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

Le Van Linh[1,1], Beurton-Aimar Marie[1,1], Zemmari Akka[1], Marie Alexia[1], Parisey Nicolas[1,1]

[a] *LaBRI - University of Bordeaux, UMR 5800, 351, cours de la Liberation, 33405 Talence, France*
[b] *UMR 1349 IGEPP, BP 35327, 35653 Le Rheu, France*
[c] *INRIA Bordeaux Sud-Ouest, 200, avenue de la Vieille Tour, 33405 Talence, France*

Abstract

Landmarks are one of the important concepts in morphometry analysis. They are morphological points that can be located precisely (e.g. corner of the eyes) and used to establish correspondence or divergence among morphology of biological specimens. Currently, the landmarks are mostly positionned manually by entomologists on numerical images. In this work, we propose a method to automatically predict the landmarks on entomological images based on Deep Learning methods, more specifically by using Convolutional Neural Network. We propose a CNN architecture, EB-Net, which is built in a modular way the concept of "Elementary Blocks", each made up of usual layer types of CNN. After using a custom data augmentation procedure, the network has been trained and tested on a dataset of different anatomical part of carabids (pronotum, head and elytra). In this numerical experiment, we have generated two strategies to evaluate the network and to improve the obtained results: training from scratch or applying a fine-tuning step. The predicted landmark coordinates have been compared to the coordinates of the manual landmarks provided by the biologists. The statistical analysis of the distances between predicted and manual coordinates has shown that our predictions can replace efficiently manual landmarking and allows to propose automatization of such operation.

Keywords: Landmarks, Morphometry, Deep learning, Convolutional Neural Network

1. Introduction

In the context of ecosystem services, there is an interest in examining complex interactions between the 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 [**?**], there are more and more researches about beneficial insect morphometrics [**?** **?**]. In such morphometric studies, it is common to analyze subject's shape independently of their poses and sizes [**?**]. Since the late 20^{th} century [**?**], rooted in a strong statistical background, geometric morphometrics address the study of such biological shapes [**?**]. It is an effective set of methods with several specialised softwares readily available [**?** **?**]. 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).

*Corresponding author
[1] both authors contributed equally to this work.

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

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 [?]. 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 [?]. 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 [? ? ? ? ?], speech recognition [? ? ?], question answering [?] and language translation [? ?]. Within deep learning, Convolutional Neural Networks (CNNs) are well known for their success in many computer vision tasks such as image classification [? ?] and objets recognition [? ?]. Recent success of this algorithm in human biometry [?] 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 [? ?] to butterfly/fly wings [?], zebrafish skeletogenesis [?], flower shapes [?]. They have been also concerned on medical imaging, e.g., cephalometry aims at analyzing the human cranium for orthodontic diagnosis and treatment planning [? ?].

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 [?] or 25 landmarks have been used in a research on zebrafish [? ?]. 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 [? ?]. In biology, the landmark identification problem has been appeared in the studies to analyse shape and size on the organisms [? ?], e.g., analyzing the corolla shape variation. In biomedical field, the problem of automatic landmark positioning has been addressed in cephalometry [? ?]. 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. [?] have proposed SIFT method to find the corresponding keypoints between two images. Palaniswamy et al. [?] have proposed a method based on probabilistic Hough Transform to automatically identify the landmarks in digital images of Drosophila

wings. In previous work [**?**], 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 could not be provided the landmarks to other parts of beetles as pronotum, head, and elytra. The reasons have been found down that these pictures are not simply as mandible ones. They do not only contain the considered objects but also other parts of beetle because they have been captured before dissection. Also, shape segmentation has been a trap for our method.

In recent years, deep learning has been widely used in computer vision. Using Convolutional Neural Network (CNN) 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. [**?**] 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. [**?**] 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. [**?**] 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 detail a CNN architecture that we have designed to automatically set landmarks on beetle 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 small dataset. We will also outline 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 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 **??**). 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.

