Towards landmarks prediction with Deep Network

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, fine-tuning, recogntion, procrustes.

Abstract

Morphometry landmarks are used in many biological applications. Mostly, the landmarks are defined manually or semi-automatically by applying the image processing techniques. In recent years, deep learning is known as a good solution to achieve image analysis tasks. It appears in many fields such as classification, recognition, face detection. In this context, we present a convolutional neural network to predict the landmarks on 2D biological images, specifically beetle's images. The proposed model is designed from an *elementary block* of three layers: convolution, pooling and dropout. The experiments on the proposed network have been done in two steps: training from scratch and fine-tuning from a trained model. The dataset includes the images of collecting from 293 beetles (on *head, pronotum, body* parts). Among these, a set of manual landmarks has been built for each part by the biologists. In this work, we have worked on prediction of pronotum landmarks. The quality of predicted landmarks is evaluated by calculating the distance in pixels between the coordinates of the predicted landmarks and manual landmarks which are considered as ground truth.

1 Introduction

Morphometrics landmarks (or point of interest) is important features in many biological investigations. They are usually used to analyze the forms of biological organs or organisms. Their analysis is mainly based on their coordinates. Depending on the problem to study, the number of landmarks may be more or less high and their location can be on the shape (border) or inside the object. For examples, the landmarks on Drosophila wings [1] have stayed on the veins of the wings but the landmarks on human ear [2] can be located at the ear border or inside it. Currently, the landmarks are set manually by the biologists, one can note that this work is time-consuming and difficult to reproduce. Therefore, a method that proposes automatically the coordinates of landmarks could be a concern.

In image processing, segmentation is most often the first and the most important step. This task remains a bootleneck to compute features of an image. In some cases, the object of interest is easy to extract and can be analyze with the help of a lot of very well-known image analysis procedures. In a previous studying [3], we have analyzed beetle mandibules with a set of algorithms based on the Hough Transform procedure [4], SIFT [5] and SURF [6] algorithms could also be suitable to work on this topic for example. But in some cases, the question of how to segment the object of interest consumes the most part of a time project. It is why we have turned our studying to a way to analyze image without a segmentation step. The application has been again on beetles images but on *pronotum, head, body* parts. As these parts have not been separated from each other, their segmentation by image processing procedures has been given up. Coordinates of manual landmarks for each part have been provided and are considered as the ground truth to evaluated the predicted landmarks by the algorithms. We have focus on the pronotum parts for this studying. Fig.1 shows the 8 landmarks that we are looking for.

To achieve the landmarks prediction, a Convolutional Neural Network (CNN)[7] has been designed based on Lasagne library[8]. From a first model version, the network has been trained from scratch on the dataset of pronotum images. In a second step the training has been modified to include a fine-tuning [9] stage.

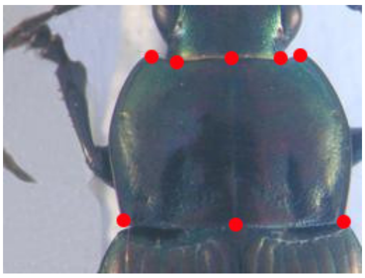


Figure 1: An example of pronotum images and its manual landmarks

In the next section, we present related works about automatic estimation landmarks on 2D images. In section 3, we present the architecture of the network and the procedure to enlarge the dataset. In section 4 we compare the results obtained with the first model and these ones after fine-tuning.

2 Related works

Deep learning methods are coming from machine learning theory. They have been introduced in the middle of previous century for artificial intelligence applications but their application encounters problems to take into account real cases. More recently, the improvement of computing capacities, both in memory size and time with GPU programming has opened a new challenge for deep learning. Many deep learning architectures have been proposed to solve the problems of classification [10, 11], image recognition [12, 13, 14], speech recognition [15, 16] and language translation [17, 18]. Along with that developments, many frameworks have been built such as Caffe [19], Theano [20], Tensorflow [21],.... These frameworks help the users to design their application by re-using networks architecture they propose. In image analysis domain, deep learning, specifically with CNN, can be used to predict the key points on the image. Yi Sun et al. [22] have proposed a cascaded convolutional network to predict the key points on the human face. Zhang et al. [23] optimizes facial landmarks detection with a set of related tasks such as head pose estimation, age estimation, ... Cintas et al. [2] have introduced a network to predict the landmarks on human ear images to characterize ear shape.

