Landmarks prediction on Beetle anatomical by applying Convolutional Neural Network

LE Van Linh

June 10, 2018

Abstract

In morphometric studies, landmarks are regarded as one of important properties to analyze the object's shape. Especially in biology, collecting complete landmarks can give the information of the oganism. From that, the biologists can study the complex interactions between evolution of the oganism and environment factors. Currently, landmarking is most oftne done by manually. It is time-consuming and difficult to reproduce. In the context of this study, we order to investigate the posibiliy of automatic idenfity landmarks on Beetles. The dataset, that we focus, is one of the most common insect of North-Western France, Carabidae (Beetle). To do this work, we propose a convolutional neural network (CNN) to predict the landmarks on the parts of Beetle: pronotum, head, and body. The proposed model will witnessed on the datasets includes 293 images (for each part) in different sizes: 192×256 and 96×96. The experiments will be done in two directions: training the model from scratch and using fine-tuning. During the experiments, the coordinates of automatic landmarks will be evaluated by comparing with the manual coordinates, which are given by the biologists.

1 Introduction

Morphometric landmarks are important features in biological investigations. They are using to analyze the shape of the organisms. Depending on the organisms, the number of landmarks on their shapes is different. In addition, when we consider the location of the landmarks with the object, most of landmarks are located on the edges (shape), for example, the landmarks on wings of Drosophila fly [1]. Besides, we can also see the landmarks which stayed inside the anatomical part, i.e, landmarks on pinna of human ears [2]. Currently, the landmarks are set manually by the biologsits. This operation is a time-consuming process and difficult to reproduce. Consequently, a process that proposed automatically the coordinates of landmarks could be interested.

A way to automatize landmarks setting is using the image processing techniques. In which, segmentation is a most often step of the methods because the result from segmentation step is very useful for many purposes. Depending on the characteristics of the image, we can choose the suitable methods to finish target work. In some cases, the object of interest is easy to extract and landmarks can be set by applying a lot of very well-known image analysis procedures, i.e. the landmarks on fly wings [6]. In a previous study [7], we have analyzed two parts of beetle: left and right mandibles. These parts are easy to segment. In that work, we have applied a set of algorithms based on segmentation, image alignment and SIFT [3] descriptor to estimate the landmarks on mandibles.

Unfortunately, the images of other parts are not simply as the mandibles. Besides the main objects, the images have also the different parts, i.e. we have a part of head and the legs

in pronotum images. So, they become very noisy. If we would like to segment the object as traditional processes, the process may have consumed a lot of time and difficult to choose a proper method. This is the reason that we turn the landmarking problem on some parts of beetle (i.e, pronotum, head, and body) to a way of analyzing images without the segmentation step.

As the beetles have not been dissected, their anatomical parts have not been set apart. So image segmentation of each part, as they are still attached to the whole specimen, is problematic and has been given up. Coordinates of manual landmarks for each part have been provided by the biologists and they are considered as the ground truth to evaluate the predicted ones by our methods. Fig.1 shows the parts of beeltes and their manual landmarks what we are looking for.

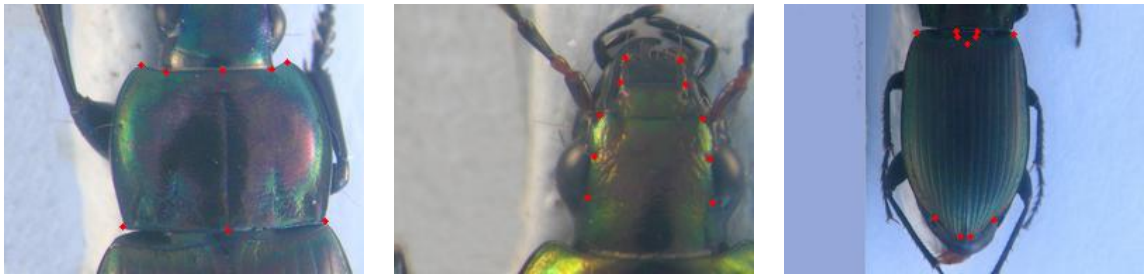

Figure 1: The dataset images with their manual landmarks.
From left to right: pronotum, head, and body

To achieve the landmarks prediction, we have proposed a CNN model [4] by using Lasagne library [5]. To evaluate the effective of the model, we have applied two scenarios to predict the landmarks on beetles anatomicals: training from the scratch and applying fine-tuning. The experiments are done on two datasets with different size of the images: 192×256 and 96×96.

