

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}

^aUMR 1349 IGEPP, BP 35327, 35653 Le Rheu, France

^bUniversity of Bordeaux, 351, cours de la Libération, 33405 Talence

^cDalat 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. The network then trained and tested on a dataset includes five parts of beetle (head, elytra, pronotum, left and right mandibles). These works have also introduced another procedure to augment the dataset which can see a little bit small in our case. 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. 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 20th century [5], rooted in a strong statistical background, geometric morphometrics addresses the study of such biological shapes [6]. It is an effective set of methods with several specialised softwares readily available [7, 8]. Classical geometric morphometrics uses a set of landmarks to describe shape, a landmark being a two-dimensional anatomically-relevant point. In order to investigate the possibility of automated morphometric geometrics on beneficial insects, we chose to focus on one of the most common and ubiquitous beneficial insect of north-western France, *Poecilus cupreus* (Carabidae). It is considered a polyphagous

*Corresponding author

Email address: van-linh.le@labri.fr (Le Van Linh)

¹both authors contributed equally to this work.

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].

15

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 thought 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?], question answering [19] and language translation [20, 21]. 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 objects recognition [22?]. Recent success of this algorithm in human biometry [23] lead us to believe in its potential for insect morphometrics.

30 1.1. Related works

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. [24] have proposed SIFT method to find the corresponding keypoints between two images. Palaniswamy et al. [25] have proposed a method based on probabilistic Hough Transform to automatically locate the landmarks in digital images of *Drosophila* wings. In our work [26], we have proposed a method which have been 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 and it seems that good solutions for the images that can not segment. Yi Sun et al. [27] have proposed cascaded convolutional neural networks to predict the facial points of interest on the human face. Zhanpeng Zhang et al. [28] 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. In biology field, Cintas et al. [23] 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 this way, we have

applied CNN computing to work with pronotum landmarks.

50 1.2. Contributions

We present an approach to predict morphometric landmarks based on standardized digital pictures of a coleoptera anatomical parts. For each anatomical parts, we train a convolutional neural network and statistically assess the suitability of the predicted landmarks to replace manual landmarks in further geometric morphometric studies.

2. Material and Methods

55 In the context of 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. 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
65 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 .

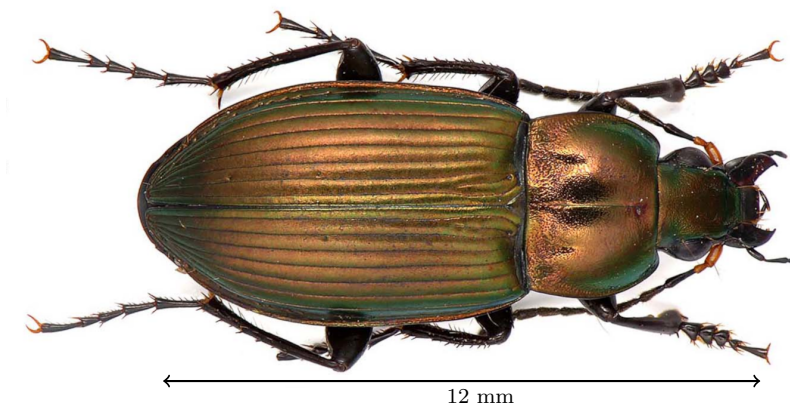


Figure 1: An illustration of the beetle.

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
70 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.

