Deep network for landmarks prediction on Beetle pronotums

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

[1]LaBRI - CNRS 5800 Bordeaux University, France, van-linh.le/beurton/zemmari@labri.fr
[2]IGEPP - INRA 1349, France, nparisey@rennes.inra.fr
[3]ITDLU - Dalat University, Vietnam, linhlv@dlu.edu.vn

Keywords: Landmarks, convolutional neural networks, finetuning, recogntion, procrustes.

Abstract

Morphometry landmarks were used in many biological applications. Mostly, the landmarks are defined manually or semiautomatic by applying the image processing techniques. In recent years, deep learning is known as a good solution for the difficult problems in computer vision. It appears in many fields such as classification, recognition, face detection. In the context applying the deep learning to solve the recognition problems, in this paper, we propose a scenario to predict the landmarks on 2D images, specify beetle's head images. The proposed method includes two stages: firstly, the landmarks are estimated by applying convolutional neural network; then, the estimated landmarks are improved to increase the accuracy. The method experimented on a set of 293 images. The accuracy of the method is evaluated by calculating the distance in pixels between the coordinates of the predicted landmarks and manual landmarks which were provided by the biologists.

1 Introduction

Morphometry landmark (or point of interest) is an important feature in many biological investigations. It was usually used to analyze the forms of whole biological organs or organisms. The analysis is mainly based on the coordinates of the landmarks. The collecting of enough the number of landmarks can help the biologists make a good estimate about organisms. Depending on the problem, the number of landmarks may be more or less; besides, the location of landmarks can be located on the shape (border) or inside the object, *for examples,* the landmarks on Drosophila wings have stayed on the veins of the wings but the landmarks on human ear can be located at the ear hole or inside. Recently, the landmarks were set manually by the biologists. This work is time-consuming and difficult to reproduce. Therefore, a method that proposes automatically the coordinates of landmarks could be a concern.

Based on the characteristics of digital images acquired for morphological studies, the images can be divided into two groups: the images where they are easy to segment the objects of interest, called *segment-able images*; and the images that we can go in tight when segment the objects, called *un-segment-able images*. For that reason, the methods that used to identify the landmarks automatically may be divided into two groups too. For segment-able images, identification of landmarks on the shape can be finished by applying the image processing techniques such as HOG[1], SIFT[2], ... But for un-segment-able images, defining the landmarks become a challenge and the image processing techniques seem to be inappropriate. This article introduces a scenario for automatic detection of the landmarks on biological images, specific beetle's pronotum images (Fig. 1). The method includes 2 stages: 1) The initially predicted landmarks are given by a convolutional neural network (CNN) [3]. The main idea of this stage is design and train a CNN with a set of images and their manual landmarks. The dataset includes 293 pronotum images and their manual landmarks which have been provided by the biologists. The images are presented in two dimensions and RGB color. After training, the trained network will be able to detect the initially predicted landmarks on the pronotum images; 2) The predicted landmarks which located in the shape of pronotum will be refined the location to increase the accuracy of coordinates. This stage is done by applying a Procrustes analysis[4]. For each manual landmark, a model is generated as a specific. Then, it is used to refine the corresponding predicted landmarks.

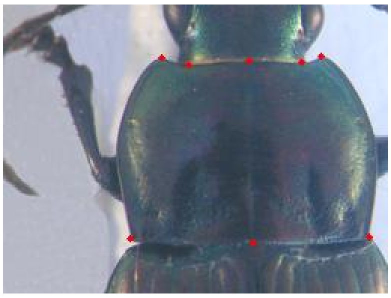


Figure 1: An example of pronotum images and its manual landmarks

In the next section, we present related works in domain automatically estimation landmarks on 2D images. In section 3, we present an overview about the stage that predict the initial automatically landmarks by applying CNN. It shows also the training process and experiment stage. The procedure that apply to improve the coordinates of predicted landmarks by CNN will be presented in section 4. In the last section, we have some conclusion based on the experiments and analysing the results.