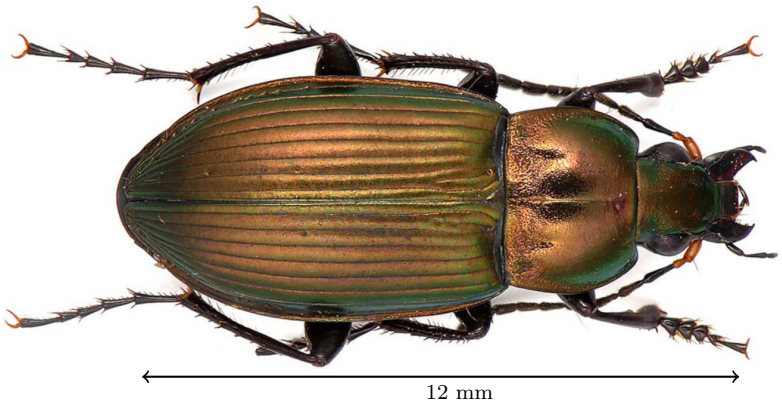
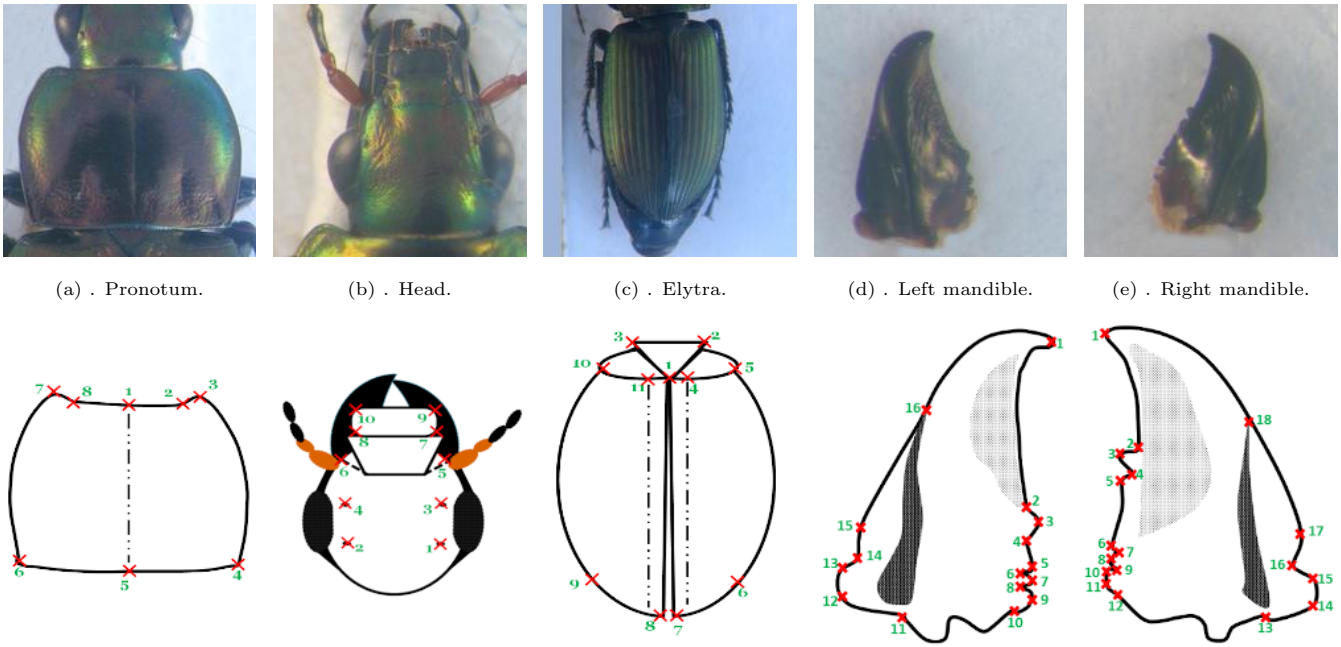


Figure 1: An illustration of the beetle.

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 **??**). 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 missing. 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 **??**). 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 [**?** **?** **?**] 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 [**?** **?**]. In order to select the

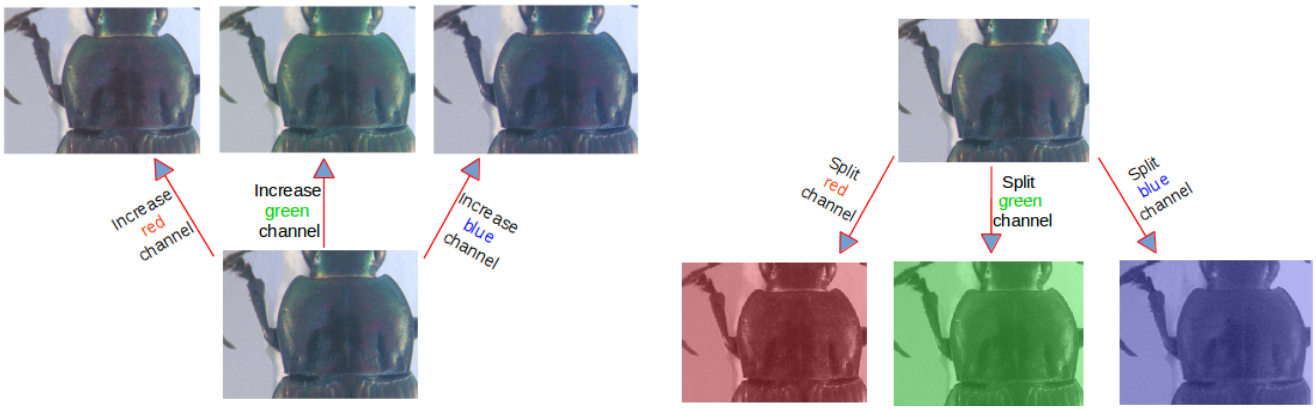


(a) . Pronotum. (b) . Head. (c) . Elytra. (d) . Left mandible. (e) . Right mandible.

(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 has been 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 **??** illustrates the two described strategies.

To perform the objective, we have observed the input size of the several CNN models [**?** **?** **?** **?**] 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 **??**. 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 [**?**]. So, the brightness of the image is normalized to $[0, 1]$.

2.2. Network architecture

Our initial trials were inspired by AlexNet architecture [**?**]. The model has 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 appear very quickly. In deep learning models, it exists another type of layer, the Dropout (DROP) layer, which is well-known to prevent over-fitting [**?**] and usually used in the last steps of the procedures. We will present now the Elementary Block (EB) model which includes all these layer types and is the core of our model architecture.

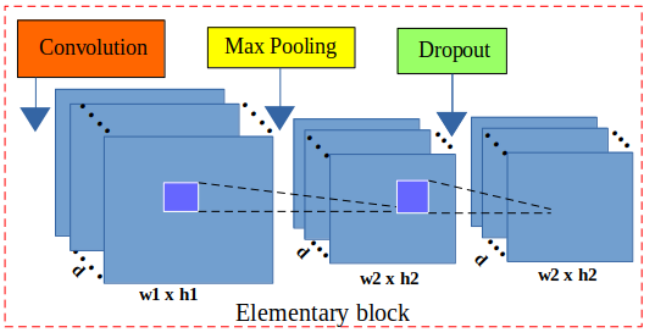


Figure 4: The components of an elementary block

Figure **??** illustrates the order of the layers in an EB item. 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, usually inserted between the FC layers to prevent over-fitting [**?** **?**], is included in our model in the extracting feature blocks to produce image noise augmentation. The DROP layer randomly drops some connections during the training process. It makes the network thinner than the original one (fewer parameters), so training a network with dropout layer is equivalent to train a set of thinner networks.