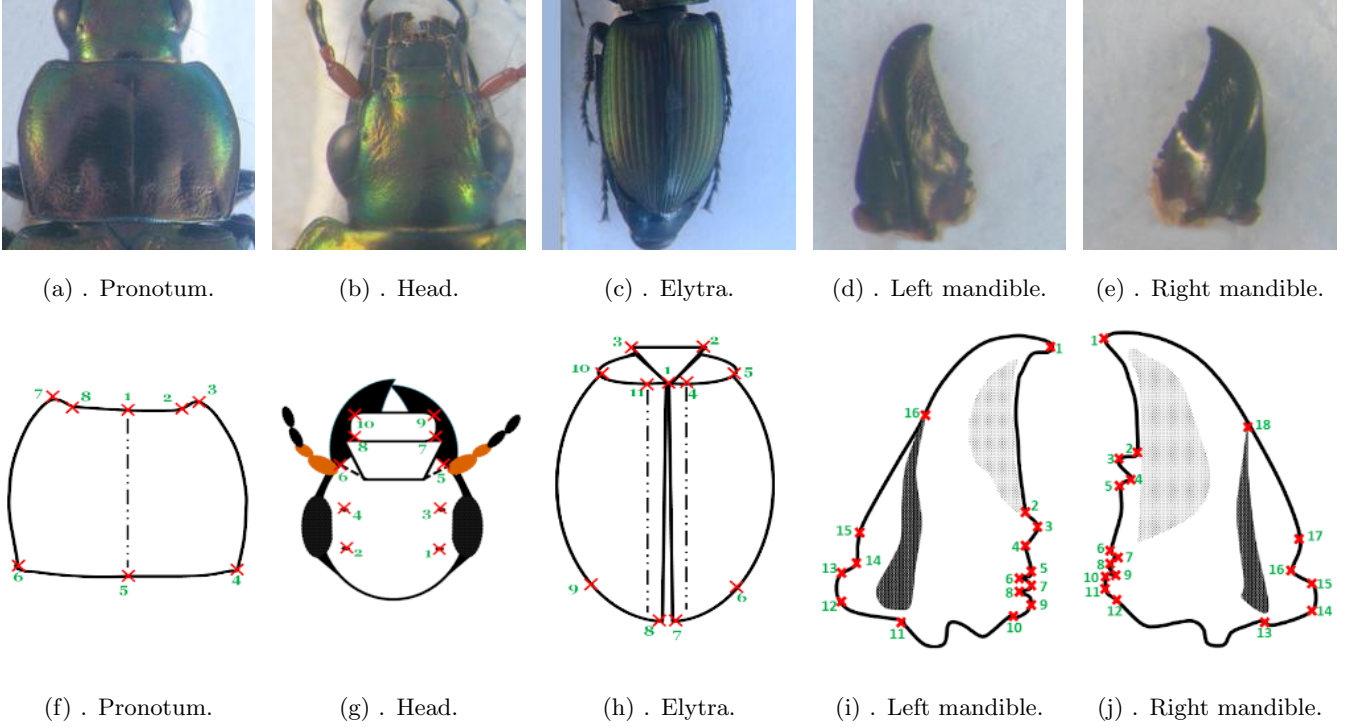


Figure 2: The sample images in our dataset (top row) and manual landmarks on each part defined by biologists (bottom row).

The success stories [] 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 []. However, these operations can be invariant in some cases []. 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 (more detail in Section X). Consequently, we have preferred different 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

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.

95 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 $293 \times 7 = 2051$ images for each set of images. Figure 3 illustrates the two strategies that we have described.

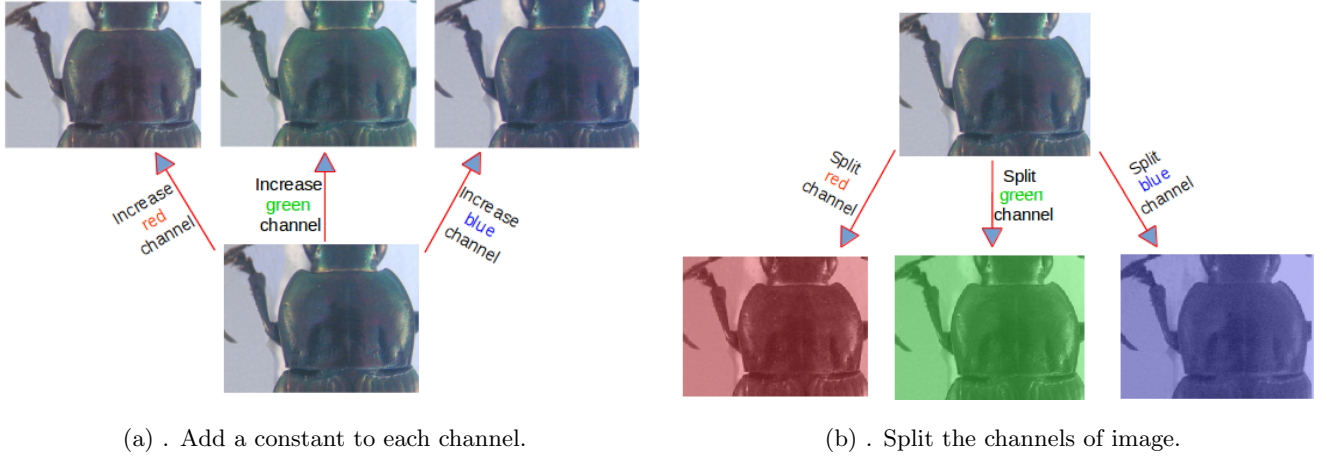


Figure 3: The two strategies to augment the number of images in our dataset.