2 Related works

Landmarks or points of interest are one of the important characteristics in geometric morphometrics. Landmark studies have traditionally analyzed on 2D images. Depending on which situation was stayed (segment-able or un-segment-able images), setting landmarks must apply the different methods.

When segmentation can be applied, Lowe et al. [2] have proposed a method to identify the key points in the 2D image. From the detected key points, the method is able to match two images. Palaniswamy et al. [1] have applied probabilistic Hough Transform to automatically estimate the landmarks in images of Drosophila wings. Adrien et al. [5] have extended Palaniswamy's method to detect landmarks automatically on beetles mandibles. Unfortunately, this method can not be applied to other parts of beetle that the segmentation has too many noises, such as pronotum images.

Recently years, machine learning is developing rapidly, specifically deep learning (CNN). It exists in most of the fields, especially in computer vision. We can finish a lot of difficult tasks with a deep convolution neural network such as classification [6, 7], image recognition [8, 9, 10], speech recognition [11, 12] and language translation [13, 14]. Using CNN to determine landmarks on 2D images will produce good results and it may be a good solution for the un-segment-able images. Yi Sun et al. [15] have proposed a cascaded convolutional network to predict the key points on the human face. Zhang et al. [16] optimizes facial landmarks detection with a set of related tasks such as head pose estimation, age estimation, ... Cintas et al. [17] have introduced a network to predict the landmarks on human ear images. In the same context, we have applied CNN to predict the landmarks on pronotum images. The predicted landmarks then refined to increase the accuracy of coordinates.

3 Automatic landmarks by using CNN

Deep learning presents a learning method with multiple levels of representation of connected layers (convolutional neural network). Data representation is transformed from a lower level to a higher level with many complex functions can be learned via backpropagation. In this section, we will present a CNN that we used to predict the landmarks on pronotum images. Besides, the techniques that applied to preprocess data before using for training the network.

3.1 Network architecture

Like the other networks [3, 10, 17], the proposed network consists of several common layers with different learnable parameters (Fig.2). It receives an input of $1 \times 256 \times 192$ to train, validate, and test. The network consists of three repeated-structure of a convolutional layer followed by a maximum pooling layer and a dropout layer. The depth of convolutional layers increases from 32, 64, and 128 with different size 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 probability values used for dropout layers are $0.1, 0.2$, and 0.3. At the end, three full connected layers have been added to the network. The outputs of the full connected layers are 1000, 1000, 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. Additional, to have a better control of overfitting, another dropout layer with a probability of 0.5 is inserted between the first two full connected layers [18].

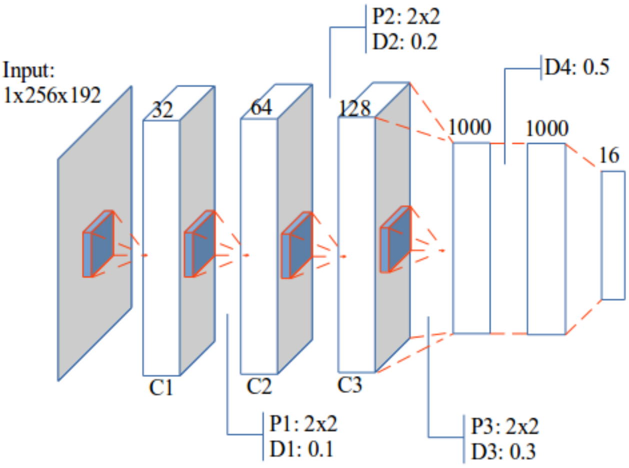


Figure 2: The architecture of the proposed convolutional neural network

During training, the values of learnable parameters have been updated to increase the accuracy of the network by applying gradient descent in backward phase. Therefore, the network is designed with a small sharing learning rate and momentum. Their values are updated over training time to fit with the number of epochs. The implementation of the network is done on Lasagne framework [19] which allows computing on GPU. The network has been trained on NVIDIA TITAN X cards.

