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

Le Van Linh[b,c,*], Zemmari Akka[b], Marie Alexia (?)[a], Beurton-Aimar Marie[b,1], 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. Finding landmarks is not only used to measure the shape of the object but also applied to analyze the inter-organisms variations. Currently, the landmarks are mostly determined manually by the biologist. In this work, we propose a method to automatic predict the landmarks on biological images: Deep Learning, more specific is Convolutional Neural Network (CNN). We proposed a CNN architecture which was built from the "elementary blocks". Each block is made up of some popular layers of CNN. These works have also introduced another procedure to augment the dataset which can see a little bit small in our case. The network then trained and tested on a dataset includes three parts of beetle (pronotum, head and elytra). In the experiments, we apply two strategies to evaluate the network and to improve the obtained results: training from scratch and applying a fine-tuning step. This is an extension from our works that have been presented at the MAPR 2018 conference. 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 are 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

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

common and ubiquitous beneficial insect of north-western France, *Poecilus cupreus* (Carabidae). It is considered 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, usually 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, higher layers of representation enlarge aspects of the input that is important for the computational task (classification, regression, ...) and suppress irrelevant variations. As supervised learning algorithms, they use gradient descent optimization method to update the learnable parameters via backpropagation. Deep learning algorithms have proved to be very efficient in a wide variety of domains, notably image recognition and classification [14, 15, 16], speech recognition [17, 18, 19], question answering [20] and language translation [21, 22]. 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 [23, 24]. Recent success of this algorithm in human biometry [25] lead us to believe in its potential for insect morphometrics.

1.1. Related works

In biology, landmarks could be divided into three types. Some landmarks are cleary defined on a structure (type I), others that are more ambigious and usually describe maxima of curvature (type II), and those that are geometric construction generated from lines (type III). In geometric morphometry analysis, the detection of landmarks in biological images is a necessary step in many researchs.

In computer vision, landmark localization is usually studied on human faces where we identify some points corresponding to important 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 een addressed in cephalometry []. The standard methods in this domain are based on the combination of template matching and prior knowledge information from feature extraction steps. More recently, several algorithms have been succeed by using machine learning techniques followed by global landmark structure refinement [].

Landmark or point of interest is a specific point that may contain the useful information. For example, the tip of the nose or the corners of the mouth are landmarks on human face. In image processing, we can consider two kinds of cases: the object of interest can or not be segmented. Setting landmarks can not be achieved in the same way depending on which situation we are. When segmentation can be applied, Lowe et al. [26] have proposed SIFT method to find the corresponding keypoints between two images. Palaniswamy et al. [27] have proposed a method based on probabilistic Hough Transform to automatically locate the landmarks in digital images of Drosophila wings. In our work [28], we have proposed a method which was extended from Palaniswamy's method, to determine landmarks on mandibles of beetles. The mandibles of beetle have the simple shape and easy to segment. We have obtained good enough results about determining the landmarks automatically on mandibles. Unfortunately, this method can not be applied to other parts of beetles than the pronotum seems is segmentation has too many noises.

In recent years, deep learning is known as a solution in computer vision. Using convolutional network to determine the landmarks on 2D images has achieved better results. It seems that this way offers an effective solution to face with the images that we have difficulty in segmentation. In the landmarking context, Yi Sun et al. [29] have proposed cascaded convolutional neural networks (three-levels) to predict the facial points of interest on the human face. Each level consider from global to local regions on the face to determine the landmarks. Zhanpeng Zhang et al. [30] 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. In a similar idea to apply deep learning algorithms, we proposed a CNN to predict the landmarks on beetle's images.

1.2. Contributions

As we mentioned before, the works presented in this article are extended from our previous works [31], where we have proposed a CNN to predict the landmarks. In this article, we first re-present our experimental data and the methods to augment the number of images in our dataset (Section 2.1). Next, we re-describe the process that we have designed our model architecture (Section 2.2). Then, we present the two scenarios to apply our model for predicting the landmarks on beetle images (Section 3). We give a comparison of the results that we have obtained on mandibles images too. Finally, we make some discussions before concluding.

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 show the network architecture model that we have designed to predict the landmarks in the beetle's images.

2.1. Dataset and preprocessing

In order to provide the experiment 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 the size of 3264×2448 pixels.

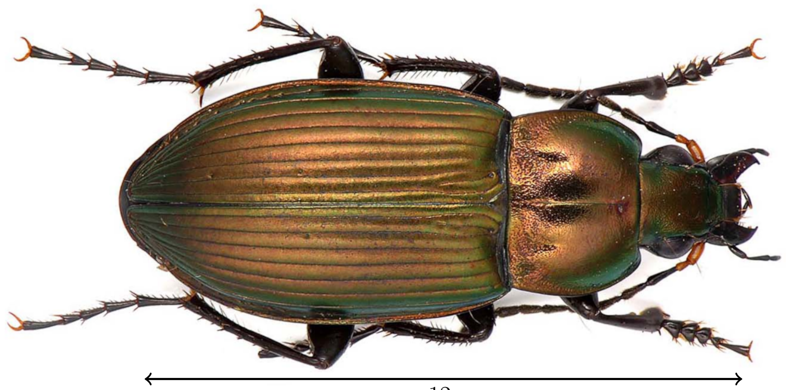


Figure 1: An illustration of the beetle.