Three Elementary Blocks have been then assembled to create the 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 $(32, 64, 128)$ with a small kernel $(3 \times 3,\ 2 \times 2,\ 2 \times 2)$ 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 three 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 is equal to two times the number of landmarks that we want to predict $(x, y$ coordinates). For example, to predict 8 landmarks on pronotum, the number of outputs of the last FC layer is set at 16. Additionally, a dropout layer with the probability equals to 0.5 has been inserted between the first and the second fully-connected layers as usual [**?**]. Figure **??** resumes the EB-Net architecture.

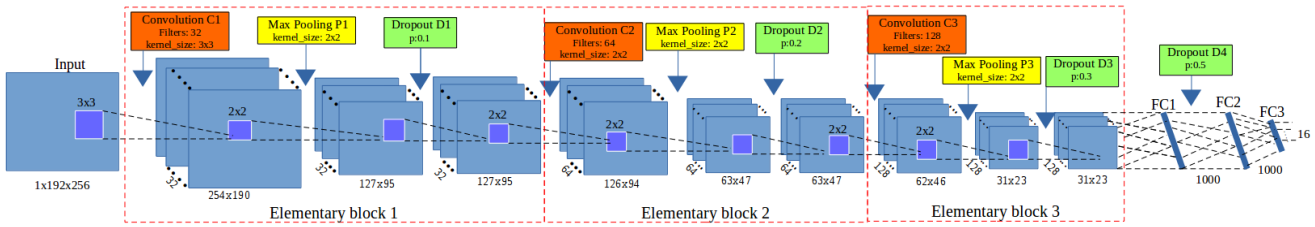


Figure 5: Elementary Blocks Network (EB-Net) architecture

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 important to design a CNN model also. These variables are related to the training of the network model, for example, the loss function, the number of epochs, the batch size, and the initialized value of learning rate. Practically, the range of the possible values is explored through empirical observations depending on the task and the dataset. In our application, we have chosen the most common optimization algorithm, Stochastic Gradient Descent (SGD) [**?** **?**]. In order to apply SGD, 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, and 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 coordinates of landmarks. The EB-Net has been implemented by using Lasagne framework [**?**], 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 technics AK-fold with $k = 9$, to select the test images. We will call a selection step a round. For each round, we take 33 images for testing step, the 260 remaining images are 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 **??**, to provide 1820 (260×7) images for training and validation steps. The cross-validation steps has been achieved after 9 rounds in total.

During the training and validation step, the 1820 images are randomly divided into training and validation sets with a ratio of 60% : 40%. At 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 in order to predict landmarks. In our case, the manual landmarks have been given by the biologists. So, they are used as ground truth to train the network, as well as to evaluate the predicted ones.

2.4. Statistical evaluation of predicted shapes

The quality of landmarks predicted by the network must be assessed in view of the desired usage for these predictions. It is quite frequent in geometric morphometrics to use the landmarks for subsequent procrustes regression [**?**] which is a method to quantify the relative amount of shape variations (among specimens) attributable to one or more factors in a linear model. Before the regression step, a generalized procrustes analysis [**?**] translates all specimens related landmarks to the origin, scales them to unit-centroid size, and optimally rotates them until the coordinates of corresponding points align at best. The resulting aligned coordinates represent the shape of each specimen. Afterward, the procrustes distance between specimens (i.e. sum-of-squared distance between corresponding landmarks) can be used for regression purpose [**?**]. The crucial step that we can take to ensure the validity of using predicted landmarks for procrustes regression is a measure of shape covariation between predicted and manual landmarks [**?**]. This problem is related to the construction of a latent variable of shape deformation and it goes further than studying, independently, each landmark correlation between predicted versus manual. Indeed, the landmarks are not independent to each other in their relative positions between specimens because they form a shape. Hence, we are looking for covariations between predicted and manual shapes that are positive, as high as possible and statistically significant.

3. Results

3.1. The first evaluation

EB-Net has been firstly trained and tested on the pronotum images. Figure **??** shows the losses of training and validation processes in one round on pronotum images. The blue curve is the training loss, and the green curve is the validation one. As we can notice, learning is effective and over-fitting does not appear. We can assume that EBs with the help of dropout layers have properly worked to prevent over-fitting.

Table **??** shows the losses of the 9 rounds of EB-Net training on pronotum images. We can observe that although the image dataset changes through each round, the loss values remain stable and always under 3×10^{-3}. Based on the success of EB-Net on pronotum images, we have employed it on the two sets of head and elytra images with the same result qualities.

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

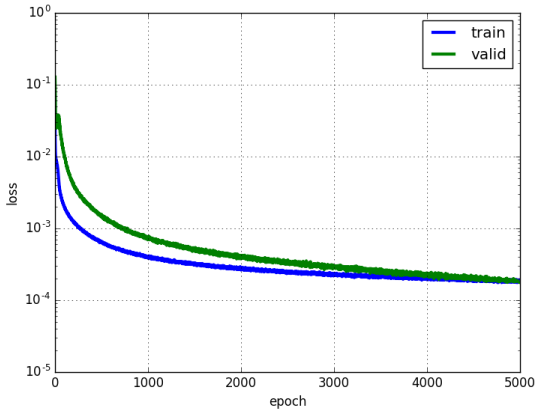
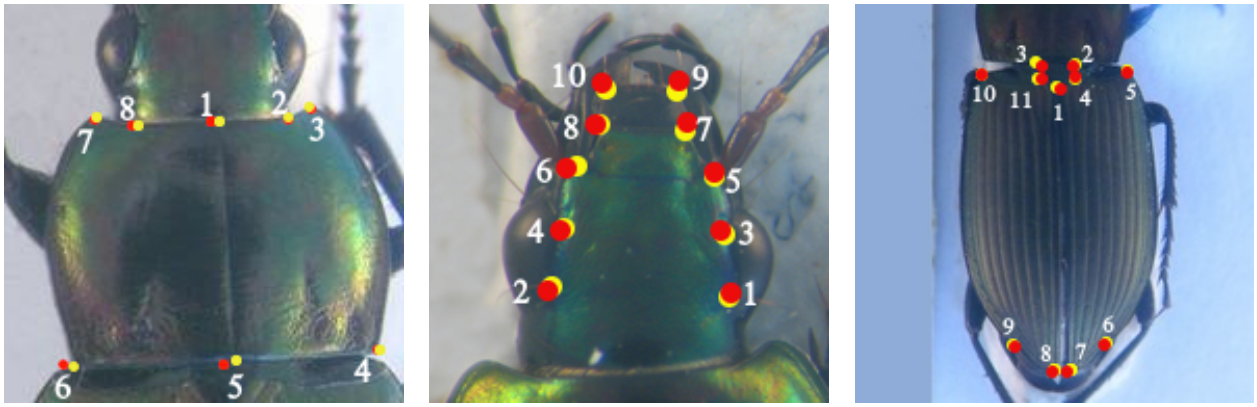