3.2 Data processing

The dataset includes 293 pronotum images of beetles. All images are taken with the same camera in the same condition with a resolution of 3264×2448. Each image has been set 8 manual landmarks by biologists (Fig. 1). The dataset was split into two subsets: training (and validation) set contains 260 images and testing set includes 33 images. In most of CNNs [3, 15, 6, 17], the size of the input was limited to 256 pixels. In our case, the resolution of input image seems that too large and it becomes a difficulty for the network. So, the images are down-sampling to a new resolution 256×192 before training and testing. Of course, the coordinates of manual landmarks are also scaled to fit with the new resolution of images.

Besides the size of the input, the number of images is also a challenge when applying CNN. Normally, training a CNN with a large dataset will give us the result better than when we training CNN on a small dataset. Moreover, working with a small dataset, we can meet a popular problem, *overfitting*. So, we need to enlarge the size of the dataset instead of 293. In image processing, we usually apply transform procedures (translation, rotation) to generate a new image but in fact, when we compute the value of the pixels, it does not change while CNN computes the values of the pixels. Therefore, we have applied two other procedures to increase the number of images in the dataset. To address this problem, we have applied two procedures to enlarge the size of the dataset.

The first procedure was 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 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 is splitting the channels of RGB images. It means that we separate the channels of RGB into three gray-scale images. This work seems promising because the network works on single-channel images. At the end, we can generate six versions from an image, the total number of images used to train and validate is $260 \times 7 = 1820$ images (six versions and original image).

3.3 Training and experiments

The network was trained on a dataset of 1820 images. The number of images that used for training and validation is splitted randomly by a ratio (training: 60%, validation: 40%) that has been set during the network setup. During the training, the network learned the information through a pair of *(image, landmarks)* in training set. At the test phase, the image without landmarks was given to the trained network and the predicted landmarks will be given at the output. In practical of 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 [20]. According to [20], 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.

The training was finished in 5000 epochs[1]. The learning rate was initialized at 0.03 and stopped at 0.00001, while the momentum was updated from 0.9 to 0.9999. Because landmarks prediction can be seen as a regression problem in deep learning. Therefore, the root mean square error (RMSE) was used as a quality metric to evaluate the result and compute the losses of the proposed architecture.

Fig. 3 shows the training error and the validation error during training time. The blue curve presents RMSE error on training data. The green curve presents the validation error. Clearly, the losses are very different from the beginning. But, the difference is narrowed when the epoch increase.

Besides the losses during training, the accuracy on coordinates of predicted landmarks of the test images is also considered. Firstly, the trained model was used to predict the landmarks on all images in the test set. Then, the distances (in pixels) between manual and corresponding predicted landmarks in each image were calculated as the error distances. Finally, the error distance per landmark was calculated for all test images. Table.1 shows the average error distance given on each landmark. With the size of the images is 256×192, if we accept an error around 3% of the image size (~ 3.5 pixels), the error distances are acceptable.

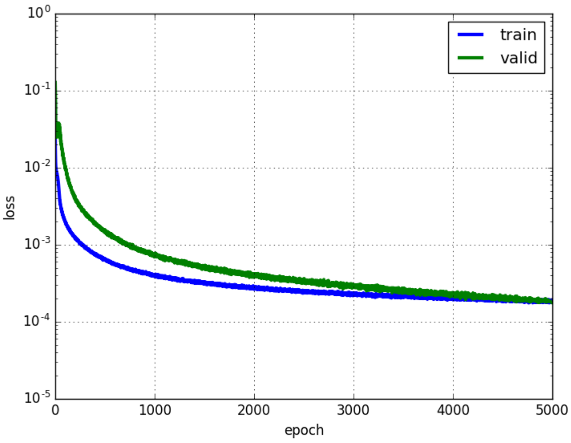


Figure 3: The loss curves during training of proposed network

#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

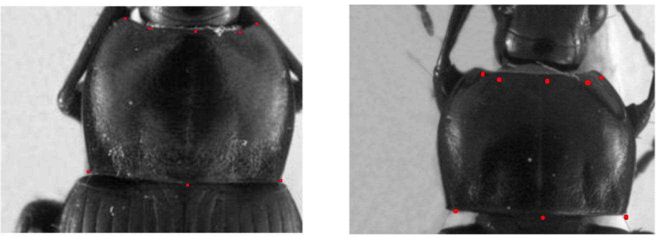
Table 1: The average error distance per landmark

Fig.4 shows the predicted landmarks on two test images. When we consider the accuracy of predicted landmarks by calculating the distance between manual and corresponding predicted landmarks, the accuracy on coordinates of predicted landmarks on Fig.4a is 99% and the propotion on Fig.4b is 80%.