To perform the objective, we have observed the input size of the several CNN models 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 ?? . 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 [29], the brightness of the image is normalized to $[0, 1]$, instead of $[0, 255]$ and the coordinates of the landmarks are normalized to $[-1, 1]$, instead of $[0, 256]$ and $[0, 192]$ before giving to the network.

2.2. Network architecture

From the first appearance to current, many CNN architectures have been available. Most often, they were designed to solve specific problems.

110 The first model has been inspired by AlexNet architecture. It received gray-scale image with the size of 256×192 as the input. The image is 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 3×3 , 2×2 , 2×2 for the kernel'sizes of CONV layers, respectively. Using POOL layers after the CONV layers is a common occurrence in CNN []. 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 with the size of 2×2 and 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 X.

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 [], but we have added them in the extracting features blocks to create some kinds of image noise augmentation. The objective of this work is to provide more studied 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 equivalent to train a set of thinner networks. These works have significantly reduced over-fitting and given more major improvements than other regularization methods [].

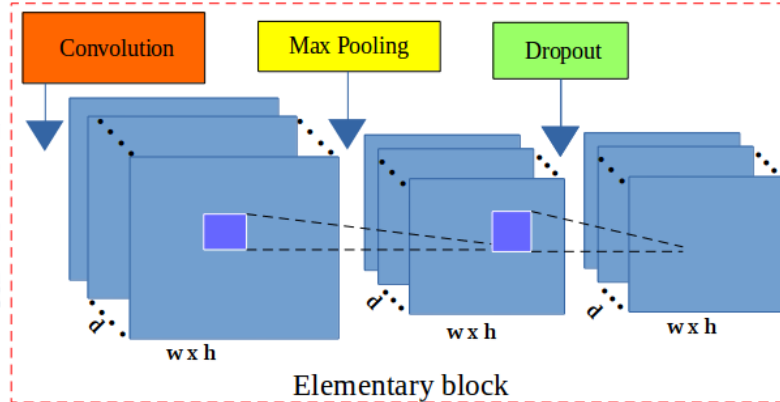


Figure 4: The components of an elementary block

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 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 [?] shows the structure of the EB-Net.

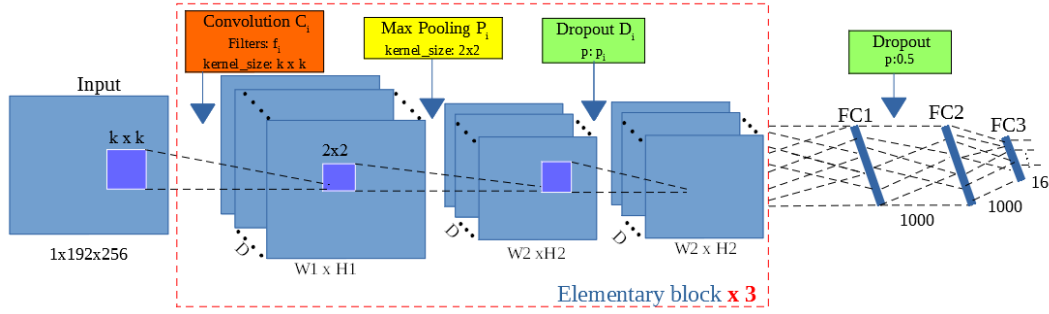


Figure 5: Elementary Blocks Network (EB-Net) architecture

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 for each layer), the optimized hyper-parameters are also important variables in designing 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 depend on the task and the dataset, they are determined empirically. In our application, we have chosen gradient descent as the optimized algorithm as usual studies [?]. 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 to use as the loss function because it is usually used for regression problems where outputs are not discrete 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 Y, 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 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.

170 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.