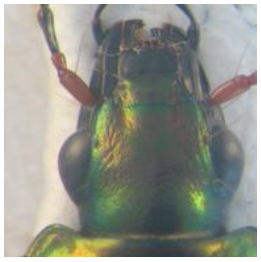
In the next step, morphological landmarks were first set manually on the dorsal views of each body part of the beetles (head, pronotum, elytra, right and left mandibles). The morphology of each body part was processed and analyzed separately in order to limit variation resulting from their relative positions 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 been trained on a large dataset with an enormous number of data samples before using it 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, medical. 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 augmented the number of images in each set of images.

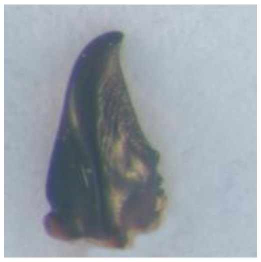
Most often in deep learning applications, dataset augmentation uses operations such as translation, rotation, or scaling, which are well-known efficient to generate the new version of existing images [14, 32]. However, these operations can be invariant in some cases [32]. In order to select the right 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.

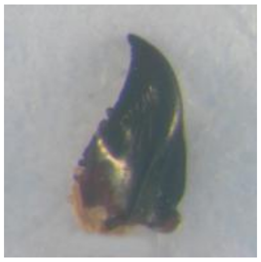
The first strategy was applied to change the value of each channel in the original image. According to this, a constant have been added to a channel of RGB image for each time. For example, if we add a constant $c = 10$ to

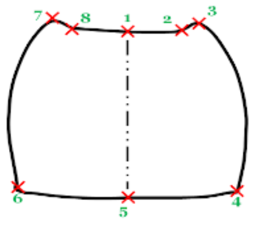
(a) . Pronotum.

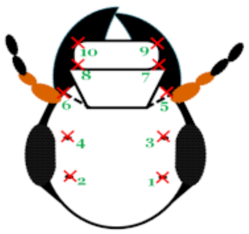

(b) . Head.

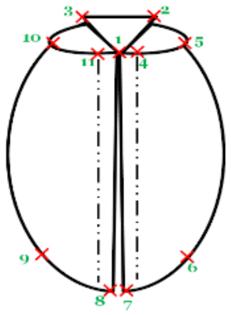
(c) . Elytra.

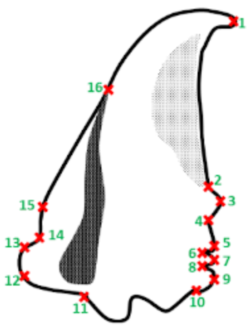

(d) . Left mandible.


(e) . Right mandible.


(f) . Pronotum.


(g) . Head.


(h) . Elytra.


(i) . Left mandible.

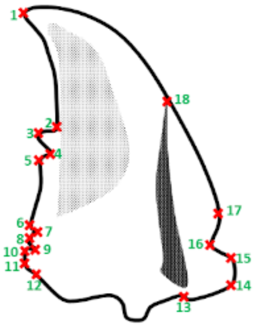
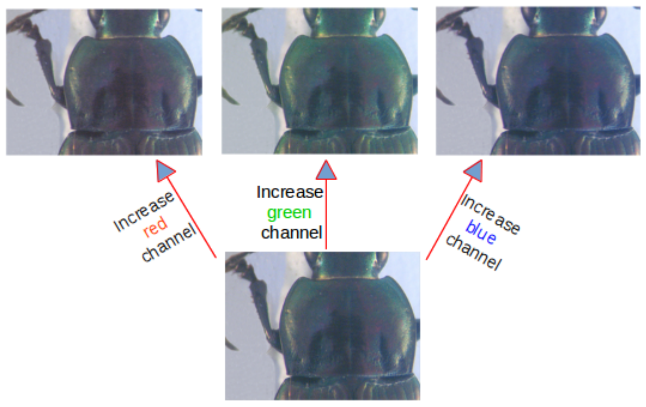

(j) . Right mandible.

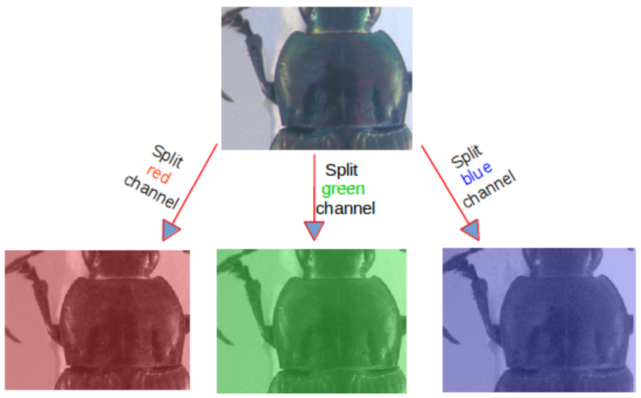
Figure 2: The sample images in our dataset (top) and manual landmarks on each part defined by biologists (bottom).

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



(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.

To perform the objective, we have observed the input size of the several CNN models [14, 15, 25, 29] and noticed that 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 has 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. According to [33], the brightness of the image is normalized to $[0, 1]$ and the coordinates of the landmarks are normalized to $[-1, 1]$ before giving to the network.