Figure 6: The losses during training EB-Net on pronotum images.

The quality of performances of a CNN is mainly measured from loss, accuracy. But in this work, we also need to appreciate correctly where the estimated landmarks are positioned by the network. So, the trained model of each round has been used to predict the landmarks in images of the corresponding testing set. Then, the coordinates of outputted landmarks are evaluated by comparing with the manual ones. Figure **??** illustrates 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.



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

Figure 7: The landmarks on three parts of beetle. The red/ yellow points present the predicted/ manual landmarks.

In order to provide a global evaluation, the distances (in pixels) have been calculated between the manual landmarks and the corresponding predicted ones, and the average value at each position has been taken into account. Table **??** shows the obtained average distance for pronotum, head and elytra. With the image's size of 256×192, we can consider that an error around 1% of image size ($\approx$ 2 pixels) could be accepted. Unfortunately, our results exhibit the average distances from 4 to 5 pixels ($\approx$ 2% of error).

It is worth to note that an average value could reflect several different cases for example values closed together (small dispersion) or two sets of values very far (large dispersion). In order to see in which situation we are, Figure

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.

?? shows the distribution of distances between the manual and predicted landmarks for the best and the worst cases in each set of images (the green and red values in the table). 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. Regarding these figures, it is clear that even a large number of points remains inside the range from 0 to average value, some points are widespread between the average and 15 or 25 pixels. The question is: is it possible to lower the mean and to lower the dispersion above the mean value.

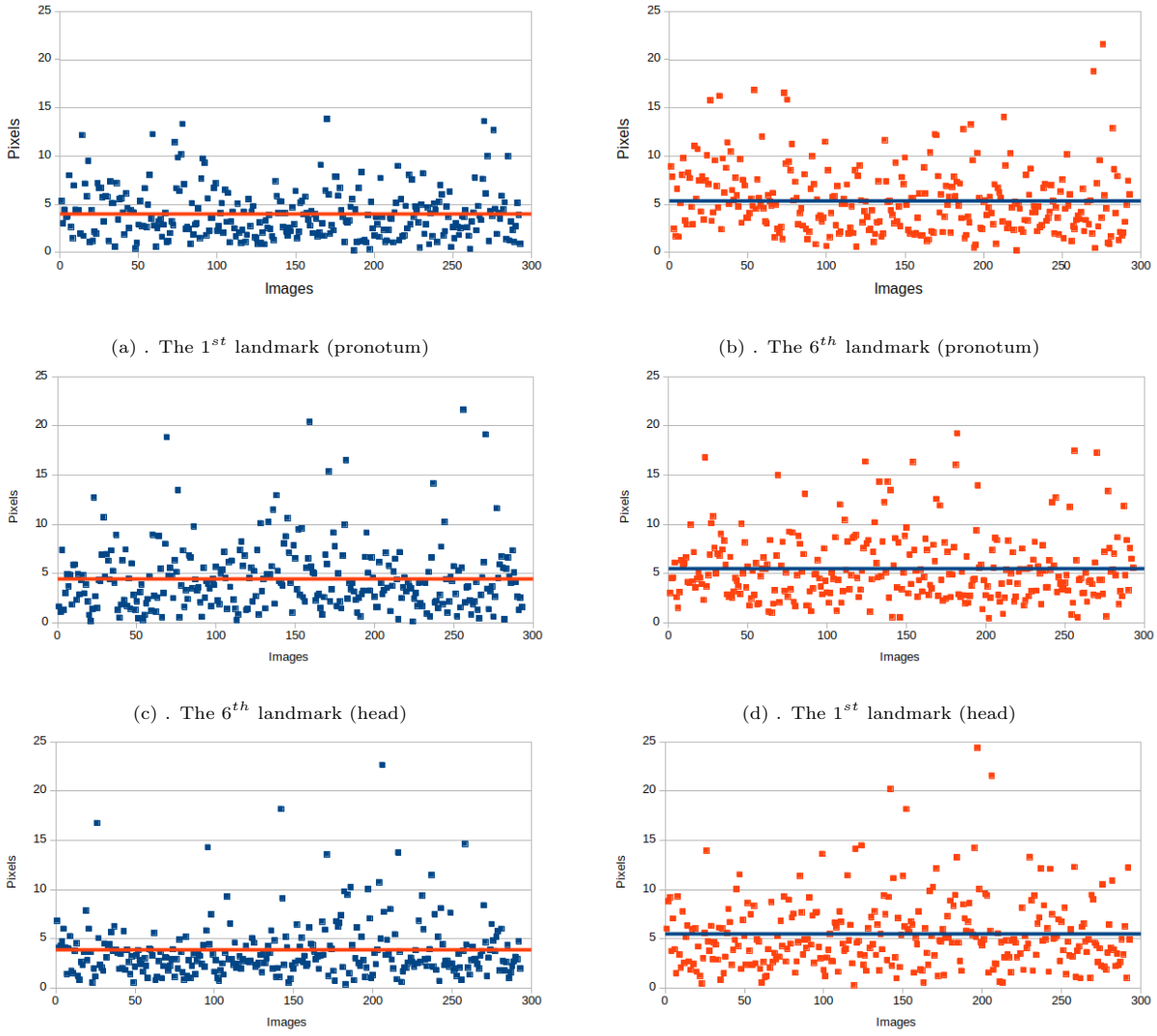
3.2. Transfer learning to improve performances

