ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 3, No. 3, XX-YY (2018)
www.astesj.com

ASTES Journal
ISSN: 2415-6698

Automatize landmarks setting on Species morphometry using Deep Neural Networks

Van-Linh Le[*,1,2], Marie Beurton-Aimar[1], Akka Zemmari[1], Nicolas Parisey[3]

[1]*LaBRI-CNRS 5800, University of Bordeaux, 33400, France*

[2]*ITDLU, Dalat University, 670000, Vietnam*

[3]*IGEPP, INRA-1349, 35653, France*

ARTICLEINFO

Article history:
Received:
Accepted:
Online:

Keywords:
Landmarks
Morphometry setting
Deep learning
Convolutional neural networks

ABSTRACT

Morphometry landmarks are known as one of the approaches to analyze the characteristics of organisms. Finding landmarks setting can give to biologists a comprehensive description of the organism. In this study, we propose a convolutional neural network (CNN) to predict the landmarks on biological's species. The network is designed as a combination of the "elementary blocks" including a convolutional layer, a maximum pooling layer, and a dropout layer. After training with a set of manually landmarks dataset, it has been used to predict the morphometric landmarks on biological images automatically. The network has been checked by applying two scenarios: training from scratch and fine-tuning. The predicted landmarks have been evaluated by comparing with the coordinates of manual landmarks which have been provided by the biologists. The network model is implemented by Python on Lasagne framework.

1 Introduction

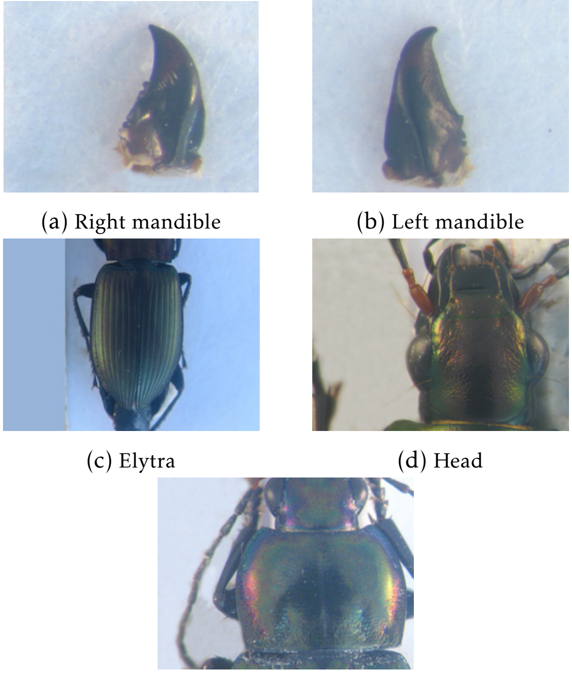
Morphometry analysis refers to measure the topography of an object, for example, its shape and its size. Biologists work with several parameters from organisms such as lengths, widths, masses, angles,... to analyze the interactions between environment and organisms development. Besides the traditional information, landmarks (or points of interest in the image) are known as one of the characteristics to analyze the shape. Instead of collecting all information, the shape is determined by a finite set of points, called landmarks. Landmarks store important information about the shape of the object, *for example*, the corners of the human mouth are a kind of landmarks. Mostly, the landmarks are along the outline of the object but in some special cases, it could be defined inside the anatomical part, *i.e* the landmarks on Drosophila wings are the intersection of veins on fly wings, but the landmarks on pronotum can be located at the shape edge or inside the pronotum. In our study, the morphometric landmarks are specific points defined by biologists. They are used in many biological studyings. Currently, the landmarks are set manually by the entomologist, the operation are time-consuming and difficult to reproduce when the operators change. Therefore, a method that gives automatic location of landmarks could have a lot of interest.

In this study, we have used a dataset including the images of collecting from 293 beetles in Brittany lands. All the images are presented in RGB color with two dimensions. For each beetle, the biologists took images of five parts: *left and right mandibles, head, body, and pronotum* (Fig.1). For each part, a set of manual landmarks has been positioned by an entomologist.