2.2. Network architecture

The first model were inspired by AlexNet architecture [14]. It received gray-scale image with the size of 256×192 as the input. The image was analyzed by three repeated structures of a convolution (CONV) layer followed by a maximum pooling (POOL) layer. As usual, the depth of CONV layers are set to increase from the beginning to the end with a small kernel. In our model, we have chosen $32, 64, 128$ for the depths and $3 \times 3, 2 \times 2, 2 \times 2$ for the kernel's sizes of CONV layers, respectively. Using POOL layers after the CONV layers is a frequent occurrence in CNN [14, 15, 25]. This work towards to two advantages: reducing the spatial size of the representation to decrease the number of parameters as well as saving the computing time, and to prevent the over-fitting during the training process. In the first model, we have used the common form for a POOL layer: a filter of 2×2 with a stride of 2 pixels. With these parameter values, the spatial size of the image will be halved after every POOL layer. In order to extract the global relationship between the features and to provide the prediction, three fully-connected (FC) layers have been added after the combination of CONV-POOL. The first FC layer takes all features from the last POOL layer as the input for computing. Then, the computed results will be passed through the activation functions to provide the outputs. The second FC layer receives the outputs of the first FC layer as the input. The process at this layer is the same than the first one. The third FC layer accepts the outputs of the second FC layer, then they will be used to compute the input. It is worth to note that we keep the linear value for the output of the third FC layer. The number of outputs at each FC layers is set to $500, 500$, and X, respectively. The number X equals to two times the number of landmarks that we want to predict. For example, to predict 8 landmarks on pronotum, we have set 16 as the outputs of the last FC layer. The first model has been trained and tested on our pronotum images. Nevertheless, the obtained results have not been considered good enough to use it. The main problem comes from the appearance of over-fitting during the training process. The detail of this experiment will be discussed in Section 3.1.

As an attempt to remove over-fitting, we have increased the number outputs of the first two FC layers from 500 to 1000 with a hypothesis considering more features. However, the obtained results have not changed, the over-fitting was not suppressed.

In order to build the third model, we have defined a concept of Elementary Block (EB). 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. As usual, the dropout layers are inserted between the FC layers to prevent the over-fitting as we have seen in the success stories [13, 14], but in our case, we have added them in the extracting features blocks (after pooling layers) to create some kinds of image noise augmentation. The objective of this work is to provide more studied

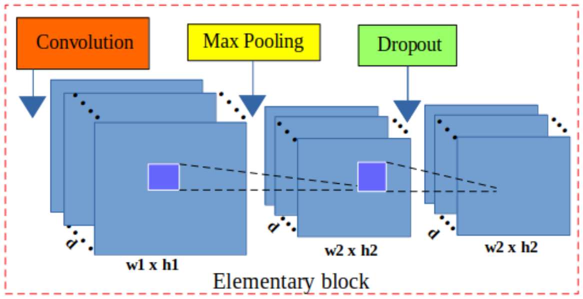


Figure 4: The components of an elementary block

cased in each training iteration. Besides, dropout layer will randomly drop some connections during the training process. It makes the network thinner than the original one (less parameters), and training a network with dropout layer is equalivalent to train a set of thinner networks. These works have significantly reduced over-fitting and given more major improvements than other regularization methods [34].

For our critical objective, we have initially assembled three Elementary Blocks to create the third model, so-called Elementary Blocks Network (EB-Net). The used parameters of the convolution and the pooling layers in the elementary blocks have been kept in the same configuration as the second model. The probabilities of the dropout layers, which were added after the pooling layers, are set increasing: $0.1, 0.2$, and 0.3, respectively. Following the three EBs is three fully-connected layers with a similar number of outputs as the second model. Moreover, a dropout layer with the probability equals to 0.5 has been inserted between the first and the second fully-connected layers. Figure 5 shows the structure of the EB-Net.

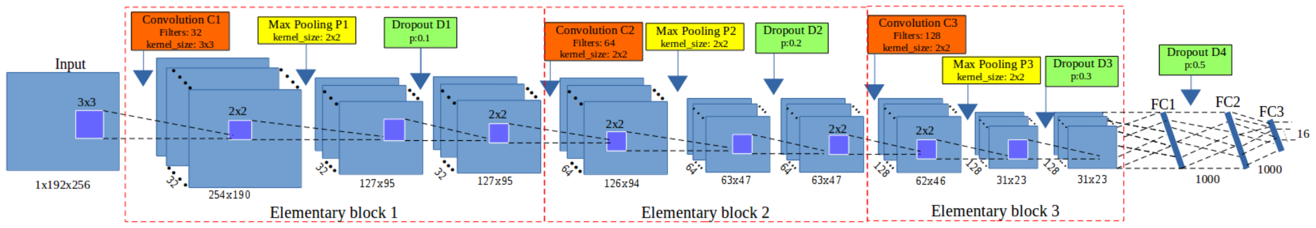


Figure 5: Elementary Blocks Network (EB-Net) architecture

In a CNN model, besides the model-specific hyper-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 train a network model, for example, the loss function, number of epochs, or initialized values of learning rate, batch size. Practically, these values are discovered through empirical observation depending on the task and the dataset. In our application, we have chosen gradient descent as the optimized algorithm as usual studies [14, 25]. Consequently, the learning rate begins from 0.03 and to stop at 0.00001; whereas the momentum rate is updated from 0.9 to 0.9999. During the training, the learning and momentum rates are adjusted to fit with the number of epochs. Besides, the Root Mean Square Error (RMSE) has been chosen as the loss function because it is employed for regression problems where outputs represent undistinct values as in the case of landmark coordinates.