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

Transfer learning is a technique in deep learning to re-purpose a model which has been designed for a specific task (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:

- **Use CNN as a fixed feature extractor**: Taking a CNN pre-trained on a large dataset, then removing the last fully-connected and using the rest layers of CNN as a fixed extractor for the new dataset.
- **Fine-tuning a CNN**: This scenario begins 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 to be adapted between the two tasks. These parameters could be the size of input images, the number of outputs, or the parameters of each layer

In Deep Learning community, ImageNet is known as a large dataset with more than 14 million images of thousands of categories [**?**]. Notable models have employed ImageNet as training data to solve different tasks [**?** **?**]. These pre-trained models have been widely shared in deep learning community as a source to re-use the features of ImageNet dataset. As a preliminary work, we have tested several well-known models, e.g., AlexNet [**?**], VGG-16 [**?**] that have been trained on ImageNet [**?**]. Unfortunately, the features from ImageNet seem to be not relevant for our application because the image features mainly concern the detection of global shape of the objects whereas



(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 8: The distribution of distances for the best and the worst cases of the three parts

landmarks can be considered as local features [**?**]. 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 to improve the quality of predicted landmarks coordinates.

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, the training datasets have been freely published. We have decided to choose such facial keypoints dataset to pre-train EB-Net and then to transfer the parameters values to fine-tune them for beetle's images.

The selected dataset to pre-train EB-Net has been published for a challenge[3] on the Kaggle website. It includes

[2]Deepfake Detection Challenge/ Facial Keypoints Detection

[3]https://www.kaggle.com/c/facial-keypoints-detection

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 **??** shows 4 examples of face images and their landmark positions in the dataset.

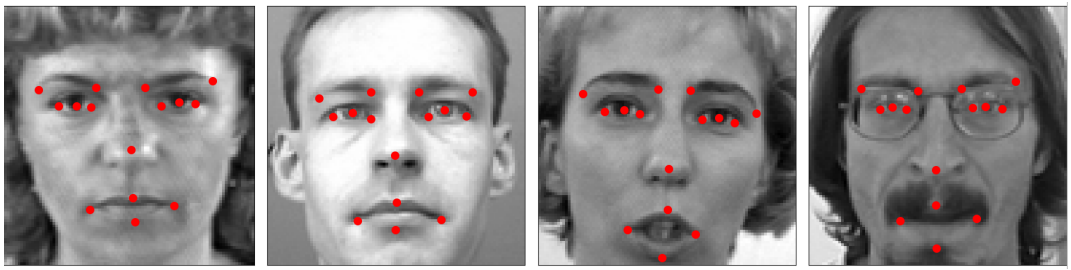


Figure 9: The samples (face and manual landmarks) in the dataset that we used to pre-train EB-Net.

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). For hyper-parameters, just the number of epochs have changed to reach the value of 10000 epochs. In table **??**, we show the RMSE score and the comparison of the effectiveness between EB-Net and the three best ones that have been published for the challenge.

Team	Olegra	Trump	Enes	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.

It is worth to note that the three scores have been obtained by testing the models on a private set of images that are not freely available. To do the test, we have had to use some images getting from the public data set, 100 images have been chosen to do that. Comparing with these scores, the three models present better results than us but we are very close. The RMSE score is around 1 pixel that is an acceptable error to display the landmarks on the images. Consequently, the pre-training of EB-Net is considered as correct and we have decided to re-use the pre-trained parameters 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 a 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 parameters 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 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, far from the beetle images. Consequently, 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 it is easy to suppress it with a pre-processing operation, we have chosen to remove a part of the background region (without any beetle's part) instead of down-sampling our pictures. So, the new beetle images are finally set to 192×192 pixels. The EB-Net parameters will be settled to take into account the differences of the values between the pre-training and fine-tuning steps. To declare the modification, we set the stride value of the first convolutional layer to 2 (usual way to do [**?**]).

3.2.3. Fine-tuning results

We present all the obtained results for the three beetle parts: pronotum, head and elytra, in the same way than in the previous results to provide in order to provide an explicit comparison. Tables **??**, **??**, **??** 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 has been 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. In all these tables, all the average distances have decreased between 1 and 1.5 pixels in both of three sets of images. Clearly, the fine-tuning process has improved the landmark predictions for each group. The best-predicted positions (green ones) has changed 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

Considering on each landmark position, the level of 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 predictions have gained from 36.85%/24.94%/18.36% to 19.18%/12.15%/12.82% on pronotuom, head, and elytra, respectively.

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

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

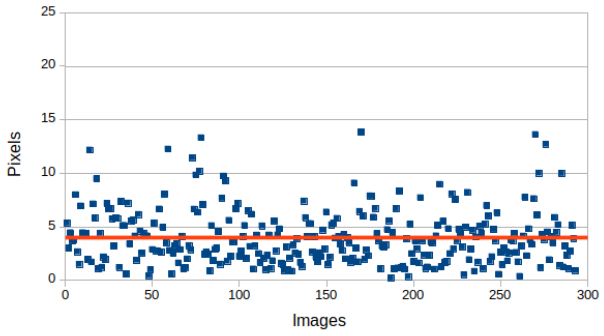
As we have mentioned before, the mean value hides different situations when deeply going to the analysis. So, we have checked again these distributions. Figure **??**, **??**, **??** show the distribution of distances on two samples in each set of images: pronotum, head, and elytra, respectively. In 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 respectively. The blue/red lines in charts present the average values. We can observe that the distances have been reduced with the help of fine-tuning process whatever they were the low or the high values. We have gained more points (in the chart) in the range from zero to the average one, and the number of extreme values have been decrease. 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. However, it exists some difficult cases that the model could not optimize in the head or the elytra parts.