Our contributions in this study are as follows: In the next section, we present the related works about automatic estimation landmarks on 2D images. In section 3, we describe the process to augment the dataset. The architecture of proposed network will be presented at section 4. All the experiments of our model will be shown in section 5. It includes the results to evaluate the model and comparison between two working strategy on neural networks.

2 Related works

In geometric morphometry, landmarks (or points of interest) are important features to describe a shape. Depending on the difficulty to segment the objects inside the images, setting automatic landmarks can rely on different methods. When segmentation can be applied, Lowe et al. [3] 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. [6] have applied probabilistic Hough Transform to automatically estimate the landmarks in images of Drosophila wings. In a previous study [7], we have extended Palaniswamys method to detect landmarks automatically on beetles mandibles with good results. Unfortunately, when the segmentation is not precise, we have observed that the results are getting worse. This is why we have turned our work on Deep Learning algorithms in order to find a suitable solution to predict the landmarks without any segmentation step.

Deep Learning models are coming from 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. More recently, the improvement of computing capacities, both in memory size and computing time with GPU programming has opened new perspective

for Deep Learning. Many deep learning architectures have been proposed to solve the problems of classification [8], image recognition [9], speech recognition [10, 11] and language translation [12]. To implement the algorithms, many frameworks have been built such as Caffe [13], Theano [14], Tensorflow [15],.... These frameworks help the users to design their application by re-using already proposed network architectures. In image analysis domain, Deep Learning, specifically with CNN, can be used to predict the key points in an image. Yi Sun et al. [16] have proposed a cascaded convolutional network to predict the key points on the human face. Zhang et al. [17] optimize 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 for biometric analysis. In the same way, we have applied CNN computing to predict the landmarks on beetles anatomical parts. The predicted landmarks will be found in two strategies: training the model from scratch and applying a fine-tuning process.

3 Dataset

The data includes images in three sets of the beetle: pronotum, body, and head (Fig.**??**). Each dataset includes 293 color images which 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. To obtain the datasets in different resolutions of images (256×192 and 96×96), we have applied different methods to down-sampling from the original images. In the first considered resolutions (256×192), we have simplied down-sampling from the original resolution. While, in the second considered resolution (96×96), the original images have been cropped from the left to obtain a new resolution of 2448×2448. Then, the cropped images have been down-sampled to the target resolution. Of course, the coordinates of manual landmarks have been also scaled to fit with the new resolutions.

One of the main characteristics of CNN is that it must use a huge number of data and one can consider that only several hundreds of images is not enough to feed a CNN. Moreover, working with small dataset can push us again to the popular problem of overfitting. So, a way to enlarge the dataset size has to be considered. In image processing, we usually apply transform procedures (translation, rotation) to generate a new image. Unluckily the methods to compute features through a CNN most often are translation and rotation independent. So, we have used another method to augment dataset from the down-sampling images.

The first procedure has been applied to change the value of each color channel in the original image. According to that, a constant is added to each channel of original image in each time. Each constant is sampled in a uniform distribution $\in [1, N]$ to obtain a new value caped at 255. For example, we can add a constant $c = 10$ to the red channel of all images in order to generate new images. This operation can be done for the three color channels (see the first row of Fig. 2).

The second procedure splits the channels of RGB images. It means that we separate the channels of RGB into three gray-scale images. The third row of Fig. 2 shows the values at three individual channels of an image.

At the end of this process, we are able to generate six versions of the same image, the total number of images used to train and to validate is $260 \times 7 = 1820$ images (six versions and original image). This has been an efficient way to proceed to the dataset expansion. Fig.2 presents two strategies that we used to augment the data: the image at the second row is the original image, three images at the first row are the result of the first procedure (adding a constant to each color channel), and the third color includes the result images of the second procedure (splitting the channels).