3. Results

3.1. The first test

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

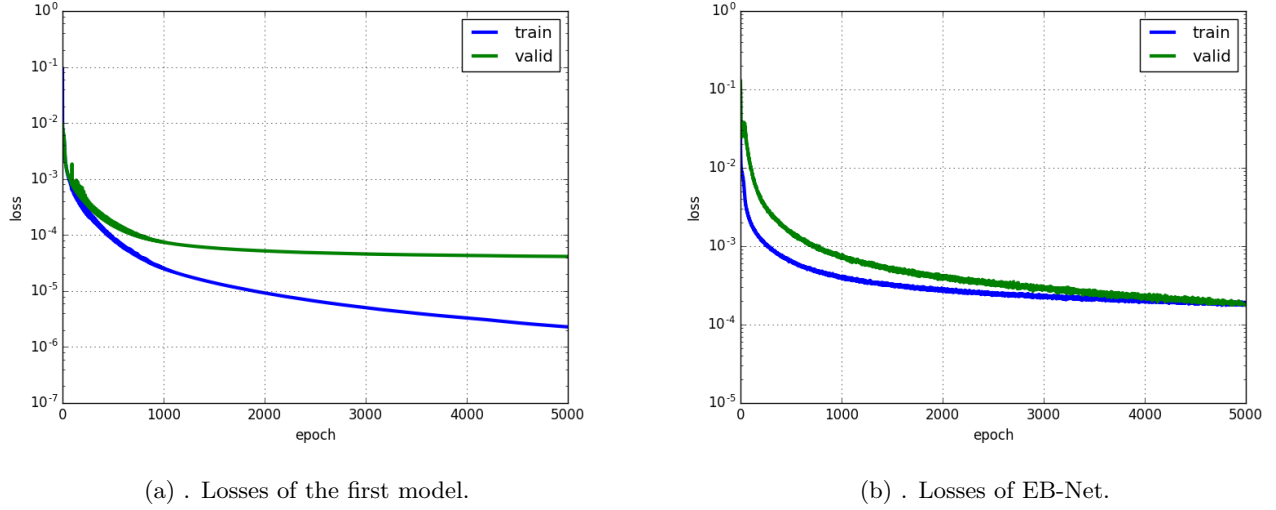


Figure 6: The losses of two models on pronotum images.

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 is able 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 first 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) are 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 detailed results from these processes. Firstly, we consider the losses during the training and validation processes. Secondly, we compare the coordinates of predicted landmarks and corresponding manual ones by calculating the distance between them. The average distance is also taken into account for each landmark position. We display also the landmarks on the image. Finally, 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.

195 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 step.

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

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

In the next step, we focus on the location of predicted landmarks by comparing these with the ground truths. 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 X shows the average distance on each position
200 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) could 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.

LM	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.

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
205 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 check the , 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 illustrate the average
210 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.

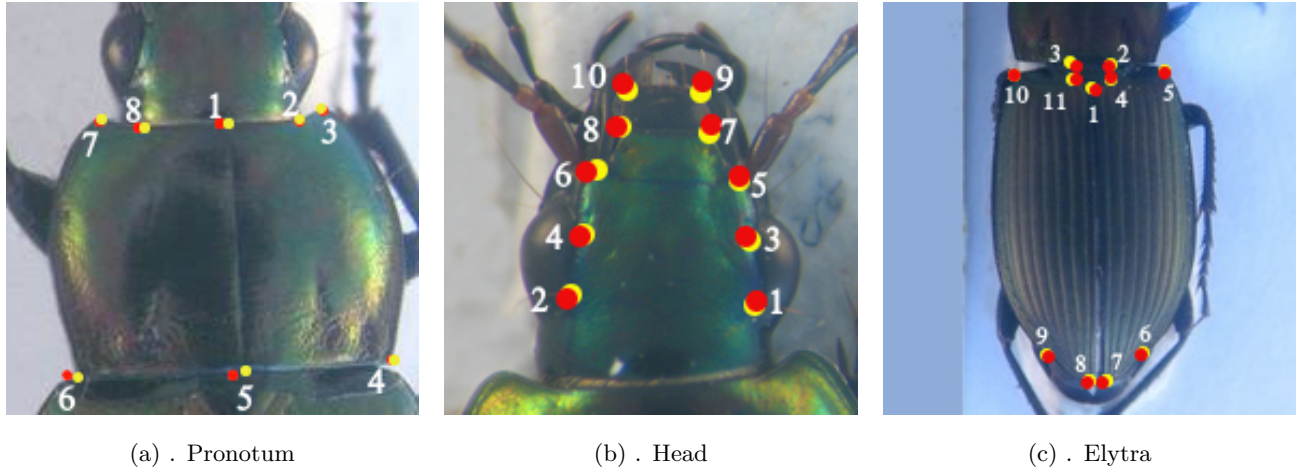


Figure 7: The landmarks on three parts of beetle. The red/ yellow points present the predicted/ manual landmarks.