In the concept of automatically landmarks setting, image processing is usually the first choice to apply. This is a process that we apply a set of algorithms (in image processing) to extract and to analyse the object of interest. In which, segmentation is most often the first and the most important step. This task remains a bottleneck to compute features of an image. In some cases, the object of interest is easy to extract and can be analyzed with the help of a lot of very well-known image analysis procedures. Like previous study [**?**], we have analyzed two parts beetle mandibles (Fig.1a and Fig.1b). These parts are pretty easy to segment (enough good quality for our goals). In that work, we have applied a set of algorithms based on the combination of principal component analysis [**?**] and SIFT descriptor [**?**]. Unfortunately, this method is irrelevant with the case of the images that are not precise or diffi-

[*] Corresponding Author Name, Address, Contact No & Email

cult to segment, *i.e.* pronotum images. So, the remain question of how to predict the landmarks on the images like the pronotum images? This is the reason why we have turned to a way of analyzing images without need for a segmentation step. So, the next step has been to work with the prontoum images (Fig.1e). For each pronotum image, a set of 8 manual landmarks have been set by the biologists (Fig.**??**). They are considered as the ground truth to evaluate the predicted landmarks by our method.


(a) Right mandible

(b) Left mandible

(c) Elytra

(d) Head

(e) Pronotum/thorax

Figure 1: The anatomical parts of beetle

To achieve the landmarks prediction, this work introduces a method for this automatic detection of the landmarks on pronotum images. The main idea consists on design and train of a CNN [**?**] with a set of manual landmarks. In the first stage, the network has been trained from scratch on the dataset of pronotum images from the first model. In the second step, the training has been modified to improve the quality of prediction by including the fine-tuning[**?**] step. The network has been implemented by using Python on Lasagne library [**?**].

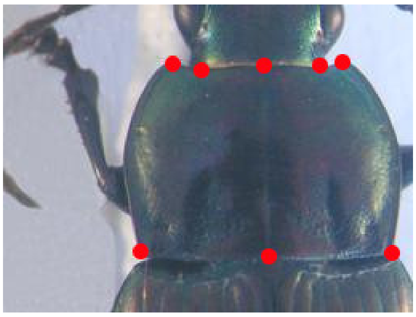


Figure 2: A pronotum image with its eight manual landmarks

The rests of the article is organized as follows. In the next sections, we first give a briefly overview of the related works on automatically landmarking. We then shortly present an overview about CNN. After that, in Section X1, we describe the architecture of the proposed network and its parameters also. The dataset augmentation processes are presented in Section X2. In Section X2, we give the first results of the model, then we present the step of fine-tuning to improve the result. Finally, we conclude the article with a discussion of future works in Section X3.

2 Related works

A landmark is a specific point that may contain useful information. For example, the tip of the nose or the corners of the mouth are landmarks on human face [**?**]. Under image processing point of view, when we want to extract the feature from the image, we can consider two kinds of cases: the object of interest can be segmented or not. Setting landmarks can not be achieved in the same way depending on which situation we are. When segmentation can be applied, Lowe et al. [**?**] have proposed SIFT method to find the corresponding keypoints in the 2D images. From the detected keypoints, the method is able to match two images. Palaniswamy et al. [**?**] have proposed a method based on probabilistic Hough Transform to automatically locate the landmarks in digital images of Drosophila wings. In previous work [**?**], 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, after several tests, we have had to conclude that this way does not provide good results with the pronotum images because the pronotum segmentation has too many noises.

In recent years, deep learning is known as a solution for many task in different topics. In image analysis domain, using deep learning, namely CNN, to determine the landmarks on 2D images has achieved better results even if the images that can not segment. Yi Sun et al. [**?**] have proposed cascaded CNNs to predict the facial points of interest on the human face. Zhanpeng Zhang et al. [**?**] proposed a *Tasks-Constrained Deep Convolutional Network* to optimize facial landmarks detection. Their model determines the facial landmarks with a set of related tasks such as head pose estimation, gender classification, age estimation, face recognition, or facial attribute inference. Cintas et al. [**?**] has introduced a network to predict the landmarks on human ears. After training, the network has the ability to predict 45 landmarks on human ears. In this way, we have applied CNN computing to work with pronotum landmarks.

3 Convolutional neural networks

Deep learning models are coming from the machine learning theory. They have been introduced in the middle of previous century for artificial intelligence applications but they encounter several problems to take real-world cases. Fortunately, the improvement of computing capacities both in memory size and computing time with GPU programming has opened the new perspective for deep learning.