2.3. Setting and training EB-Net

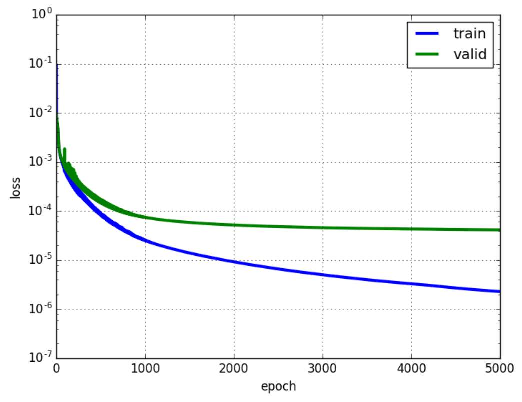
In order to provide the predicted landmarks for all images, we have applied cross-validation technique to select the test images, we will call a selection step is a round. For each round, we take 33 images and keep them for testing, the 260 remaining images will be used to train and to validate the network 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 images for these two steps. 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 two sets with a ratio of 60% : 40% and used for the corresponding processes (training : validation). 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 trained model and predicted landmarks will be provided. 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. The EB-Net has been implemented by using Lasagne framework [35], and trained in 5000 epochs on Linux system by using a NVIDIA GPU (Titan X) card.

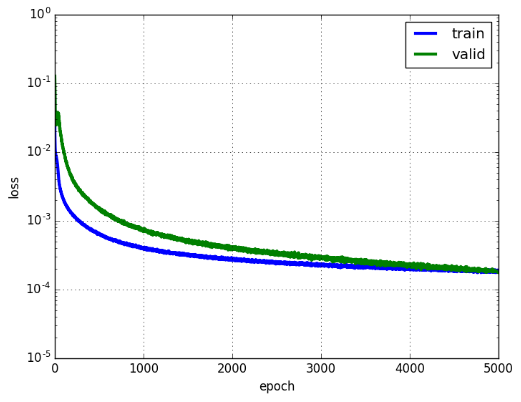
3. Results

3.1. The first test

As mentioned in Section 2.2, we have designed three models for our objective. In order to select the best one for our goal, we have first tested the three models on pronotum, before applying the best one on the two other parts of beetle (head and elytra).



(a) . Losses of the first model.



(b) . Losses of EB-Net.

Figure 6: The losses of two models on pronotum images.

Figure 6 shows the losses during the training and validation processes of two models (the first one and EB-Net) on pronotum images. The blue curves are training losses, and the green curves are validation losses. In the first model (Figure 6a), the training loss ables to decrease along with the number of epochs, but the validation loss is stable after 2000 epochs. Clearly, the over-fitting has appeared in the fist model. In the second model, no concrete change appears in the curves even the parameters of fully-connected layers have been modified. We have also checked the losses of the second model, the result is also the same as the first one. So, we have not mentioned it in this document. On the opposite side of the first two models, the losses of EB-Net (Figure 6b) is really different. They are far from the beginning, but become more close at the end of process. One can note that the over-fitting has disappeared in this architecture. Therefore, we can assume that using elementary block works well to prevent over-fitting in our case. For these reasons, we have decided to use EB-Net for producing the landmarks.

In order to achieve the best quality of predicted landmarks, we have tested EB-Net in two different strategies: *training from scratch* and *transfer learning*. The following sections describe specific results from these processes. Initially, we consider the losses during the training and validation processes. Next, we compare the coordinates of predicted landmarks and corresponding manual ones by calculating the distance between them. Then, the average distance is taken into account for each landmark position. We also display the landmarks on the image. Ultimately, the distribution of distances will be discussed in some cases of landmarks.

3.2. Training from scratch

In the first strategy, EB-Net has been separately trained on three sets of images: pronotum, head, and elytra. 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. On head and elytra images, we have obtained the same situation when we trained the EB-Net on these two sets of images.

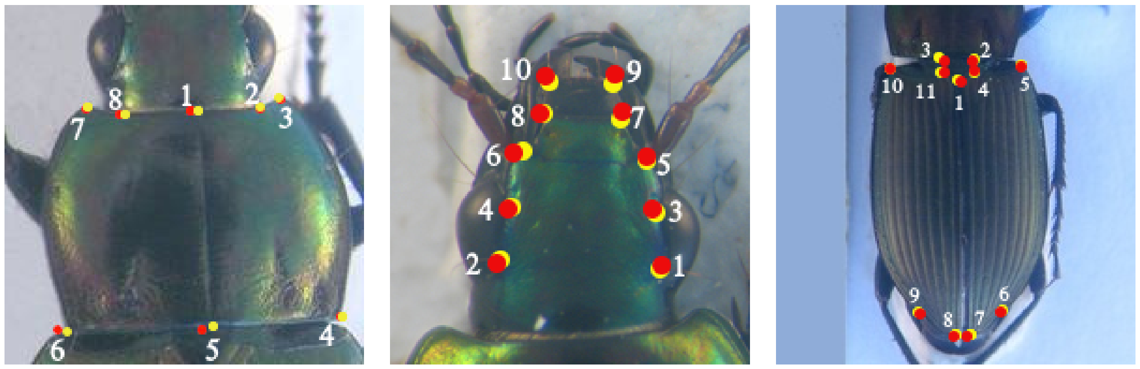
Round	1	2	3	4	5	6	7	8	9
Training loss	0.00018	0.00019	0.00019	0.00021	0.00021	0.00019	0.00018	0.00018	0.00020
Validation loss	0.00019	0.00021	0.00026	0.00029	0.00029	0.00018	0.00018	0.00021	0.00027

Table 1: The losses during the training of the third model 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.