In geometric morphometry, landmarks or points of interest are one of the important features. Landmark studies have traditionally analyzed on 2D images. Depending on if the analyzed images are easy or not to segment, setting landmarks can apply the different methods. When segmentation can be applied, Lowe et al. [5] 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. [4] have applied probabilistic Hough Transform to automatically estimate the landmarks in images of Drosophila wings. Krahenbuhl et al. [3] have extended Palaniswamy's method to detect landmarks automatically on beetles mandibles. Unfortunately, after testing, when the segmentation has not good quality, we have observed that re-using this method produces too many noises. This is why we have turned our work on deep learning algorithms in order to find suitable solution to predict the landmarks on images hard to segment.

3 Network model

Deep learning is 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 one with many complex functions that can be learned via backpropagation. In this section, we present the initial version of the CNN that we have used to begin to predict the landmarks.

3.1 Network architecture

The first step to work in CNN is to study the network architecture. After several tests, we have chosen to work with a model provided in Lasagne framework [8] coming from Theano [20]. We will first present the original model and then, we will describe how we have modified it by definition of an *elementary block* that we compose in the final model.

Like the networks have been proposed by Cintas et al. [2], Li et al. [14], and LeCun et al. [7], the proposed network consists of common layers with different learnable parameters. It receives an input image with the size $1 \times 256 \times 192$ to train, to validate, and to test. The network consists of three repeated-structures of a convolutional layer followed by a pooling layer (keeping the maximum value). The depth of convolutional layers increases with different size of the filter kernels. All the kernels of pooling layers have the same size. At the end, three full connected layers are added to the network. The output of the last full-connected layer corresponds to the 16 values which are the coordinates of the 8 landmarks to predict.

Experiments with this origin model show that this architecture is still not good enough to predict the landmark positions precisely. For instance, overfitting appears during training and validation steps. Srivastava et al. [24] suggest to use dropout sequence to correct overfitting artefacts. Dropout step randomly drops units from the neural network during training and so includes some variations between the different runs. We have updated the model architecture in that way. An *elementary block* is defined as a sequence of convolution (C_i), pooling (P_i) and dropout(D_i) layers that can be repeated several times before to achieve the computation with the full-connected layers. For our purpose, we have assembled 3 *elementary blocks* in our model (see Fig.2). The parameters for each layer are as below, the list of values follows the order of *elementary blocks*:

- CONV layers:
  - Number of filters: $32, 64$, and 128,
  - Kernel filters size: $(3 \times 3), (2 \times 2)$, and (2×2),
  - Stride values: $1, 1, 1$,
  - No padding is used for CONV layers.
- POOL layers:
  - Kernel filters size: $(2 \times 2), (2 \times 2)$, and (2×2),
  - Stride values: $2, 2, 2$.
  - No padding is used for POOL layers.
- DROP layers:
  - Propabilities: $0.1, 0.2$, and 0.3.

In the last full-connected layers (FC), the parameters are: FC1 output: 1000, FC2 output: 1000, FC3 output: 16. As usual, a dropout layer is inserted between FC1 ond FC2 with a probability equal to 0.5.

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[1]. The network is designed to finish the training in 5000 epochs. The learning rate was initialized at 0.03 and stopped at 0.00001, while the momentum was updated from 0.9 to 0.9999.

The implementation of the network is done on Lasagne framework [8] which allows computing on GPU. The network has been trained on NVIDIA TITAN X cards.

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

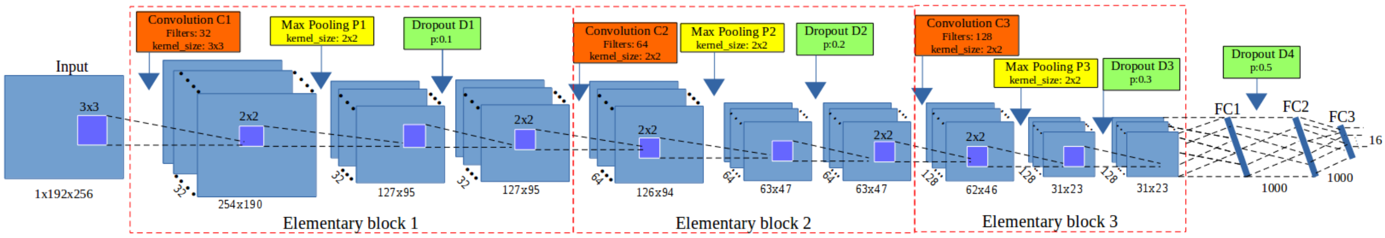


Figure 2: Network architecture using 3 *elementary blocks*. Convolution layer in red, pooling in yellow and dropout in green color.