To illustrate the results, Figure **??** 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.

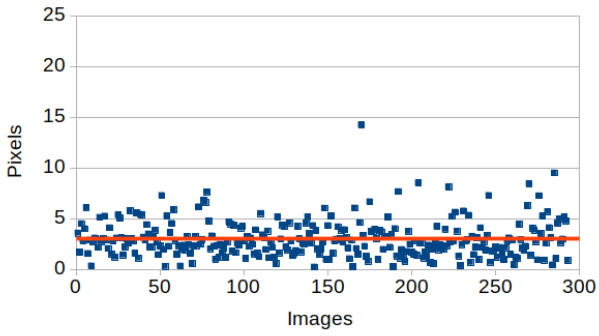
3.2.4. Results on mandibles

In previous work [**?**], mandible images belonging to the same dataset have been analyzed with the help of a pipeline of classical image analysis procedures and based on a segmentation step. We have evaluated the performances of EB-Net on this dataset. It is worth to note that the fine-tuning process has been applied for this experiment in the same way than the other parts: pre-training EB-Net with the facial key-points dataset and transferring the parameter values to fine-tune on mandibles.

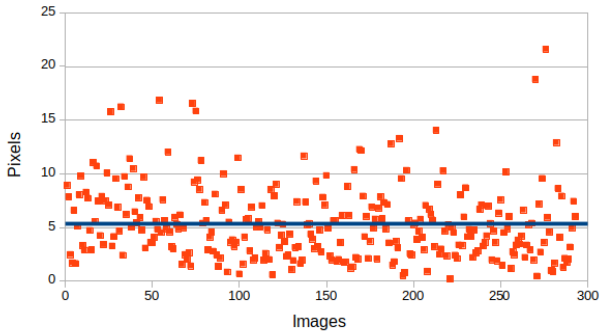
To achieve the comparison, the obtained values in [**?**] have been re-computed to match the new size of the images by scaling these errors (distances) with the same ratio that we have used to down-sample images. Then, the average value has been computed for each landmark's position. Figure **??** shows the comparison of average distance at each landmark position between the obtained results of the two methods (deep learning and image processing methods). The red curves illustrate the average distances which have been obtained from image processing technique



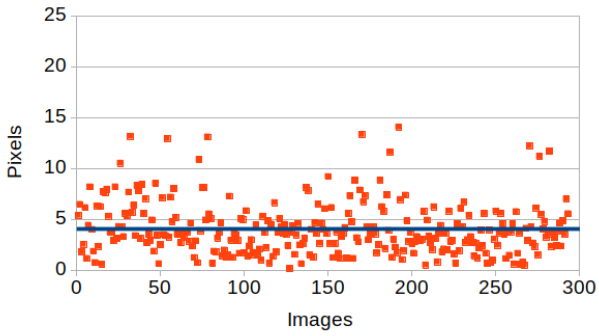
(a) . 1^{st} landmark (from scratch)



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

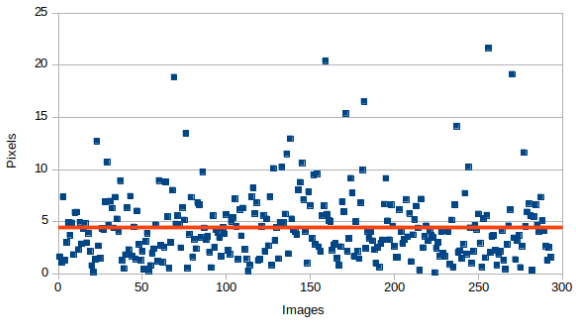


(c) . 6^{th} landmark (from scratch)

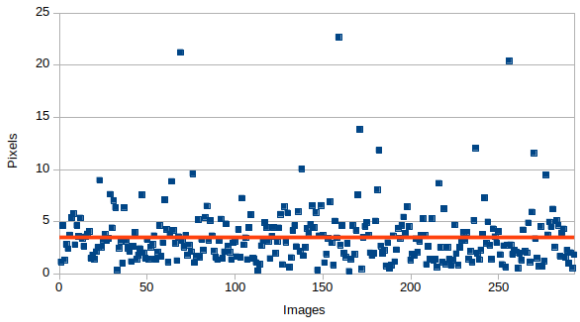


(d) . 6^{th} landmark (fine-tuning)

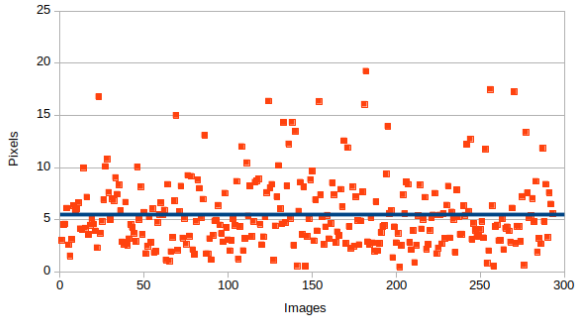
Figure 10: A comparison of distances distribution of the 1^{st} landmark and the worst case (6^{th} landmark) on pronotum images.