To have a better evaluation of the predicted landmarks, we have applied the standard deviation [21] to quantify the dispersion of a set of distances. In this case, a predicted landmark is considered as acceptable if its error distance to the corresponding manual landmark is less than the average error (per landmark) plus standard deviation value. Fig. 5 shows the ratio of acceptable per landmark. Most of landmarks have been predicted with the accuracy grater than 80%. In which, the lowest and highest prediction accuracies are 83.62% and 89.08%, respectively.

As a result, the network is able to predict the landmarks

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



(a) Image with well-predicted landmarks

(b) Image with inaccuracy landmarks

Figure 4: The predicted landmarks on an image in test set. The read points present for the predicted landmarks

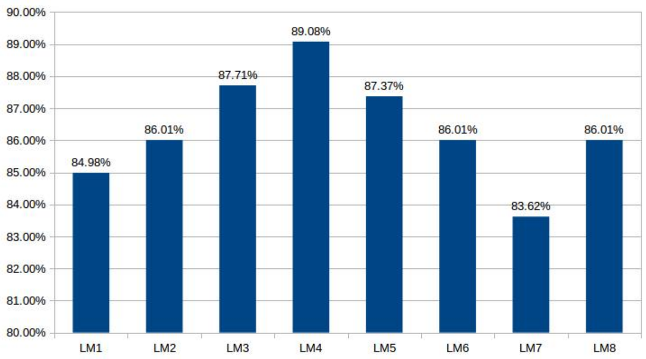
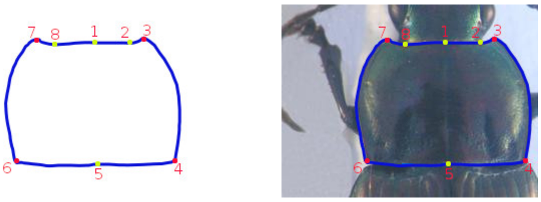


Figure 5: The proportion of acceptable predicted landmarks

on a test set of pronotum images. At statistic side, the predicted landmarks are acceptable. But in image processing side, we expect more about the accuracy (coordinates of predicted landmarks), and the result of CNN is still needed to improve. However, improving on all landmarks is difficult to perform because a number of landmarks have stayed inside the pronotum. Instead of, we will improve the result of the landmarks that stay in the shape of the pronotum.

4 Improving the predicted landmarks

In the previous section, we have applied a CNN to predict the landmarks on pronotum images. The results of experiments have shown that the network has worked well to detect the landmarks on the images in the test set. However, when we consider the predicted landmarks by displaying the landmarks on the images, the result is still not precise, specifically, the landmarks stayed on the shape border and at the corner of pronotum. In this section, we describe a scenario to improve the locations of the predicted landmarks which stay at the corner of the pronotum shape, *i.e* 3^{rd} and 7^{th} landmarks, called *corner landmarks*. Fig.6 shows the simulation of a pronotum with its shape and manual landmarks. Fig.6a shows the shape of pronotum and its manual landmarks. The red points represent for the corner landmarks, the yellow points represent the landmarks inside the pronotum. Fig.6b display an overlap between the shape and real image.



(a) Pronotum shape with manual landmarks

(b) Overlap the pronotum shape and real image

Figure 6: The simulator of a pronotum with its shape

The main idea of this process is generated a general curve through a landmark and tried to adapt the predicted landmark to corresponding general curve, this work looks like a Procrustes analysis. The scenario starts from the segmentation result of the image. Firstly, from the training dataset, a "mean curve" is generated for each manual landmark that stays in the shape of pronotum (i.e 3^{rd}, 7^{th}). Secondly, for each predicted landmark from CNN, we try to adapt the landmark to the contour of pronotum. Finally, we search a pixel around "adapted point" that the curve via it is closest to the "mean curve".