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

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

To illustrate the points, Figure 7 shows the predicted 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.

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 8 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.

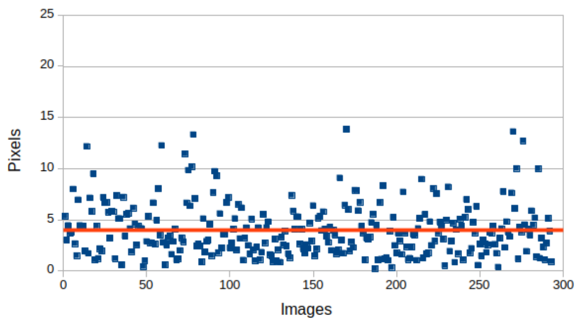
The EB-Net has achieved good outcomes in most cases, but it exists of several complex images in our dataset that the model meets a difficulty to recognize. It explains why some high distance values have appeared in Figure 8. The next step is to improve the predicted results.

3.3. Transfer learning process

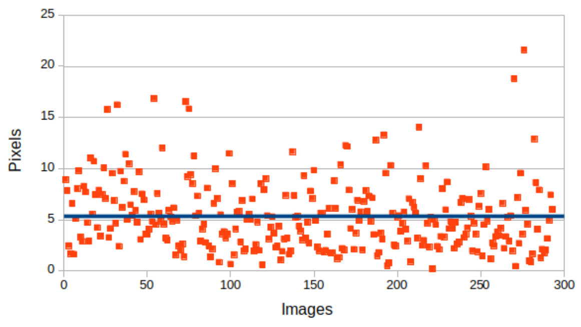
Working with deep learning requires not only to design a good architecture but also to provide a large dataset to train and to test the model. Practically, this is a potential problem in some application domains as in biology. In section 2.1, we have augmented the number of images in our dataset and 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. We describe this process in this section.

Transfer learning [36, 37] 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). Chosing 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:

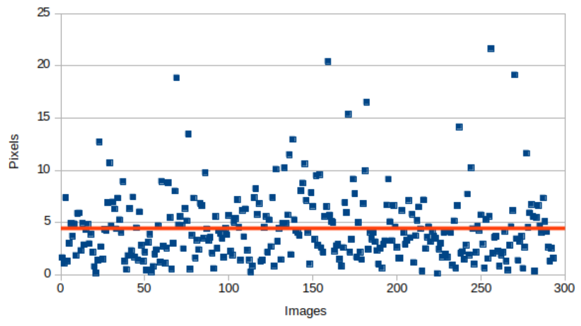
- **Use CNN as a fixed feature extractor**: Take a CNN pre-trained on a large dataset, then remove the last fully-connected and use the rest layers of CNN as a fixed extractor for the new dataset.



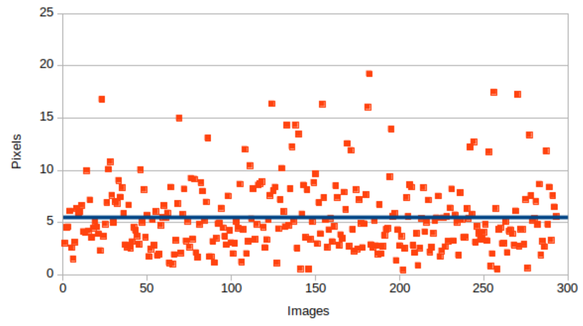
(a) . The 1st landmark (pronotum)



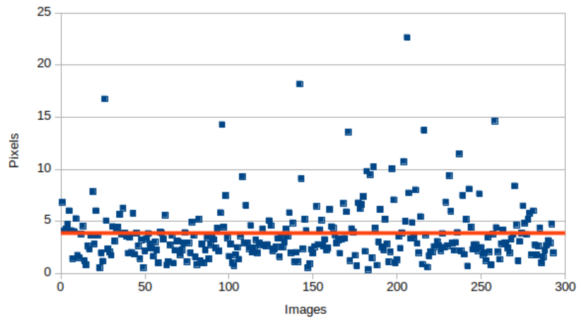
(b) . The 6th landmark (pronotum)



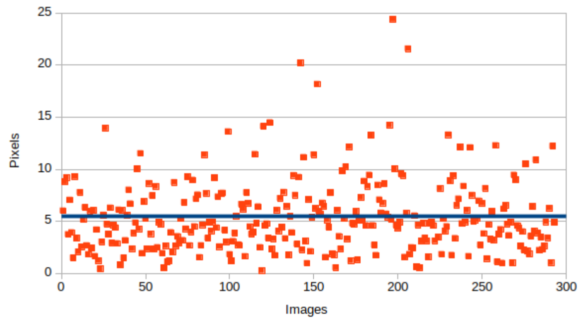
(c) . The 6th landmark (head)



(d) . The 1st landmark (head)



(e) . The 1st landmark (elytra)



(f) . The 8th landmark (elytra)

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

- **Fine-tuning a CNN**: This situation is the same as the first one. 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, 38] that have been trained on ImageNet [39]. These pre-trained models have been shared in deep learning community as a source to re-use the features of ImageNet dataset. Unfortunately, the features from ImageNet seem that do not relevant for our application because Image features mainly concern the detection of global shape of the objects whereas landmarks can be considered as local features [40]. Luckily, searching for landmarks is well defined in other application such as face recognition, or facial keypoints detection. Consequently, we have decided to continue with EB-Net: we have pre-trained EB-Net with a public facial keypoints dataset, then transferred the pre-trained parameters to fine-tune on beetle's images.