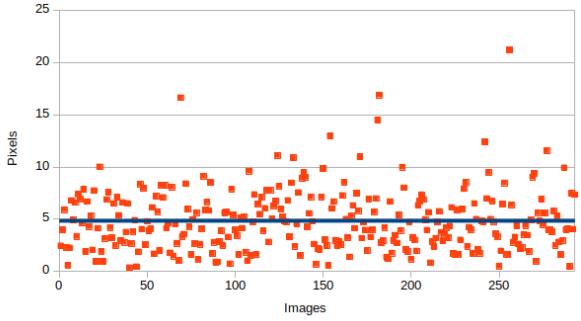
(a) . 6^{th} landmark (from scratch)



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

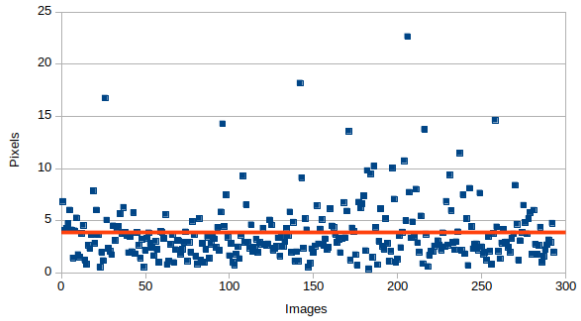


(c) . 1^{st} landmark (from scratch)

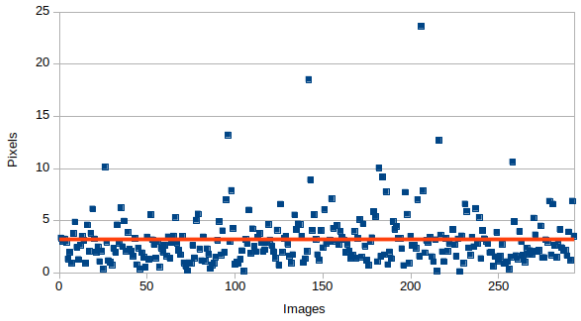


(d) . 1^{st} landmark (fine-tuning)

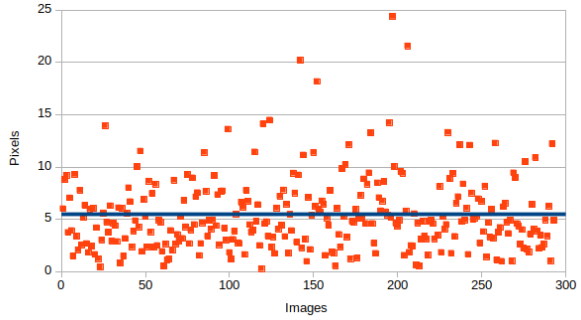
Figure 11: The distribution of distances of all head images on 1^{st} landmark and 6^{th} landmark.



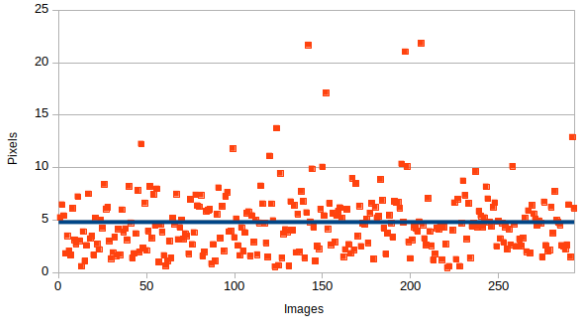
(a) . 1^{st} landmark (from scratch)



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

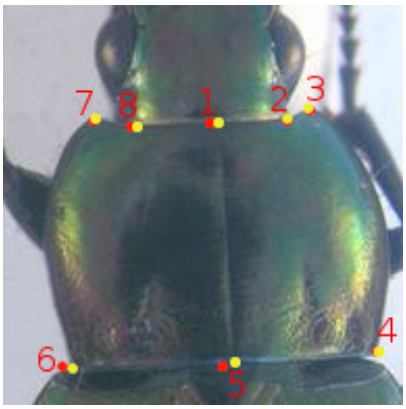


(c) . 8^{th} landmark (from scratch)

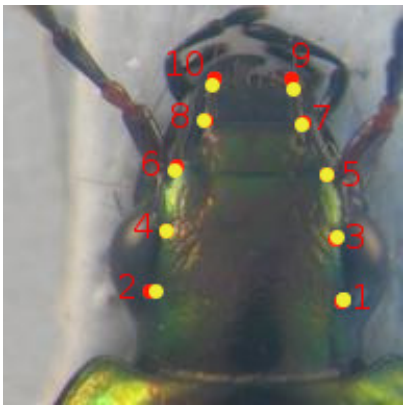


(d) . 8^{th} landmark (fine-tuning)

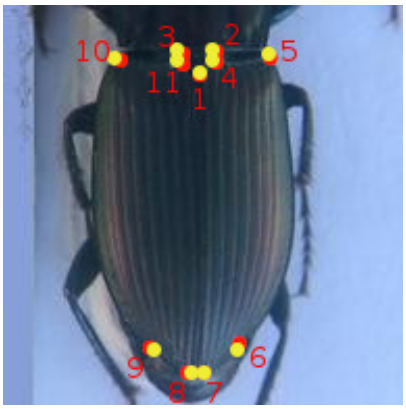
Figure 12: The distribution of distances of all elytra images on 1^{st} landmark and 8^{th} landmark.