Deep learning allows computational model composed of multiple processing layers to learn representations of data with multiple levels of abstraction [?]. Each layer extracts the representation of the input data from the previous layer and computes a new representation for the next layer. In the hierarchy of a model, higher layers of representation enlarge aspects of the input that is important for discrimination and suppress irrelevant variations. Each level of representations is corresponding to the different level of abstraction. During training, it uses gradient descent optimization method to update the learnable parameters via backpropagation. The development of deep learning opens promise results for well-known problems artificial intelligence on high dimensional data, therefore applicable to many domains: image recognition and classification [?, ?, ?], speech recognition [?, ?, ?], question answering [?], language translation [?] [?], and recognition [?][?].

A CNN consists of a number of connected layers. The layers of a CNN has neurons arranged in three dimensions: *width*, *height*, *and depth* with learnable parameters. Fig. 3 shows a classical example of CNN. It is a pipeline of usual layers: convolutional layers (CONV), pooling layers (POOLING), dropout layers (DROPOUT), and full-connected layers (FC).

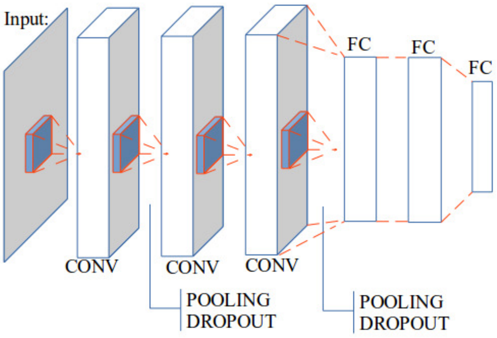


Figure 3: An example of usual convolutional neural network

A *convolutional* layer computes a dot product between its weights and a small region in the input. At the output, the results of connected local regions are combined. Convolution layer uses a set of learnable filters as parameters. Each filter is small spatially but extends the depth of the input. *Pooling* layer is used to down-sampling the input, to reduce the computational cost in remaining layers, and to control overfit. *Dropout* layer refers to dropping out units in the network. Dropping a unit out means temporarily removing it from the network, along with all its incoming and outgoing connections. *Full connected* layer refers to the output of the network. The number of outputs of the last full-connected layer are corresponding to the number of predicted values.

From the beginning of deep learning until now, many deep learning frameworks have been developed. These frameworks help the users to design their application by re-using already proposed network architectures. Almost frameworks are open source. According to the written programming languages, the frameworks can be separated into two main groups: C++, *such as Caffe*, *Deeplearning4j*, *Microsoft Cognitive Toolkit* and Python *i.e Keras*, *Theano*, *PyTorch*. Another framework exists using more confidential languages as *Lua*.

Theano [?] is an open source framework developed by the machine learning group at the University of *Montréal*. It is a Python library that allows to define, to optimize and to evaluate mathematical expressions relating multi-dimensional arrays efficiently by using a Numpy package. Theano supports compilation on either CPU or GPU architectures. Lasagne [?] is a lightweight library in Theano. It allows to build and to train the neural networks. In this work, we have used Lasagne to implement the proposed neural network. Recently, Theano has been stopped to develop but its community is still large. The networks which have been designed by Theano are still useful and efficient in deep learning area.

4 Application to landmarks identification

In the previous sections, we have presented an overview of automatic landmarks setting and CNN. In the first of this section, we describe the process to augment the dataset which is considered as a little bit small to apply deep learning. Then, we present the designing the processes of the network architecture that we use to predict the landmarks on pronotum images.

4.1 Data augmentation

The images come from a collection of 293 beetles from Brittany lands. All the images are taken with the same camera under same conditions with a 3264 × 2448 resolutions. For each specific part, a set of manual landmarks has been determined by biologists. The provided dataset contains 293 pronotum images, each image with 8 landmarks provided by biologists (Fig.**??**). The dataset was split into a training set with 260 images (training and validation) and a testing set of 33 images. During the training, the network learned the information through a pair of *image and manual landmarks* in the training set. At the testing stage, the image without landmarks is given to the trained network and the predicted landmarks coordinates will be given as output. Fig.**??** shows an example of pronotum image with its manual landmarks.

In some published networks [?][?][?], the maximum size of the inputs is not over 256 pixels. In our case, the resolution of the image is large, it becomes a difficulty for the computing. During training and testing, the images are down-sampling to a new resolution of 256×192. Obviously, the landmark coordinates of the image are also scaled to suit their new resolution.