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

In recent years, several competitions have been published for predicting facial keypoints on human face, where they have publiced a dataset for training the models. As mention before, our problem has a revelant to these kind of competition. So, we have decided to choose a facial keypoints dataset to train EB-Net, and then to transfer the parameters to fine-tune on beetle's images.

The dataset that we have selected has been published for a challenge in the Kaggle website. It includes 2140 images of human faces with a size of 96 × 96. Each image contains 15 landmarks on the face: 6 landmarks for eyes, 4 landmarks for eyebrows, 4 landmarks for the mouth, and 1 landmark for nose tip. Figure 9 shows four face images in the dataset and the landmarks on each face.

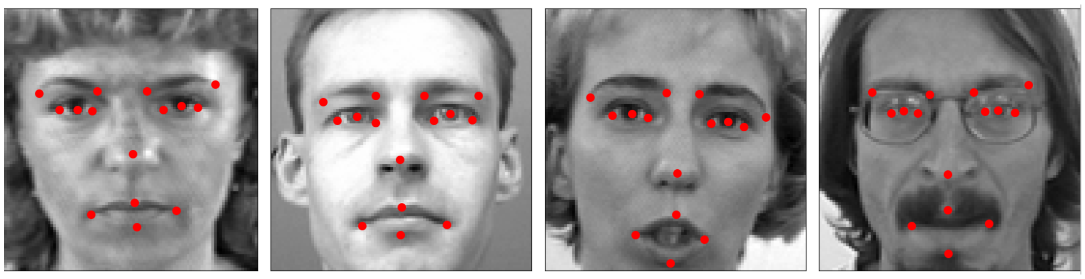


Figure 9: The samples (face and manual landmarks) in the dataset that we used to pre-train EB-Net.

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 requirements of the challenge. The new parameter values are 96 × 96 for the input size, and 30 for the number of output (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.

Team	Olegra 1st	Trump 2nd	Enes 3rd	Our
RMSE score (in pixels)	1.2824	1.4004	1.4026	**1.497**

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

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

3.3.2. Fine-tuning on beetle images

As we have mentioned, fine-tuning is a strategy of transfer learning that we 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 [37]).

3.3.3. 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

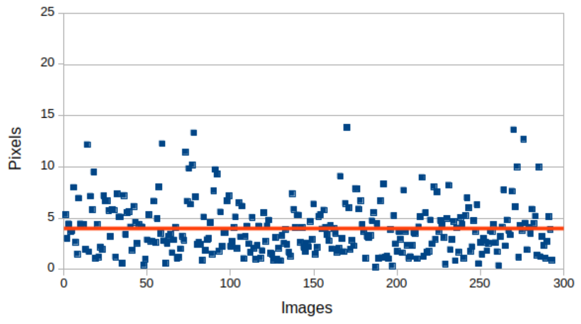
Landmark	1	2	3	4	5	6	7	8	9	10	11
From scratch	3.87	3.97	3.92	3.87	4.02	4.84	5.21	5.47	5.27	4.07	3.99
Fine-tune	3.21	3.28	3.2	3.22	3.31	4.21	4.54	4.76	4.55	3.39	3.29
% of impr.	17.04	17.34	18.36	16.61	17.66	13.13	12.82	12.96	13.69	16.68	17.54

Table 6: Average distances comparison on elytra images

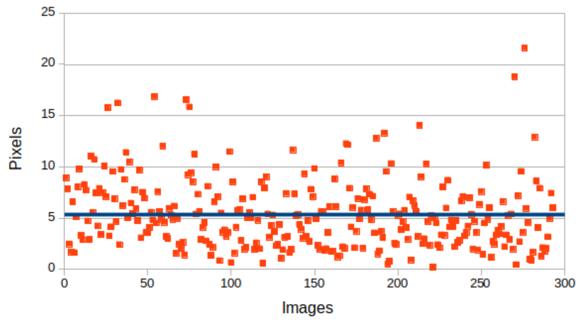
Considering on each landmark position, the improvement is different depending on the difficulty of its location. But, all cases have been improved even they are the best or the worst cases. With the help of fine-tuning, the predictions have gained from 36.85%/24.94%/18.36% to 19.18%/12.15%/12.82% on 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.

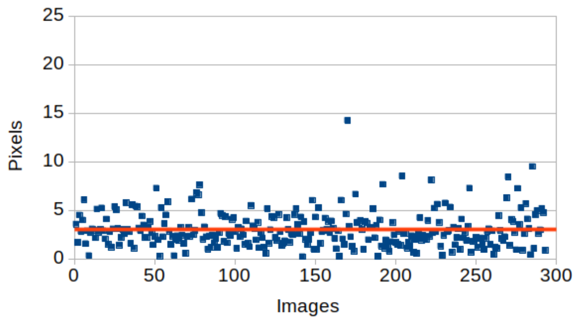
Figure 10, 11, 12 show the distribution of distances on two samples in each set of images: pronotum, head, and elytra, respectively. In these charts, the x-axis and y-axis present the number of images and the distance (in pixel). Each point in the chart represents a distance between manual and predicted landmarks. The blue/red lines in charts present the average values. We can observe from the figure 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 6th landmark), we do not see any point greater than 15 pixels with the fine-tuning process.