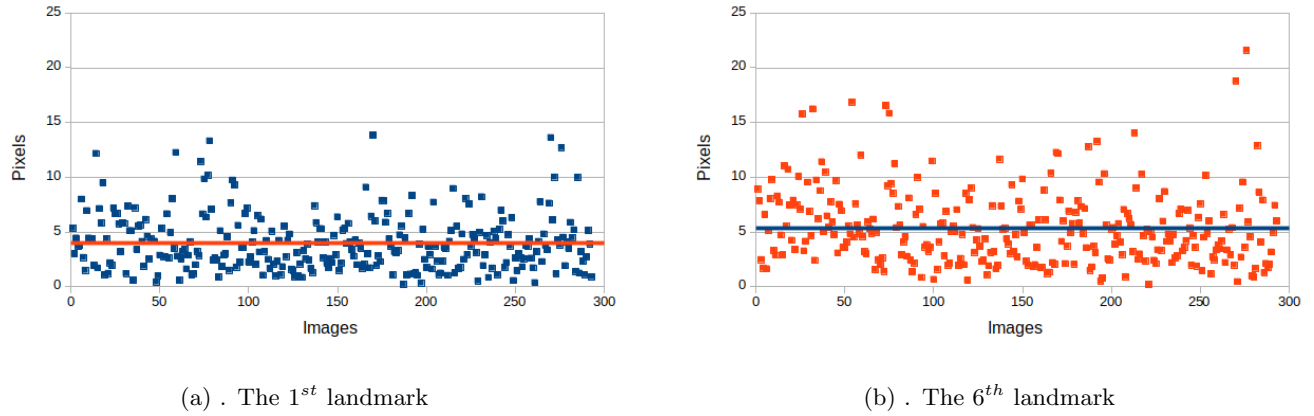


Figure 8: The distribution of distances between manual and corresponding landmarks of all pronotum images for the best (1st landmark) and the worst case (6th landmark)

The EB-Net has achieved good outcomes in most cases. However, it exists several difficult images in our dataset that the model can not recognize. That leads to the appearance of some high distance values in Figure X. The next step is to improve the predicted results.

215 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. As mentioned in section X, 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
220 between tasks could be an advice. We describe this process in this section to improve our predictions.

Transfer learning [] 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.
- **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 [] that have been trained on ImageNet []. 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 []. 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.

3.4. 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 X shows four face images in the dataset and the landmarks on each face.

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 1 st	Trump 2 nd	Enes 3 rd	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.

3.5. Fine-tuning on beetle images

One can note that the sizes of images between the two datasets are different, the beetle images have a size of 256×192 pixels; whereas the facial images are 96×96 . 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. We have also respected the form of images that they are square. 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 []).

3.6. Automated landmarks prediction for different anatomical parts

The dataset has been built by the biologists. It includes the images and manual landmarks. So, we can use the manual landmarks coordinates as ground truth to evaluate the coordinates of predicted landmarks. In the context of deep learning, landmark prediction can be seen as a regression problem. Therefore, the quality metric is used to evaluate the results. In particular, we use root mean square error (RMSE) to compute the accuracy of the implemented architecture.

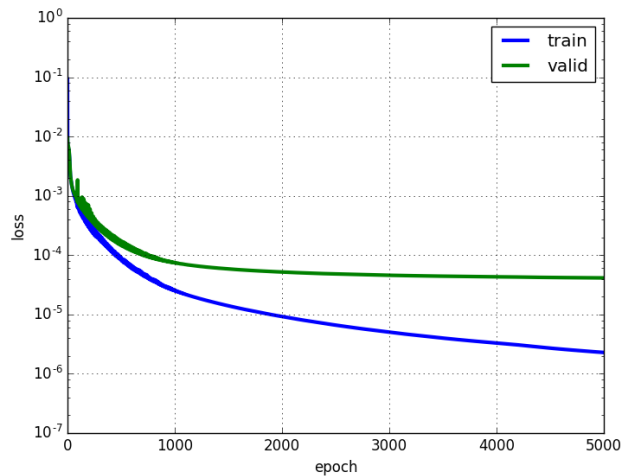


Figure 9: Learning curves of the first model.