3.2 Expanding the dataset

The dataset includes 293 images of beetles (for each part). All images are taken with the same camera in the same condition with a resolution of 3264×2448. Each image has the manual landmarks setting by biologists i.e pronotum has 8 manual landmarks. Thte experiments have been designed with a testing set which includes 33 and the remained 260 images are put in two subsets, one for training and one for validation. For performance considerations, in most of CNNs [7, 22, 10, 2], the size of the input is limited to 256×256 pixels, thus we have down-sampling our images to a new resolution 256×192 (to respect the ratio between x and y), of course the coordinates of manual landmarks have been also scaled to fit with the new resolution.

One of the main characteristics of CNN is to deal with a huge size of dataset and one can consider that only several hundreds of images is not relevent to use CNN. Moreover, working with small dataset can push us again to the popular problem of *overfitting*. A way to to enlarge the dataset size has to be considered. In image processing, we usually apply transform procedures (translation, rotation) to generate a new image but unfortunatly the methods to compute features through a CNN mostoften are translation and rotation independant. Another way to enlarge the dataset has to be imagined.

A first procedure has been applied to change the value of each channel in the original image. According to that, 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 separates the channels of RGB into three gray-scale images. As the network works on single-channel images we are able to generate six versions of the same image, the total number of images used to train and validate is $260 \times 7 = 1820$ images (six versions and original image). This has been an efficient way to proceede the dataset expansion.

3.3 First results

The set of images that have been used for both training and validation has been built randomly from the original dataset with a ratio of 60% for training and 40% for validation. The training step takes into account a pair of informations *(image, manual landmark coordinates)*. At the test phase, images without landmarks is given to the previous trained network that produces as output coordinates of the predicted landmarks. To obtain a fast convergence during the computing of CNN, it is useful to normalized the pixel color value between $[0; 1]$ range, instead of $[0; 255]$ [25]. The coordinate values has also been normalized.

In order to test predictions for all pronotum images (instead of only 33 images), an algorithm to choose the test images is executed, called *round*. For each round, a set of 33 images have been chosen for the test set in the whole dataset; the remaining images have been put into the training set. Following that, the network will be trained with many different training datasets and the output model will be used to predict the landmarks on the images in the corresponding test set. After 9 rounds all images have been tested. Table.1 resumes the training losses for the 9 rounds.

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

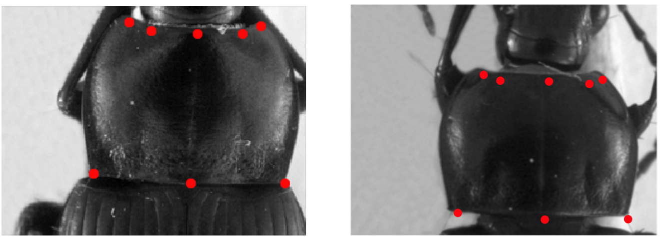
Table 1: The losses during training the model on pronotum images dataset

The main goal of the computing is to predict position of landmarks so the distance (in pixels) between the manual ones (the ground truth) and the predicted ones has to be now considered. A correlation test gives us a good correlation between position of a manual landmark and its corresponding predicted one. But we have considered that this measure is not good enough to provide a useful result to biologist. We have prefered to evaluate the distance in pixels between the ground truth and the prediction. Table.2 shows the average distance between manual and predicted landmarks for all images, landmark per landmark. With images of 256×192 size we can consider that an error of 1% corresponds to 2 pixels that could be an acceptable error. Unfortunatly our results exhibits average distance of 4 pixels in the best case, landmark 1 and more than 5 pixels, landmark 6. More the error distance is more than 2% pixels.

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

Table 2: The average error distance per landmark

To illustrate this purpose, Fig.3 shows the predicted landmarks on two test images. One can note that even some predicted landmarks (left side) are closed to the manual ones, in some case (right side) the predicted ones are far from the expect results. The next step has been dedicated to the improvement of these results.