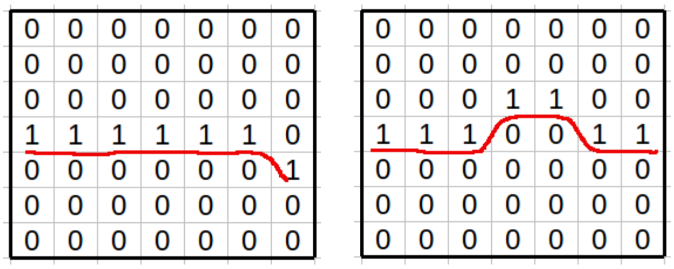
4.1 Generating mean curve

The mean curve was generated from a set of training images with their manual landmarks. For each image and its manual landmarks, the process to extract the curve via a manual landmark in an image as following:

1. Segmenting the image to obtain the contour (see [5]),
2. Extracting a patch with the size of 7×7 centering at the manual landmark[2],
3. Detecting the contour points inside the patch.

The mean curve has obtained by calculating the mean coordinates of points of curves via the manual landmarks that have the same order (same position).

Fig.7 shows the patch where contains the mean curve through the 3^{rd} and the 7^{th} landmarks. In the images, the *1-positions* and *0-positions* represent for the pixels belongs to the mean curves (red lines) and not, respectively.



0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
1	1	1	1	1	1	0
0	0	0	0	0	0	1
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(a) The 3^{rd} landmark

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	1	0	0
1	1	1	0	0	1	1
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(b) The 7^{th} landmark

Figure 7: The patches contain mean curves through the manual landmarks

[2] The size of patch was chosen by experiments.

4.2 Adapting and improving the predicted landmark

The predicted landmark (corner landmark) closed to the pronotum but it did not mostly stay on the curve while most of the manual landmark has stayed on the border of pronotum. Therefore, before improving the position of predicted landmark, we need to adapt the landmark on the curve. This process was finished in a simple way that we find the point on the curve that nearest with the predicted landmark.

After finding the point on contour to replace the predicted landmarks. A patch P_l with the size of 15×15 is created centering at this point. For each contour point in patch P_l, a small patch P_s of size 7×7 is extracted (this patch has the same size with the patch that contains the mean curve). Then, the curve belongs to P_s was detected and compared to the mean curve. The process has been repeated until all contour points of P_l are considered. The point that has the minimum distance with the mean curve will be kept and it was the new coordinates of predicted landmarks.

4.3 Results

The proposed scenario has experimented on the 3^{rd} and the 7^{th} landmarks. For each landmark, the mean curve has been generated from data training images and their manual landmarks. During improving the coordinates of predicted landmarks, Root Mean Square Distance (RMSD) is used to compute the distance between a candiate curve and mean curve. Table.2 shows the average error distance of each landmark after improving. *The first column* presents for the order of landmarks; *the second column* presents for the average error distance from CNN result (Table 1); *the third column* presents for the average error distance after we improve the coordinates of predicted landmarks. Clearly that, the coordinates of predicted landmarks have been significantly improved (33.84% for 3^{rd} landmark and 26.3% for 7^{th} landmark).

#Landmark	CNN	Improved
3^{rd}	4.2959	**2.8421**
7^{th}	4.6360	**3.4166**

Table 2: A comparison of the average error distances on each landmark

In another side, we would like take into account the number of acceptable landmarks by applying the same way than we have done with CNN (standard deviation). Fig.8 shows a comparison of the proportions of acceptable per landmark (in percent) between two periods (before and after improving the coordinates of predicted landmarks). According to the figure, the proportion of acceptable on 3^{rd} landmark has improved around 2.05%, while this rate at 7^{th} landmark is 7.16%. Comparing with the output from CNN, the coordinates of predicted landmarks at position 3 and 7 have been vastly improved.