In CNN, the network usually has a large number of learnable parameters. In addition, if the dataset is limited, the result, that we obtain, will have a large errors between the training and the testing processes. It means that the over-fitting had occured during the training process. In our case, the dataset is limited in 293 pronotum images. This number is very small in the context of deep learning. Therefore, we need to enlarge the size of the dataset. In image processing, we usually apply transform procedure (i.e rotation, translation,...) to generate a new image but the analysis of image by CNN is most often translation and rotation invariant. Therefore, we have applied two another procedures to increase the number of images in the dataset.

The first procedure has been applied to change the value of each channel in the original image. According to this, a constant is added to a channel of RGB image and for each time, we just change the value of one of three channels. For example, from an original RGB image, if we add a constant $c = 10$ to the red channel, we will obtain a new image with a different profile histogram. By this way, we can generate three new RGB images from one RGB image.

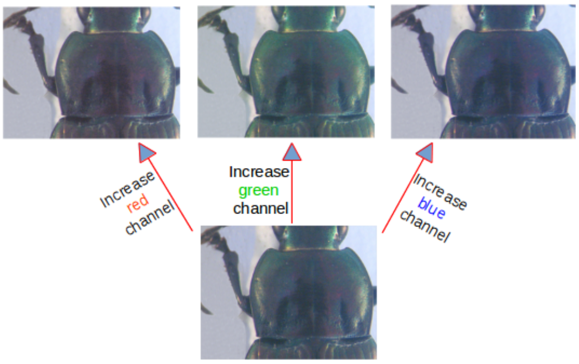


Figure 4: A constant (c=10) has been added to each channel of an original image for each time

The second procedure splits the channels of RGB images. It means that we separate the channels of RGB into three gray-scale images. At the end, we can generate six versions of original image, the total number of images used to train and validate is $260 \times 7 = 1820$ images (six versions and original image). The dataset that has been used for training and validation is split randomly by a ratio (training: 60%, validation: 40%) that has been set during the network setup.

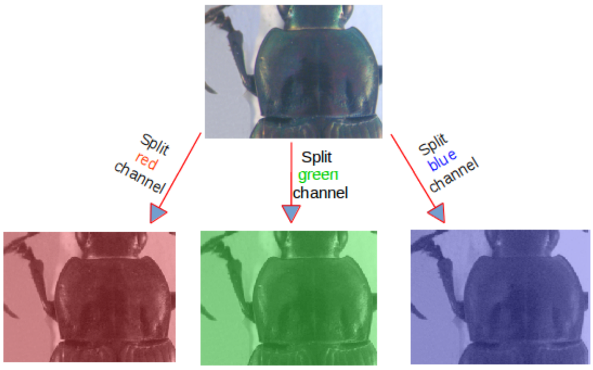


Figure 5: The individual channels have been extracted from an original image

In practical, when we work with CNN, convergence is usually faster if the average of each input variable over the training set is close to zero. Moreover, when the input is set closed with zero, it will be more suitable with the sigmoid activation function [?]. According to [?], 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 to be giving to the network.

4.2 Network architecture and training

Three different networks models have been proposed and trained to perform the best architecture for automatically landmarking predictions. They receive the same input of $1 \times 256 \times 192$ to train but they have different number of layers. In this section, we introduce the architectures of the networks and the process to improve the architecture from the beginning of designing.

Fig. 6 shows the architecture of the first model which is a very classical one. The network consists on three repeated-structure of a convolutional layer followed by a maximum pooling layer. The depth of convolutional layers increases from $32, 64$, and 128 with different sizes of the filter kernel: 3×3, 2×2, and 2×2. All the kernels of pooling layers have the same size of 2×2. The kernel sizes are classical as the literature. At the end, three full connected layers have been added to the network. The outputs of the full connected layers are $500, 500$, and 16, respectively. The output of the last full-connected layer corresponds to 8 landmarks (x and y coordinates) which we would like to predict. The training result shows that the architecture of this model provides overfitting and so is not good enough to solve the problem.

The second network has the same architecture as the first one. But, number of outputs at full-connected layers have been modified from 500 to 1000 to prevent the overfitting, but the result did not lead to better performances.