(a) Image with well-predicted landmarks

(b) Image with inaccuracy landmarks

Figure 3: The predicted landmarks, in red, on the images in test set.

4 Fine-tuning to transfer learning

In section 3.3, the proposed network has been experimented by training from scratch on pronotum dataset. The results of experiments have shown that the network has worked well to detect the landmarks on the pronotum images. However, when we consider the predicted landmarks by displaying the landmarks on the images, the result is still not precise, the average error still high (≥ 4 pixels).

In order to reach more acceptable results for biologists, we have broadened the model with the step of transfer learning. That is a method that re-uses the model developed for a specific task/dataset to lead another task with another dataset. This allows rapid progress or improved the performance of the model on the second task [26]. The most popular example has been given with the project ImageNet of Google [27] which has labelled several millions of images. The obtained parameter values can be used in another context to classify another dataset, eventually very different dataset [28]. The name of this procedure to re-use parameters to pre-train a model is called *fine-tuning*.

Fine-tuning is not only to replace and to retrain the model on the new dataset, but also to fine-tune the weight of 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, of course the size of the used data has been drastically decreased. In that way, the network has been trained on a 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.

4.1 Training data preparation

The training dataset includes a combination of the images from three sets: pronotum, body and head (Fig.4). For each set, 260 original images have been chosen radomly for training and validation. By applying the same procedure in section 3.2, the training dataset was enlarged to 5460 images ($260 \times 7 \times 3$). However, the number of manual landmarks on each part is difference: 8 *landmarks on pronotum part,* 11 *landmarks on body part, and* 10 *landmarks on head part*. The manual landmarks have a specific meaning for the biologists. So, we can not insert the landmarks arbitrarily. Instead of, we will keep the smallest number of landmarks among three parts and we remove some landmarks on other parts. Therefore, we kept the number of the landmark on pronotum as a reference and we suppressed some landmarks on the body and head part. 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.4).

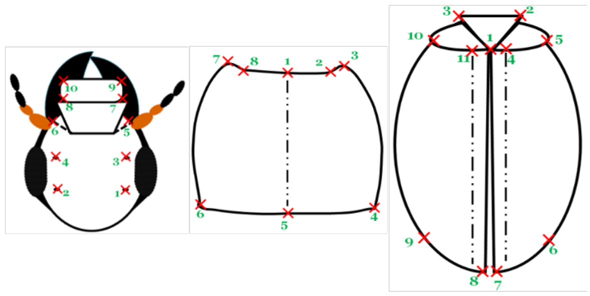


Figure 4: A presentation of head, pronotum and body part with corresponding manual landmarks

4.2 Using fine-tuning for pronotum dataset

At the first step, the network is trained with 5460 images following the same way than explained in (section 3.1). After that, this trained model has used to fine-tune the pronotum dataset. To compare the result with the previous one, the trained model has been fine-tuned in many rounds with different datasets. The losses during fine-tuning are shown in Table.3. Comparing with the losses when we trained the model from scratch (Table. 1), the validation losses of this scenario have been significantly improved (around 40%).

The output model has been used to predict the landmarks on the test images. Then, average error based on the distance between predicted and corresponding manual landmarks has been computed. The results are shown in Table.4. The **Average** (CNN) column reminds the average distance obtained previously. The **Average** (fine-tuning) column presents for the new average distance after fine-tuning the pronotum from the trained model. Besides, the standard errors of both cases have

Round	Training loss	Validation loss
1	0.00019	0.00009
2	0.00018	0.00010
3	0.00018	0.00010
4	0.00019	0.00008
5	0.00019	0.00009
6	0.00018	0.00008
7	0.00019	0.00008
8	0.00018	0.00006
9	0.00018	0.00009

Table 3: The losses during fine-tuning model

been presented (SD columns). It is clearly shown that the result of predicted landmarks with the help of fine-tuning is more precise than the first way to do it.

Landmark	CNN		Fine-tuning	
	Average	SD	Average	SD
LM1	**4.002**	**2.5732**	**2.486**	**1.5448**
LM2	4.4831	2.7583	2.7198	1.7822
LM3	4.2959	2.7067	2.6523	1.8386
LM4	4.3865	3.0563	2.7709	1.9483
LM5	4.2925	2.9086	2.4872	1.6235
LM6	**5.3631**	**3.4234**	**3.0492**	**1.991**
LM7	4.636	2.8426	2.6836	1.7781
LM8	4.9363	3.0801	2.8709	1.9662

Table 4: A comparing between the average error distances, the standard deviation values per landmark of two steps.