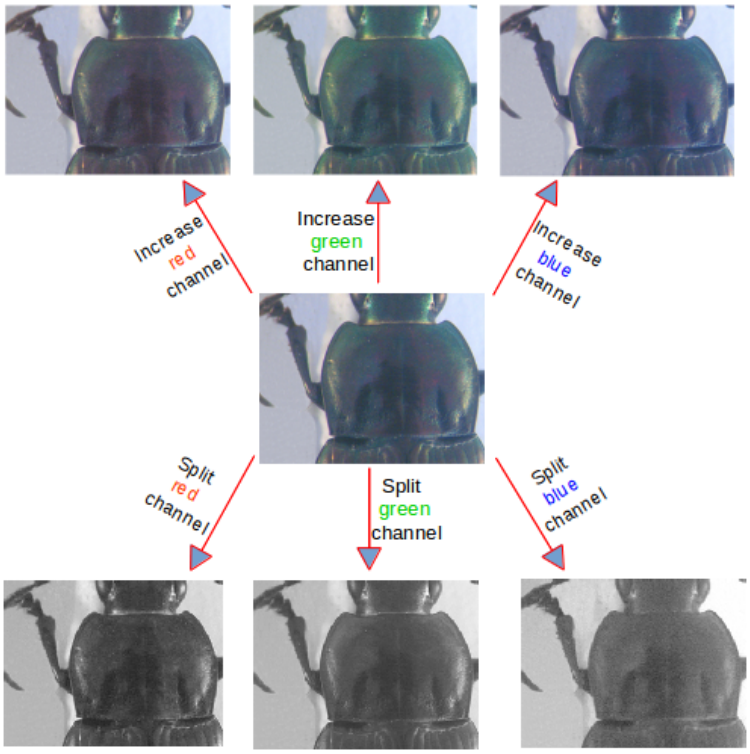

Figure 2: An example of augmentation data. From an original image, six augmented version have generated.

4 The model

The model, what we used to predict the landmarks, receives an image of $1 \times w \times h$ as the input where $w = 256$ or 96 and $h = 192$ or 96. The network was constructed from 3 *"elementary blocks"* following by 3 full-connected layers. An elementary block is defined as a sequence of convolution (C_i), maximum pooling (P_i) and dropout (D_i) layers. The *depth* of each convolutional layer and the drop probability of each dropout layer in each elementary block are different. While the parameters of pooling layers are kept the same values. The parameters for each layers are as below, the list of values follows the order of elementary blocks:

- CONV layres:
  - 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
- MAXPOOL layers:
  - Kernel filters size: $(2 \times 2), (2 \times 2)$, and (2×2)
  - Stride values: $2, 2, 2$
  - No padding is used for CONV layers
- DROP layers:
  - Probability: $0.1, 0.2$ 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. The output of the last full-connected layer is a set of 16 values which corresponds to 8 landmarks (x and y coordinates) which we would like to predict. Fig.3 illustrate the order of the layers in the network when we apply the model to predict the landmarks on the input of $1 \times 256 \times 192$.

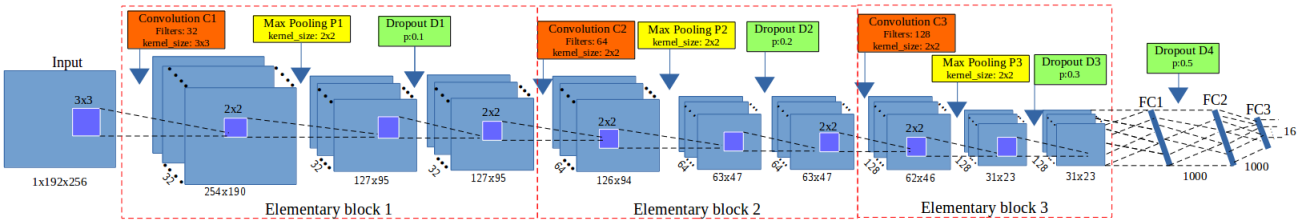

Figure 3: Network architecture using 3 *elementary blocks*. Convolution layer in red, pooling in yellow and dropout in green color.

The parameters of CNN are shown in Table.1.

Parameter	Initial value	End value
Epochs	10000	
Training batch size	128	
Testing batch size	128	
Learning rate	0.03	0.0001
Momentum	0.9	0.9999

Table 1: The network parameters in proposed model

5 Experiments

The model has been experimented on two resolution datasets: 256×192 and 96×96. For each resolution dataset, the model was evaluated in two strategies: training from scratch and fine-tuning. To have the predicted landmarks on all images, during the training process, the cross-validation has been applied to select the training data.

The dataset was trained on the model with 5000 epochs[1]. The images are randomly divided into training set and validation set followed the ratio 6 : 4 automatically during training step. The learning rate begin at 0.03 and decreased to 0.00001. In vice versa, the momentum started at 0.9 and increasing to 0.999 at the end of the training process.

To evaluate the prediction, the distance between predicted and manual landmarks have been computed in all images. Then, the average distance and standard deviation (SD) have been calculated on each landmark. Following sub-sections describe the average distance and SD of each dataset in each part of beetle.

5.1 Experiment on the first resolution dataset (256×192)

Table 2 shows the average distances and standard deviations on each landmarks of pronotum part. The table shows the results in three experiment processes on pronotum dataset: training from the scratch, fine-tuning with and without freezing some first layers in the model. The minimum distances belong to the first landmark, while worst distances belong to the sixth landmark in both three cases. In other view, when we consider the results of three experiment processes, the distances have been improved by applying fine-tuning: all the average distances and standard deviations in fine-tuning cases are smaller than the same values when we train the model from the scratch, for example, if we consider the first landmark of head part, the average distance has decreased from 4.002 to 2.486. However, we do not see the significant difference of the fine-tuning processes (with and without freezing some first layers).