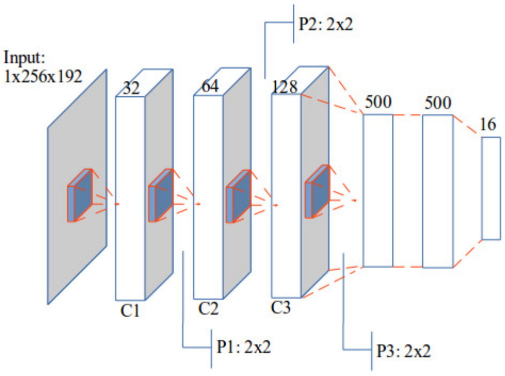


Figure 6: The architecture of the first network

In the third network, instead of changing the parameters, we have added 4 dropout layers to the network. This is considered as the good solution to prevent the overfitting. The idea of dropout is to randomly drop units from the neural network during training. It prevents units from co-adapting too much. During training, dropout samples are done from an exponential number of different "thinned" network. At test times, it is easy to approximate the effect of averaging the prediction of all thinned networks by simply using a single unthinned network with smaller weights. This significantly reduces overfitting and gives major improvements over other regularization methods [**?**]. Fig.7 shows the architecture of the third network. The first three dropout layers are supplemented to the repeated-structures followed the maximum pooling layers. In that way, structure becomes a convolution layer with square filter, followed by a *maximum* pooling and dropout layer, called *"elementary blocks"*. The probability values used for dropout layers are 0.1, 0.2, and 0.3. Actually, we keep the same value for the parameters of the convolutional (32, 64, and 128), pooling (3×3, 2×2, and 2×2) and full-connected layers (1000, 1000, and 16) as the second one.

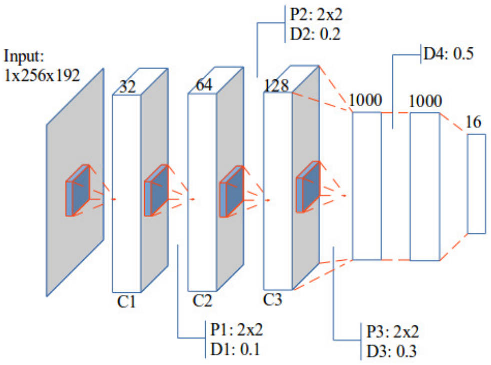


Figure 7: The architecture of the last network

The remaining dropout layer is inserted between the first two full connected layers. The probability value of this layer is set to 0.5. The output layer still contains 16 units corresponding to the coordinates of 8 predicted landmarks.

To obtain the predicted landmarks on all images, we have applied cross-validation technique to select the training data in the training process. The model has been trained with 1,820 image in 5000 epochs. During the training, the network is designed with a small sharing learning rate and a momentum. As usual, these parameters have been used to perform gradient descent during backward phase to update the parameters of the layers. The value of learning rate and momentum are updated over training time to fit with the remaining number of iterations: the momentum value has been adjusted in a range of 0.9 → 0.9999 and the learning rate value has been adjusted from 0.03 to 0.00001. The implementation of this architecture used Python on Lasagne framework [**?**] which allows to train the network on GPU. The training process took around 3 hours using NVIDIA TITAN X cards. The design of the network is available on GitHub[1].

4.3 First results

Usually, the first performance of a CNN is appreciated from the loss values in training and validation steps. In the context of deep learning, landmark prediction can be seen as a regression problem. Therefore, to evaluate the results, we have used root mean square error (RMSE) to compute the accuracy of the implemented architecture.

Fig.8 and 9 show the training errors and the validation errors of a training time on the first and the third model, respectively. The blue curves present RMSE on training dataset, the green curves present the validation errors. Clearly, the overfitting has appeared in the first model. In Fig.8, we can see that if the training is able to decrease with the number of epochs[2], 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.

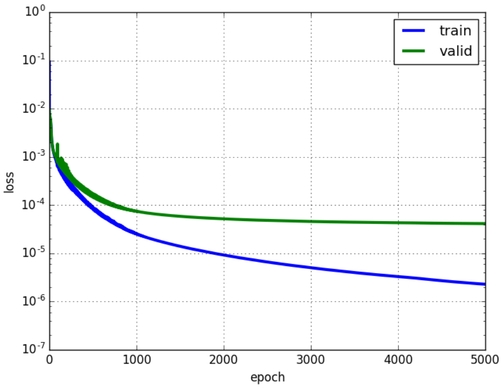


Figure 8: Learning curves of the first model.

[1] It is freely obtained by request the authors.