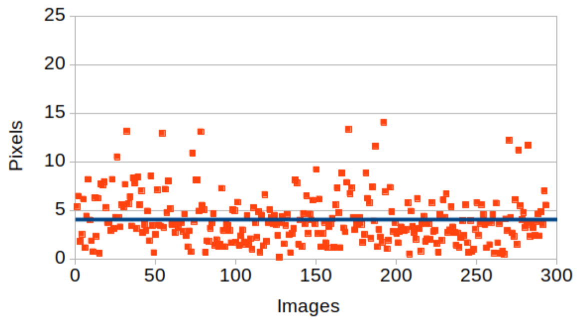
(a) . 1^{st} landmark (from scratch)



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



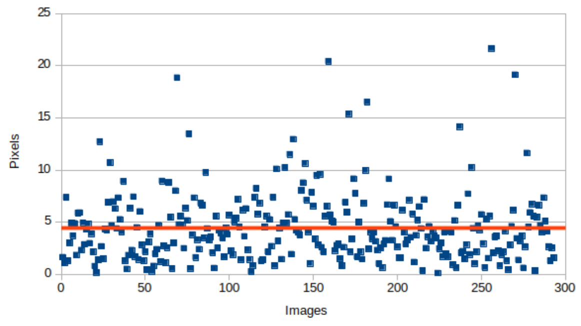
(c) . 1^{st} landmark (fine-tuning)



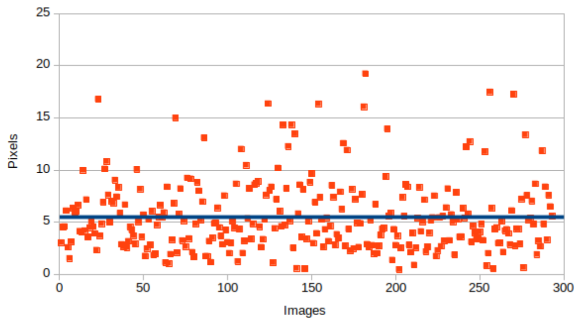
(d) . 6^{th} landmark (fine-tuning)

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

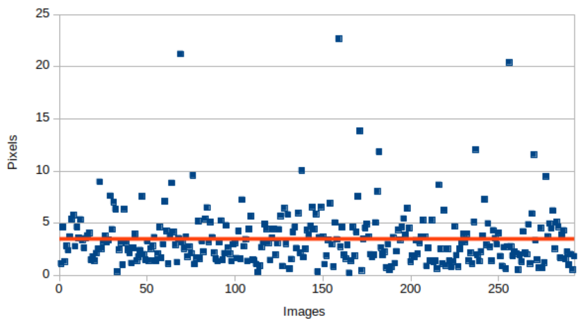
To illustrate the results, Figure 13 shows both predicted (in red) and manual (in yellow) landmarks on three random images from the three sets of images. Clearly, the predictions have been improved, they are more close to the manual ones. For example, we have obtained 7 well-predicted landmarks on the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



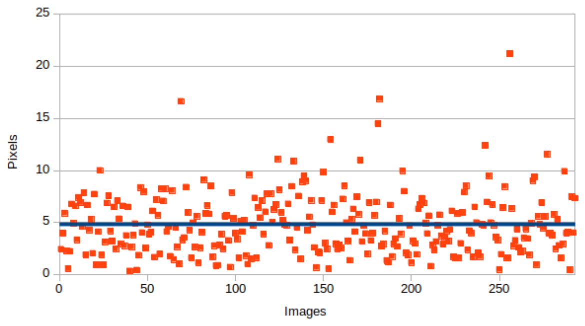
(a) . 6th landmark (from scratch)



(b) . 1st landmark (from scratch)

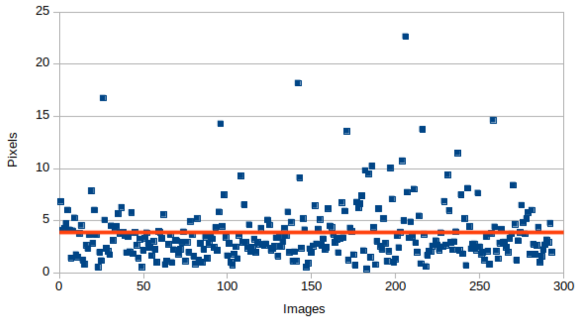


(c) . 6th landmark (fine-tuning)

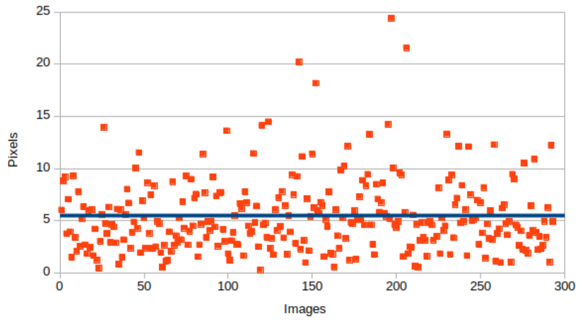


(d) . 1st landmark (fine-tuning)

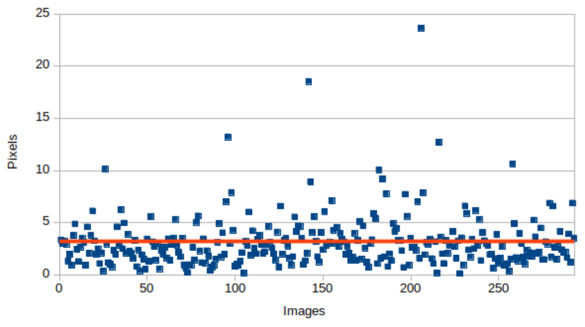
Figure 11: The distribution of distances of all head images on 1st landmark and 6th landmark.