(a) . Pronotum



(b) . Head

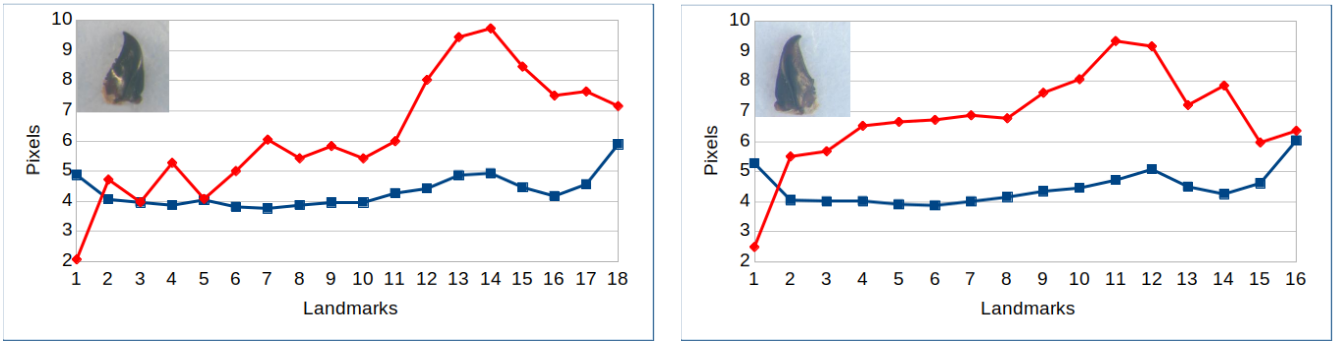


(c) . Elytra

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

while the blue curves present the results of fine-tuning process.

For the right mandible (Figure **??**), EB-Net has got better results than the image processing pipeline for all landmarks (except the first one). One can notice that the first position represent the tip of the shape, and it is easy to detect after a segmentation step. EB-Net works without segmentation, it is why the result for this point is similar to the other ones. Moreover, it is clear that the results of EB-Net exhibit of less variation than the previous method. For the left mandible (Figure **??**), the pipeline has provided the worst results than for the right one. The hypothesis has been given that the size of this part varies more than the right one. On the opposite side, we can



(a) . Right mandible

(b) . Left 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.

observe that EB-Net has produced results similar to the right one.

To summarize, the average value at each landmark position is better than the previous ones in both two cases: left and right mandibles, with the help of the fine-tuning process. The predictions are more stable and significantly improved: it has reduced the range between the highest and the lowest values in both of two sets, for example, from 6.85/7.66 pixels to 2.17/2.13 pixels for the left and right mandibles, respectively.

3.2.5. Predicted shapes : comparing 'from-scratch' to fine-tuning

We have measured shape covariations between predicted landmarks and manual ones [**?**], assessing the performance of fine-tuned and 'from scratch' networks (described previously) on each anatomical parts. The shape covariation is expressed as a correlation coefficient and an associated p-value.

	From scratch		Fine-tune	
	cor	p-value	cor.	p-v.
Pronotum	0.4118	0.001	0.7424	0.001
Head	0.5145	0.001	0.7337	0.001
Elytra	0.2513	0.407	0.3025	0.069
Left mandible	0.2666	0.173	0.3112	0.057
Right mandible	0.3071	0.239	0.4597	0.002

Table 7: Shape covariation between predicted and manual landmarks : shape covariation as a correlation (cor.) and an associated p-values (p-v.)

We can see from table **??** that shape correlations varies between 0.2513 and 0.7424 and that 5 out of 10 of them are statistically significant (i.e. p-value < 0.05). For all the anatomical parts, the fine-tuning improved the shape correlations. For all the correlations with p-values > 0.05, the fine tuning improved at the same time the correlation strength and the statistical significance. Best results are obtained for fine-tuned networks predicting pronotum's and head's landmarks with high significant correlations, 0.7224 and 0.7337 respectively. Right mandible predictions by fine tuned network is also of interest with a mild correlation (0.4597) but still significant (p-value $= 0.002$). We

end up with a majority ($\frac{3}{5}$) of anatomical parts were the predicted landmarks are good candidate for replacing manual ones in any procrustes regressions.

4. Discussion and perspectives

In this article, we have presented a convolutional neural network, EB-Net, for automatically predict the landmarks on entomological images. It is based on three times the repetitions of a generic block followed by fully connected layers. The block consists of a convolutional layer, a max-pooling layer, and a dropout layer. In the first step, the EB-Net has been separately trained on each part of the beetles. In order to improve the results, the fine-tuning process has been added by pre-training EB-Net on a facial key-points dataset before transferring 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 have shown that using CNN to predict the landmarks on biological images leads to satisfying results without need for the complex pre or post processing steps on the studied objects. These predictions are statistically significant enough to replace manual landmarks in procrustes regression studies, which are fairly common in morphometric studies e.g. two classic methodological papers on the subject[**?** **?**] have a combined 299 direct citations and we can assume that most citations are indirect.

This work has addressed a complex task in computational biology: automatic landmark digitalization. The interest in this topic is not new, but the relevant algorithms and computational power have only been available fairly recently [**?** **?** **?** **?**]. Even so most methods are seldom used by biologist and even so by entomologist. Maybe it is because the software tools are too complex and the necessary datasets not always available (e.g. too few training images and related manual landmarks). We show here that a neural network with a relevant architecture can provide a high quality of results in only a few methodological steps. We took care of constructing this network using standard software tools to ease any technological transfer. Also, our model, EB-Net, can easily work with a limited number of images and improve its prediction results by fine-tuning the parameter values obtained from pre-training on another dataset.

One can note that the elementary block that we used to build EB-Net, is generic so it's easy to adjust the suitable number of blocks for ones applications depending 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 slightly improved, but we have spent more resources and computing times to reach this improvement. The average distances have been improved by 0.5 pixels. However, this change was statistically insignificant (data not shown). 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. This elementary block approach is not, formally, a neural architecture search [**?**] i.e. a method for automating the design of a CNN, but it is an easy to implement practical solution.

In this application, we focused on a carabid species which is linked to studies in agroecology and conservation ecology. But EB-Net has the potential to be applied to other insects and invertebrates as well. So in the future, we plan to test our model on more datasets that are linked to other ecological fields most notably invasive species and quarantine species (e.g. nematods). All the implementations, EB-Net model and EB-Net parameters, are available

freely [4]. 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 in biomedicsl imaging most notably MRI images analysis.

[4] https://github.com/linhlevandlu/cnnBeetles