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

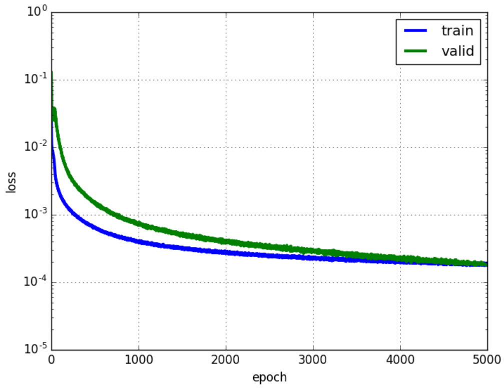


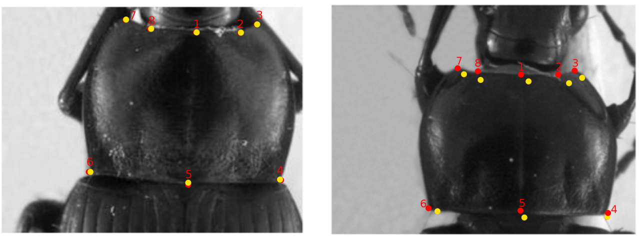
Figure 9: Learning curves of the last model.

To have the correct about the predicted landmarks, we would like to evaluate the results in other views by using the manual landmarks coordinates (which have provided by the biologists) as ground truth. Firstly, we have considered the results in a statical point of view. We have computed the quality metric to measure the linear cofficient between the coordinates of predicted and manual landmarks. The scores have been calculated based on 3 quality metrics: cofficient of determination (r^2), explained varicance (EV), and Pearson correlation (using *scikit-learn* [**?**]) like Cintas et al. [] have done. Table.**??** shows the comparison between our and their scores. From the scores, we can see that the quality of predicted coordinates are very precise.

Metric	**r**^{2}	**EV**	**Pearson**
Cintas et al	0.884	0.951	0.976
Our score	**0.9952**	**0.9951**	**0.9974**

Table 1: The cofficient on the quality metrics

Then, we have considered the results on the point view of image processing. Fig.**??** display the landmarks on the images. The red points are manual landmarks and the yellow points are predicted landmarks. The landmarks in Fig.**??** are a well prediction, while they are less accurated in the case of Fig.**??**.



(a) Image with well-predicted landmarks

(b) Image with inaccuracy landmarks

Figure 10: The predicted landmarks on an image in test set (yellow points)

To have a correct assessment of predicted coordinates, we have calculated the distances between the predicted and corresponding manual landmarks in all images. Then, we have computed the average distance per landmark as presented in Table.**??**. The minimum distance is around 4 pixels in the 1^{st} landmark, while the worst distance is more than 5 pixels in the 6^{th} landmark. The other distances are approximate 4.5 pixels.

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

Table 2: The average distance per landmark

In other view, we have computed the standard deviation [**?**] which is used to quantify the dispersion of a set of distances. For example, Fig.11 shows the distribution of the distances on the first landmark of all images. The accuracy based on the distance in 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 6.5 pixels (average distance plus standard deviation): *yyyyy*%; and the images have the distance greater than 6.5 pixels: *zzzz*%.

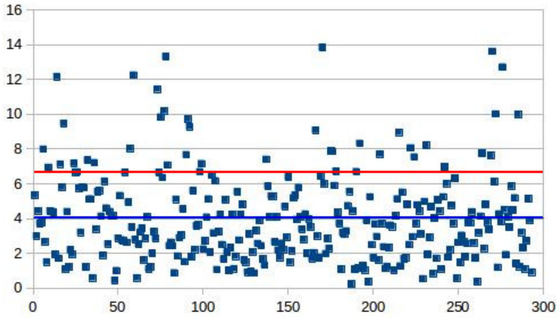


Figure 11: The distribution of the distances on the first landmark. The blue line is the average value, the red line is standard deviation value.

Fig.12 shows the proportion of acceptable landmarks, i.e a predicted landmark is acceptable if the distance between it and the corresponding manual one is less than the average distance plus the standard deviation value. Most of the landmarks have been detected with an accuracy greater than 75%.

In the image processing point of view, the distance in 4 pixels (1^{st} landmark) is still high, it could be considered as not enough precise. Morever, when we consider the distribution of the distance in the 1^{st} landmark, the percent rate of the distance less than average value is not enough good. It means that we have less the predicted landmarks closed to the manual ones as we expect (Fig.**??**). So, the next step has been decided to improve these results.