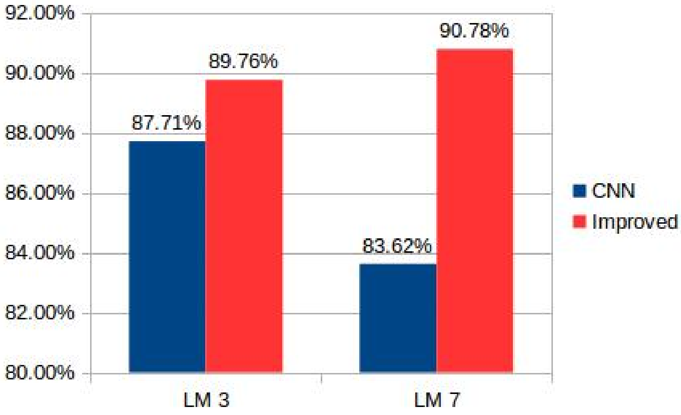


Figure 8: A comparison of the proportions of acceptable predicted landmarks on 3^{rd} and 7^{th} landmarks. The *blue* and *red* columns present for the ratio of accuracy of predicted landmarks which given by CNN and improving method, respectively.

As a result of working, the program outputs the predicted-landmarks of the images as TPS files. With the outputs are TPS files, the user can use MAELab framework[3] to display the landmarks on the images.

5 Conclusions

In this paper, we have presented a scenario to predict the landmark on beetle's head images. It includes two steps: predicting and improving the coordinates of estimated landmarks.

The prediction step has been done by applying deep learning. A CNN has been designed with 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, the CNN have been trained with several times in different selections of training data. After training, the network was able to predict the landmarks on the images in the test set. The result has been evaluated by comparing the coordinates between predicted and manual landmarks. The quality of prediction allows using automatic landmarking to replace manual landmarks in some aspects. However, we expect more accuracy on the landmarks belongs to the contour of the images, and they have been improved.

The improvement of predicted landmarks has finished as a technique of Procrustes analysis. From the training dataset and their manual landmarks, the mean curves through the landmarks in the contours have been generated. Then, the predicted landmarks by CNN have been adapted and improved. The results have shown that using the convolutional network to predict the landmarks on biological images is promising good results in the case that the image was difficult to segment. But with a limited number of data, we need to improve the result by another method. Therefore, future research in landmarking identification appears as an improved of the worth exploring.

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

Acknowledgements

The research has been supported by INRA. We would like to thank the biologists, who have provided the dataset. We thank also our colleague, Nicolas PARISEY, who provided insight and expertise that greatly assisted the research.

References

[1] S. Palaniswamy, N. A. Thacker, and C. P. Klingenberg, "Automatic identification of landmarks in digital images," *IET Computer Vision*, vol. 4, no. 4, pp. 247–260, 2010.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pp. 253–256, IEEE, 2010.

[4] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[5] V. L. LE, M. BEURTON-AIMAR, A. KRÄHENBÜHL, and 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), May 2017.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

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

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, *et al.*, "Going deeper with convolutions," Cvpr, 2015.

[9] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[10] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5325–5334, 2015.

[11] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černockỳ, "Strategies for training large scale neural network language models," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pp. 196–201, IEEE, 2011.

[12] 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*, vol. 29, no. 6, pp. 82–97, 2012.

[13] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On using very large target vocabulary for neural machine translation," *arXiv preprint arXiv:1412.2007*, 2014.

[14] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[15] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3476–3483, 2013.

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

[17] 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*, vol. 6, no. 3, pp. 211–223, 2016.

[18] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[19] S. Dieleman, J. Schlter, C. Raffel, E. Olson, S. K. Snderby, D. Nouri, *et al.*, "Lasagne: First release.," Aug. 2015.

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

[21] J. M. Bland and D. G. Altman, "Statistics notes: measurement error," *Bmj*, vol. 313, no. 7059, p. 744, 1996.