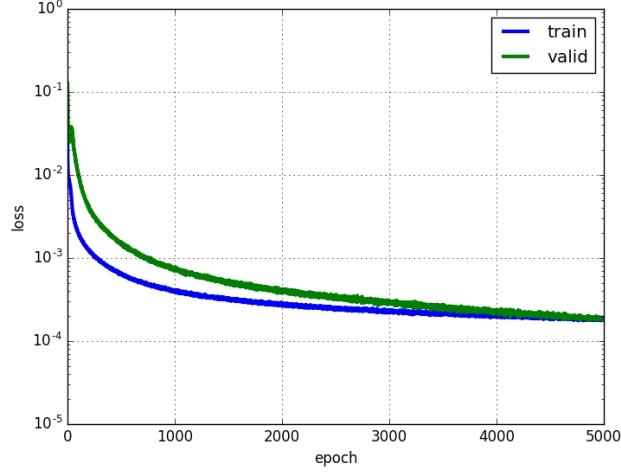


Figure 10: Learning curves of the last model.

Fig.9 and 10 show the training errors and the validation errors of a training time on the first and the third model, respectively. The blue curve presents RMSE on training set, the green curve presents the validation error. Clearly, the overfitting has appeared in the first model. In Fig.9, we can see that if the training is able to decrease with the number of epochs², it is not the case of validation loss. At the opposite in the third model, we can see some different values for the two losses at the beginning but after several epochs, these values become more proximate and the overfitting problem has been solved.

Fig.11 shows the predicted landmarks on test images set by the thrid model. When we consider the distance between the predicted and manual landmarks, the accuracy on coordinates of predicted landmarks on Fig.11a is 99%. The propotion on Fig.11b is 80%.

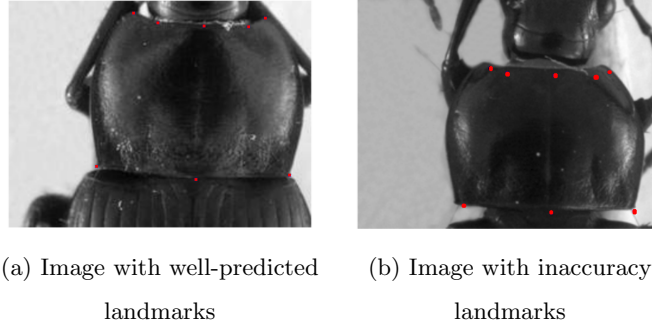


Figure 11: The predicted landmarks on an image in test set (red points)

Besides the losses of training, the distance from predicted landmarks to manual landmarks of the test images deserve attention also. Firstly, the distance between them is calculated. Then, the standard deviation [30] is used to quantify the dispersion of a set of distances. Table.4 shows the average error distance given on each landmark.

²An epoch is a single pass through the full training set.

Table 4: The average distance per landmark

#Landmark	Distance
1	4.002
2	4.4831
3	4.2959
4	4.3865
5	4.2925
6	5.3631
7	4.636
8	4.9363

Fig.12 shows the distribution of the distances on the first landmarks of all images. The accuracy based on the distance of each image can be separated into three spaces: the images have the distance less than average value (4 pixels): **56.66%**; the images have the distance from average value to 10 pixels: **40.27%**; and the images have the distance greater than 10 pixels: **3.07%**. The network has enabled to detect the landmark on pronotum automatically.

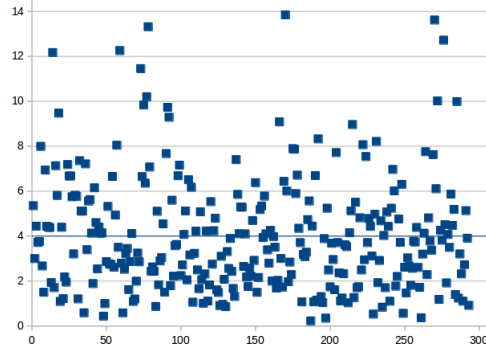


Figure 12: The distribution of the distances on the first landmark. The blue line is the average value of all distances.

