned by the INRA team. All the implementations, EB-Net model and EB-Net parameters, are available freely on the Github website. It is possible to reuse EB-Net parameters for another landmark setting application and to apply transfer learning. We plan also to export EB-Net architecture to other application domains stuided in our team such as MRI images analysis, pose identification.

References

[1] C. P. Klingenberg, Evolution and development of shape: integrating quantitative approaches, Nature Reviews Genetics 11 (9) (2010) 623–635. doi:10.1038/nrg2829.
URL https://www.nature.com/articles/nrg2829

[2] K. Sasakawa, Utility of geometric morphometrics for inferring feeding habit from mouthpart morphology in insects: tests with larval Carabidae (Insecta: Coleoptera), Biological Journal of the Linnean Society 118 (2) (2016) 394–409. doi:10.1111/bij.12727.
URL https://academic.oup.com/biolinnean/article/118/2/394/2194832

[3] L. Raymond, A. Vialatte, M. Plantegenest, Combination of morphometric and isotopic tools for studying spring migration dynamics in Episyrphus balteatus, Ecosphere 5 (7) (2014) 1–16. doi:10.1890/ES14-00075.1.
URL http://onlinelibrary.wiley.com/doi/10.1890/ES14-00075.1/abstract

[4] D. G. Kendall, The diffusion of shape, Advances in Applied Probability 9 (3) (1977) 428–430. doi:10.1017/S0001867800028743.
URL https://www.cambridge.org/core/journals/advances-in-applied-probability/article/diffusion-of-shape/7CFF1175D4DCCF6063E403847120BE7B

[5] F. L. Bookstein, Foundations of Morphometrics, Annual Review of Ecology and Systematics 13 (1) (1982) 451–470. doi:10.1146/annurev.es.13.110182.002315.
URL http://www.annualreviews.org/doi/10.1146/annurev.es.13.110182.002315

[6] F. J. Rohlf, On Applications of Geometric Morphometrics to Studies of Ontogeny and Phylogeny, Systematic Biology 47 (1) (1998) 147–158. doi:10.1080/106351598261094.
URL https://academic.oup.com/sysbio/article-lookup/doi/10.1080/106351598261094

[7] D. C. Adams, E. Otrola-Castillo, geomorph: an r package for the collection and analysis of geometric morphometric shape data, Methods in Ecology and Evolution 4 (4) (2013) 393–399. doi:10.1111/2041-210X.12035.
URL http://onlinelibrary.wiley.com/doi/10.1111/2041-210X.12035/abstract

[8] C. P. Klingenberg, MorphoJ: an integrated software package for geometric morphometrics, Molecular Ecology Resources 11 (2) (2011) 353–357. doi:10.1111/j.1755-0998.2010.02924.x.

[9] A. Larochelle, The Food of Carabid Beetles:(coleoptera: Carabidae, Including Cicindelinae), Sillery: Association des entomologistes amateurs du Qubec, 1990.

[10] B. Kromp, Carabid beetles in sustainable agriculture: a review on pest control efficacy, cultivation impacts and enhancement, Agriculture, Ecosystems & Environment 74 (1) (1999) 187–228. doi:10.1016/S0167-8809(99)00037-7.
URL http://www.sciencedirect.com/science/article/pii/S0167880999000377

[11] T. Eldred, C. Meloro, C. Scholtz, D. Murphy, K. Fincken, M. Hayward, Does size matter for horny beetles? A geometric morphometric analysis of interspecific and intersexual size and shape variation in <Emphasis Type="Italic">Colophon haughtoni</Emphasis> Barnard, 1929, and <Emphasis Type="Italic">C. kawaii</Emphasis> Mizukami, 1997 (Coleoptera: Lucanidae), Organisms Diversity & Evolution 16 (4) (2016) 821–833. doi:10.1007/s13127-016-0289-z.
URL https://link.springer.com/article/10.1007/s13127-016-0289-z

[12] R. C. Gonzalez, R. E. Woods, Digital Image Processing (3rd Edition), Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

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

[14] 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.

[15] D. Ciregan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 3642–3649.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[17] 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.

[18] J. J. Tompson, A. Jain, Y. LeCun, C. Bregler, Joint training of a convolutional network and a graphical model for human pose estimation, in: Advances in neural information processing systems, 2014, pp. 1799–1807.

[19] T. Mikolov, A. Deoras, D. Povey, L. Burget, J. Černockỳ, 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.

[20] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, 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.

[21] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, B. Ramabhadran, Deep convolutional neural networks for lvcsr, in: 2013 IEEE international conference on acoustics, speech and signal processing, IEEE, 2013, pp. 8614–8618.

[22] A. Bordes, S. Chopra, J. Weston, Question answering with subgraph embeddings, arXiv preprint arXiv:1406.3676.

[23] 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.

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

[25] C. Cintas, M. Quinto-Sánchez, V. Acuña, C. Paschetta, S. de Azevedo, C. C. S. de Cerqueira, V. Ramallo, C. Gallo, G. Poletti, M. C. Bortolini, et al., Automatic ear detection and feature extraction using geometric morphometrics and convolutional neural networks, IET Biometrics 6 (3) (2016) 211–223.