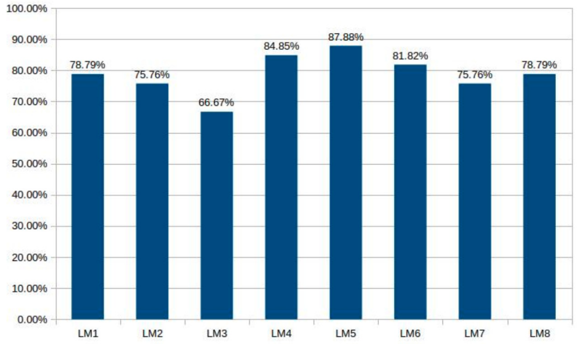


Figure 12: The proportion of acceptable predicted landmarks

5 Result improvement by using fine-tuning

As we have discussed in previous sections, the proposed network has been trained from scratch on the pronotum dataset. Even if the strengh of the correlation seems to validate the results, when the predicted landmarks are displayed on the image, the results are not enough precise since the average error is still high (≥ 4 pixels).

In order to reach more acceptable results for biologists, we have broadened the model with a step of transfer learning. That is a method that re-uses model developed for a specific task/dataset to lead another task (called *target task*) with another dataset. This allows rapid progress and improves the performance of the model on the target task [**?**]. The most popular example has been given with the project ImageNet of Google [**?**] which has labeled several millions of images. The obtained parameter values which can be used in another context to classify another dataset, eventually very different dataset [**?**]. The name of this procedure to re-use parameters to pre-train a model is currently called *fine-tuning*.

Fine-tuning does not only replace and retrain the model on the new dataset but also fine-tunes the weights of a trained model by continuing the backpropagation. Unfortunately, some rapid tests have shown that re-using ImageNet features has not been relevant for our application. We have designed a way to reproduce the method with our own data. It is worth noting that of course the size of data to pre-train has drastically decreased. For our pre-training step, the network has been trained on the whole dataset including the images of three parts of beetle i.e pronotum, body and head. Then, the trained model will be used to fine-tune and test on pronotum set.

5.1 Training data preparation

As we have mentioned, the training dataset includes a combination of the images from three sets: pronotum, body, and head (Fig.13). For each set, 260 original images have been chosen randomly for training and validation. By applying the same procedure in section **??**, the training dataset was enlarged to 5,460 images ($260 \times 7 \times 3$). However, the number of manual landmarks on each part is different: 8 *landmarks on pronotum part*, 11 *landmarks on body part, and* 10 *landmarks on head part* (Fig.13). The manual landmarks have a specific meaning for the biologists. So, we cannot insert the landmarks arbitrary. Instead of to do that, we keep the smallest number of landmarks among the three parts and remove the supernumerary when it is needed. Specifically, we have removed three landmarks on the body part $(1^{st};6^th;9^{th})$ and two landmarks on the head part $(5th;6^{th})$ (Fig.13).

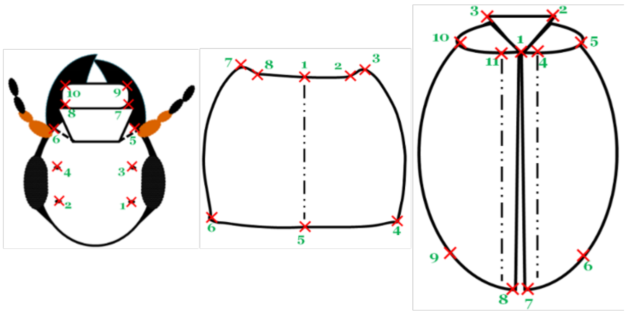


Figure 13: A presentation of head, pronotum and body part with corresponding manual landmarks

5.2 Using fine-tuning for pronotum dataset

At the first step, the network is trained with 5,460 images following the same way than explained in Section **??**. After that, this trained model is used to fine-tune the pronotum dataset. To compare the result with the previous one, the trained model has been fine-tuned on pronotum images with different cross-validation (as described in section **??**). The losses during fine-tuning are shown in Table.**??**. Comparing with the losses when we trained the model from scratch (Table.**??**), the validation losses of this scenario have been significantly improved (around 40%).

6 Prepare Your Paper before Styling

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar.

6.1 Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Do not use abbreviations in the title or heads unless they are unavoidable.