To illustrate the final results, we display the distribution of the distance of both the best and the worst results (resp. landamark 1 and 6). The Fig.5 shown in (a) and (b) diagrams how much the average distance (blue lines) and standard error (red lines) have been improved for the landmark 1, the (c) and (d) diagrams for the landmark 6.

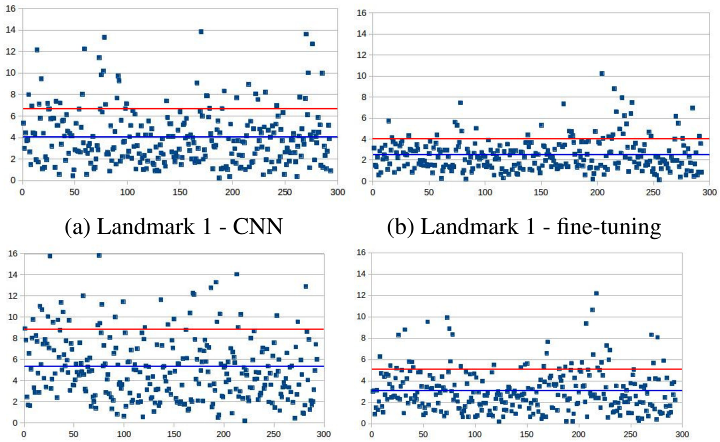
As a result of working, the program outputs the predicted-landmarks of the images as TPS files. With the outputs are TPS files, the user can use MAELab framework[2] to display the landmarks on the images.

5 Conclusion

In this paper, we have presented a CNN to predict the landmarks on beetle's pronotum images. The CNN network has been designed with three times repeated of elementary block which consists of a convolutional, a max pooling, and a dropout layers, followed after by the full connected layers. In the training phase, the CNN have been trained several times with different selections of training data.

In the first step, the model has been trained from scratch and tested on the dataset of pronotum images. While in the second step, the model has been trained on a dataset includes all the images from three parts of beetles. Then, the trained model has been used to fine-tune and to test on pronotum images.

The result has been evaluated by comparing the coordinates between predicted and manual landmarks. The results have shown that using the convolutional network to predict the landmarks on biological images is able to provide good results in the case that the image was difficult to segment. The quality of prediction allows using automatic landmarking to replace manual landmarks in some aspects. Training model with fine-tuning the trained model has provided the best results. Future works will have to test our model on another kind of biological images.



(a) Landmark 1 - CNN (b) Landmark 1 - fine-tuning

(c) Landmark 6 - CNN (d) Landmark 6 - fine-tuning

Figure 5: The distribution of distance error on 1^{st} and 6^{th} landmarks of all images in two testing step (CNN and fine-tuning). The blue and red lines present for the average distances and standard deviation values, respectively.

Acknowledgements

The research has been supported by DevMAP project[3]. We would like to thank the my colleague, ALEXIA Marie, who have provided manual landmarks on beetle images.

References

[1] A. Sonnenschein, D. VanderZee, W. R. Pitchers, S. Chari, and I. Dworkin, "Supporting material and data for "an image database of drosophila melanogaster wings for phenomic and biometric analysis"," 2015.

[2] 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.

[3] V. L. Le, M. Beurton-Aimar, A. Krahenbuhl, and N. Parisey, "MAELab: a framework to automatize land-

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

[3] https://www6.rennes.inra.fr/igepp_eng/Research-teams/Demecology/Projects/INRASPEDevMAP

mark estimation," in *WSCG 2017*, 25th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2017, (Plzen, Czech Republic), May 2017.

[4] 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.

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

[6] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, pp. 404–417, Springer, 2006.

[7] 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.

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

[9] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, pp. 3320–3328, 2014.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[20] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016.

[21] M. A. et al, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] L. Torrey and J. Shavlik, "Transfer learning," *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, vol. 1, p. 242, 2009.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[28] J. M. et al, "Fine-tuned convolutional neural nets for cardiac mri acquisition plane recognition," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 5, no. 5, pp. 339–349, 2017.