[26] A. Rosas, L. Pérez-Criado, M. Bastir, A. Estalrrich, R. Huguet, A. García-Tabernero, J. F. Pastor, M. De la Rasilla, A geometric morphometrics comparative analysis of neandertal humeri (epiphyses-fused) from the el sidrón cave site (asturias, spain), Journal of human evolution 82 (2015) 51–66.

[27] J. L. Fearon, D. J. Varricchio, Morphometric analysis of the forelimb and pectoral girdle of the cretaceous ornithopod dinosaur oryctodromeus cubicularis and implications for digging, Journal of Vertebrate Paleontology 35 (4) (2015) e936555.

[28] N. Chazot, S. Panara, N. Zilbermann, P. Blandin, Y. Le Poul, R. Cornette, M. Elias, V. Debat, Morpho morphometrics: shared ancestry and selection drive the evolution of wing size and shape in morpho butterflies, Evolution 70 (1) (2016) 181–194.

[29] J. Aceto, R. Nourizadeh-Lillabadi, R. Marée, N. Dardenne, N. Jeanray, L. Wehenkel, P. Aleström, J. J. van Loon, M. Muller, Zebrafish bone and general physiology are differently affected by hormones or changes in gravity, PloS one 10 (6).

[30] T. van der Niet, C. P. Zollikofer, M. S. P. de León, S. D. Johnson, H. P. Linder, Three-dimensional geometric morphometrics for studying floral shape variation, Trends in plant science 15 (8) (2010) 423–426.

[31] C. Lindner, C.-W. Wang, C.-T. Huang, C.-H. Li, S.-W. Chang, T. F. Cootes, Fully automatic system for accurate localisation and analysis of cephalometric landmarks in lateral cephalograms, Scientific reports 6 (2016) 33581.

[32] V. Grau, M. Alcaniz, M. Juan, C. Monserrat, C. Knoll, Automatic localization of cephalometric landmarks, Journal of Biomedical Informatics 34 (3) (2001) 146–156.

[33] S. Palaniswamy, N. A. Thacker, C. P. Klingenberg, Automatic identification of landmarks in digital images, IET Computer Vision 4 (4) (2010) 247–260.

[34] R. Vandaele, J. Aceto, M. Muller, F. Peronnet, V. Debat, C.-W. Wang, C.-T. Huang, S. Jodogne, P. Martinive, P. Geurts, et al., Landmark detection in 2d bioimages for geometric morphometrics: a multi-resolution tree-based approach, Scientific reports 8 (1) (2018) 1–13.

[35] X. P. Burgos-Artizzu, P. Perona, P. Dollár, Robust face landmark estimation under occlusion, in: Proceedings of the IEEE international conference on computer vision, 2013, pp. 1513–1520.

[36] H. Yang, I. Patras, Sieving regression forest votes for facial feature detection in the wild, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1936–1943.

[37] Y. Savriama, A step-by-step guide for geometric morphometrics of floral symmetry, Frontiers in plant science 9 (2018) 1433.

[38] H. Mohseni, S. Kasaei, Automatic localization of cephalometric landmarks, in: 2007 IEEE International Symposium on Signal Processing and Information Technology, IEEE, 2007, pp. 396–401.

[39] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (2) (2004) 91–110.

[40] V. L. LE, M. BEURTON-AIMAR, A. KRÄHENBÜHL, N. PARISEY, MAELab: a framework to automatize landmark estimation, in: WSCG 2017, 25th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2017, Plzen, Czech Republic, 2017.
URL `https://hal.archives-ouvertes.fr/hal-01571440`

[41] B. Ibragimov, B. Likar, F. Pernus, et al., A game-theoretic framework for landmark-based image segmentation, IEEE Transactions on Medical Imaging 31 (9) (2012) 1761–1776.

[42] R. Donner, B. H. Menze, H. Bischof, G. Langs, Global localization of 3d anatomical structures by pre-filtered hough forests and discrete optimization, Medical image analysis 17 (8) (2013) 1304–1314.

[43] 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.

[44] Z. Zhang, P. Luo, C. C. Loy, X. Tang, Facial landmark detection by deep multi-task learning, in: European Conference on Computer Vision, Springer, 2014, pp. 94–108.

[45] C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, Journal of Big Data 6 (1) (2019) 60.

[46] Y. A. LeCun, L. Bottou, G. B. Orr, K.-R. Müller, Efficient backprop, in: Neural networks: Tricks of the trade, Springer, 2012, pp. 9–48.

[47] 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.

[48] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv:1207.0580.

[49] S. Dieleman, J. Schlter, C. Raffel, E. Olson, S. K. Snderby, D. Nouri, et al., Lasagne: First release. (Aug. 2015). doi:10.5281/zenodo.27878.
URL http://dx.doi.org/10.5281/zenodo.27878

[50] E. S. Olivas, Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques: Algorithms, Methods, and Techniques, IGI Global, 2009.

[51] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, in: Advances in neural information processing systems, 2014, pp. 3320–3328.

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

[53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[54] 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.