Fig.13 shows the proportion of acceptable landmarks. In our case, a predicted landmark is acceptable if the distance between it and corresponding manual landmarks is less than the average distance plus a value of standard deviation. Most of the landmarks have been detected with the accuracy greater than 70%.

At the test phase, the trained network is used to predict the landmarks on a set of test images. The program outputs the predicted-landmarks of the images as TPS files; in additional, it also fills and displays the predicted-landmarks on sixteenth firstly images of test data. With the outputs are TPS files, the user can use MAELab [26] framework³ to display the landmarks on the images.

³MAELab is a free software written in C++. It can be directly and freely obtained by request at the authors.

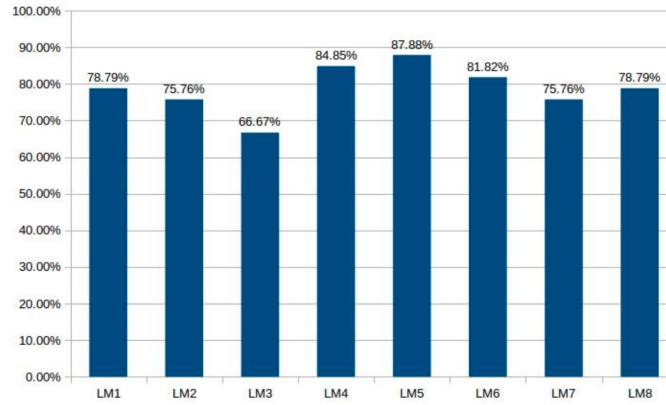


Figure 13: The proportion of acceptable predicted landmarks

4. Conclusion and future works

With beetle mandibles images, the object is easy to segment and we have succeeded to determine the landmarks automatically. In opposite, the pronotum images are difficult to segment. Methods which not suppose to be based on segmentation are necessary. In this paper, after testing several models, we have presented a convolutional neural network for automatic detection landmarks on the pronotum. 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. During the training phase, suitable techniques are used to prevent overfitting, a common issue of the neural networks. The network was trained several times in different selections of training data. After training with the manual landmarks given by the biologist, the network is able to predict the landmarks on the set of unseen images.

The results from the test set have been evaluated by calculating the distance between manual landmarks and corresponding predicted-landmarks. The average of distance errors on each landmark has been also considered. Using the convolutional network to predict the landmarks on biological images is promising good results in the case that the image can not be segmentation. The quality of prediction allows using automatic landmarking to replace manual landmarks in some aspects. In our case, the training dataset is limited. As a result, the accuracy of the network is acceptable. However, when we expect more about the accuracy of predicted landmarks (coordinates of predicted landmarks), the result of this work is still needed to improve (for example using a larger training dataset). Therefore, future research in landmarking identification appears as an improved of the worth exploring.

4.1. Evaluation metrics for further predicted landmarks usage

TODO

5. Discussion

TODO : a biological part, a computational part

- 325 [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)
330 (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>
- 335 [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)
340 451–470. 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>
- 345 [7] D. C. Adams, E. Otárola-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.
- 350 [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.
355 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 *Colophon haughtoni* Barnard, 1929, and *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. Černocký, 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] A. Bordes, S. Chopra, J. Weston, Question answering with subgraph embeddings, *arXiv preprint arXiv:1406.3676*.
- [20] 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.
- [21] S. Jean, K. Cho, R. Memisevic, Y. Bengio, On using very large target vocabulary for neural machine translation, *arXiv preprint arXiv:1412.2007*.
- [22] 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.
- [23] 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.
- [24] D. G. Lowe, Distinctive image features from scale-invariant keypoints, *International journal of computer vision* 60 (2) (2004) 91–110.

- [25] S. Palaniswamy, N. A. Thacker, C. P. Klingenberg, Automatic identification of landmarks in digital images, *IET Computer Vision* 4 (4) (2010) 247–260.
- [26] 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>
- [27] 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.
- [28] 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.
- [29] 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.
- [30] J. M. Bland, D. G. Altman, Statistics notes: measurement error, *Bmj* 313 (7059) (1996) 744.