1. Use SI (MKS) as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
2. Do not mix complete spellings and abbreviations of units: “Wb/m2” or “webers per square meter”, not “webers/m2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.
3. Use a zero before decimal points: “0.25”, not “.25”. Use “cm3”, not “cc”. (*bullet list*)

6.2 Equations

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multi-leveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled. Number equations consecutively. Equation numbers, within parentheses, are to position flush right, as in (1), using a right tab stop. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize all the symbols for quantities and variables, Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in

$$\alpha+\beta=\gamma \tag{1}$$

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not “Eq. (1)” or “equation (1)”, except at the beginning of a sentence: “Equation (1) is . . .”

6.3 Figures

Figure 14: ASTESJ logo

Figure 15: ASTESJ logo

6.4 Tables

a	aa	sd
a	aa	sd
a	aa	sd

Table 3: Summary of datasets used

6.5 Units

6.5.1 Some Common Mistakes

1. The word “data” is plural, not singular.
2. The subscript for the permeability of vacuum μ_0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter “o”.
3. A graph within a graph is an “inset”, not an “insert”. The word alternatively is preferred to the word “alternately” (unless you really mean something that alternates).
4. Do not use the word “essentially” to mean “approximately” or “effectively”.
5. In your paper title, if the words “that uses” can accurately replace the word “using”, capitalize the “u”; if not, keep using lower-cased.
6. Be aware of the different meanings of the homophones “affect” and “effect”, “complement” and “compliment”, “discreet” and “discrete”, “principal” and “principle”.
7. Do not confuse “imply” and “infer”.
8. The prefix “non” is not a word; it should be joined to the word it modifies, usually without a hyphen.
9. There is no period after the “et” in the Latin abbreviation “et al.”.
10. The abbreviation “i.e.” means “that is”, and the abbreviation “e.g.” means “for example”.

7 Using the Template

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

7.1 Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named

7.2 Heading

8 Tables and Figures

All illustrations (photographs, drawings, graphs, etc.), not including tables, must be labelled "Figure." Figures must be submitted in the manuscript. All tables and figures must have a caption and/or legend and be numbered (e.g., Table 1, Figure 2), unless there is only one table or figure, in which case it should be labelled "Table" or "Figure" with no numbering. Captions must be written in sentence case (e.g., Macroscopic appearance of the samples.). The font used in the figures should be Times New Roman, normal, **size 8**. If symbols such as ×, η, or ν are used, they should be added using the Symbols menu of Word.

All tables and figures must be numbered consecutively as they are referred to in the text. Please refer to tables and figures with capitalization and unabbreviated (e.g., "As shown in Figure 2...", and not "Fig. 2" or "figure 2"). The tables and figures themselves should be given in the running text.

The resolution of images should not be less than 118 pixels/cm when width is set to 16 cm. Images must be scanned at 1200 dpi resolution and submitted in jpeg or tiff format. Graphs and diagrams must be drawn with a line weight between 0.5 and 1 point. Graphs and diagrams with a line weight of less than 0.5 point or more than 1 point are not accepted. Scanned or photocopied graphs and diagrams are not accepted.

Tables and figures, including caption, title, column heads, and footnotes, must not exceed 16 × 20 cm and should be no smaller than 8 cm in width. Please do not duplicate information that is already presented in the figures.

Tables and Figures can be single or double column. For double column use section breaks.

Conflict of Interest The authors declare no conflict of interest.

Acknowledgment Time New Roman, 10 Normal. Acknowledge your institute/ funder.

References Citations in the text should be identified by numbers in square brackets. The list of references at the end of the paper should be given in order of their first appearance in the text. All authors should be included in reference lists unless there are 10 or more, in which case only the first 10 should be given, followed by 'et al.'. Do not use individual sets of square brackets for citation numbers that appear together, e.g., [2,3, 5–9], not [2], [3], [5]–[9]. Do not include personal communications, unpublished data, websites, or other unpublished materials as references, although such material may be inserted (in parentheses) in the text. In the case of publications in languages other than English, the published English title should be provided if one exists, with an annotation such as "(article in Chinese with an abstract in English)". If the publication was not published with an English title, cite the original title only; do not provide a self-translation. Font size of references are Time New Roman, normal, **size 8**. Capitalize only the first word in a paper title, except for proper nouns and element symbols. References should be formatted as follows (please note the punctuation and capitalization):

Note that you should include DOI of correspondence reference at the end. No need to categorize the references into journal, conference and thesis headings. References should be cited in text in ascending order.

Journal articles: Journal titles should be abbreviated according to ISI Web of Science abbreviations. The example of referencing journal paper is [**?**].

References

90