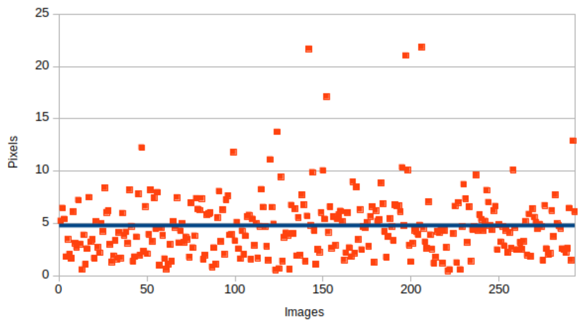
(a) . 1st landmark (from scratch)



(b) . 8th landmark (from scratch)

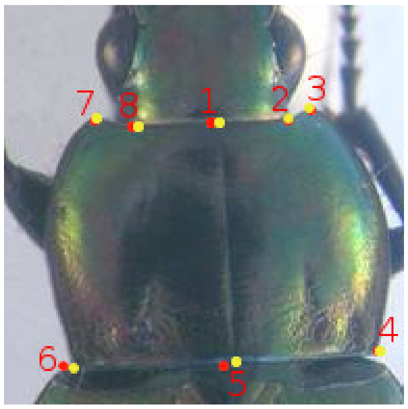


(c) . 1st landmark (fine-tuning)

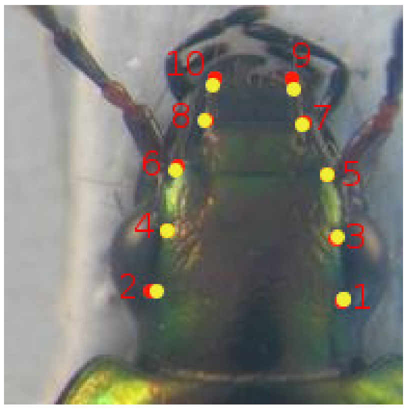


(d) . 8th landmark (fine-tuning)

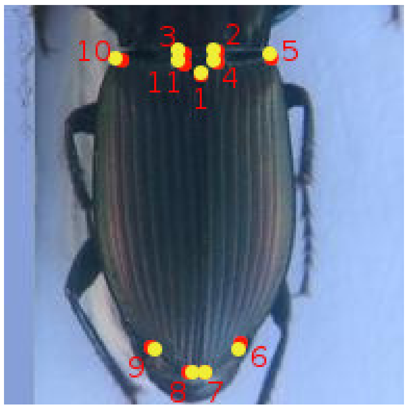
Figure 12: The distribution of distances of all elytra images on 1st landmark and 8th landmark.



(a) . Pronotum



(b) . Head

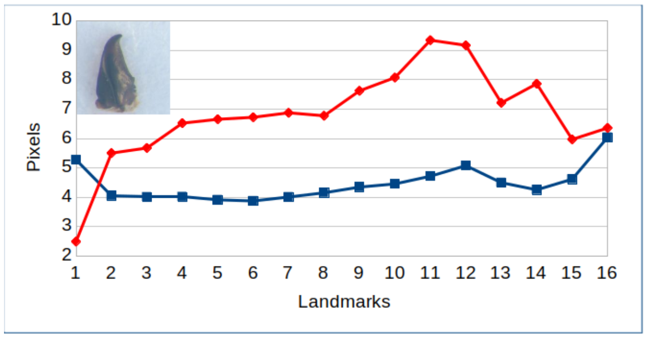


(c) . Elytra

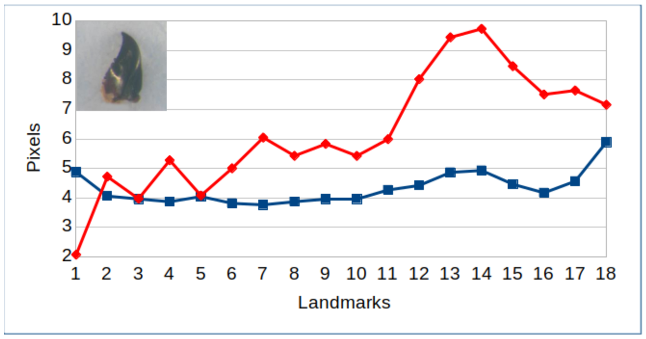
Figure 13: The location of predicted/manual landmarks in one case of each part. The red/yellow points represent the predicted/manual landmarks.

3.3.4. Results on mandibles

In the five sets of images, the mandibles are considered as easy to extract by using the image processing algorithms. In [28], 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 [28] 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 [28]. 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 paper, 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] 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.

[18] 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.

[19] 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.

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

[21] 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.

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

[23] 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.

[24] 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.

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

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

[28] 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`

[29] 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.

[30] 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.

[31] V.-L. Le, M. Beurton-Aimar, A. Zemmari, N. Parisey, Landmarks detection by applying deep networks, in: 2018 1st International Conference on Multimedia Analysis and Pattern Recognition (MAPR), IEEE, 2018, pp. 1–6.

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

[33] 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.

[34] 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.

[35] 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`

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

[37] 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.

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

[39] 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.

[40] 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.