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

Landmarks	From scratch		Fine-tuning without freezing		Fine-tuning with freezing	
	Average	SD	Average	SD	Average	SD
LM1	**4.002**	**2.5732**	**2.486**	**1.5448**	**2.47**	**1.55**
LM2	4.4831	2.7583	2.7198	1.7822	2.67	1.75
LM3	4.2959	2.7067	2.6523	1.8386	2.67	1.77
LM4	4.3865	3.0563	2.7709	1.9483	2.74	1.91
LM5	4.2925	2.9086	2.4872	1.6235	2.57	1.60
LM6	**5.3631**	**3.4234**	**3.0492**	**1.991**	**3.09**	**2.13**
LM7	4.636	2.8426	2.6836	1.7781	2.65	1.80
LM8	4.9363	3.0801	2.8709	1.9662	2.96	1.97

Table 2: The average distance and standard deviation on pronotum images (256 × 192)

Landmarks	From scratch		Fine-tuning without freezing		Fine-tuning with freezing	
	Average	SD	Average	SD	Average	SD
LM1	**3.87**	**3.40**	**2.34**	**3.11**	**2.33**	**3.10**
LM2	3.97	3.63	2.27	3.15	2.27	3.19
LM3	3.92	3.36	2.27	2.96	2.33	2.93
LM4	3.87	3.50	2.25	3.26	2.20	3.09
LM5	4.02	3.54	2.27	3.28	2.34	3.07
LM6	**4.84**	**3.59**	**3.14**	**3.47**	**3.12**	**3.30**
LM7	5.21	3.76	3.14	3.40	3.10	3.47
LM8	5.47	3.89	3.29	3.36	3.28	3.43
LM9	5.27	3.67	3.42	3.09	3.30	3.06
LM10	4.07	3.45	2.49	3.05	2.45	2.75
LM11	3.99	3.34	2.30	3.06	2.31	2.97

Table 3: The average distance and standard deviation on body (elytra) images (256 × 192)

Landmarks	From scratch		Fine-tuning without freezing		Fine-tuning with freezing	
	Average	SD	Average	SD	Average	SD
LM1	**5.53**	**3.40**	**3.03**	**1.89**	**3.03**	**1.88**
LM2	5.16	3.63	2.94	1.97	2.86	1.87
LM3	5.38	3.46	2.96	1.81	2.96	1.86
LM4	5.03	3.58	2.88	1.92	2.79	1.91
LM5	4.84	3.26	2.76	1.75	2.77	1.78
LM6	**4.45**	**3.37**	**2.67**	**2.02**	**2.61**	**1.97**
LM7	4.79	3.08	2.29	1.64	2.31	1.69
LM8	4.53	3.11	2.20	1.54	2.20	1.58
LM9	5.14	3.13	2.57	1.60	2.66	1.63
LM10	5.06	3.17	2.44	1.53	2.55	1.61

Table 4: The average distance and standard deviation on head images (256 × 192)

Table 3, and 4 show the comparation on average distances and standard deviations on body, and head part, respectively. The results of these parts are also the same with the result of pronotum part: the distances of fine-tuning processes are more improved than the result when training from the scratch.

5.2 Experiment on the second resolution dataset (96×96)

Landmarks	From scratch		With fine-tuning	
	Average	SD	Average	SD
LM1	**1.61**	**0.93**	**0.83**	**0.55**
LM2	1.84	1.12	0.92	0.73
LM3	1.62	1.00	0.88	0.62
LM4	1.88	1.21	0.86	0.64
LM5	1.76	1.10	0.90	0.65
LM6	**2.13**	**1.34**	**1.00**	**0.79**
LM7	1.61	1.02	0.81	0.64
LM8	1.98	1.27	0.94	0.82

Table 5: The average distance and standard deviation on pronotum images (96×96)

Landmarks	From scratch		With fine-tuning	
	Average	SD	Average	SD
LM1	**1.50**	**1.27**	**0.878**	**0.77**
LM2	1.54	1.32	0.82	0.80
LM3	1.56	1.21	0.82	0.70
LM4	1.45	1.31	0.81	0.78
LM5	1.52	1.36	0.89	0.75
LM6	**2.09**	**1.58**	**1.21**	**0.94**
LM7	2.27	1.69	1.06	1.01
LM8	2.41	1.67	1.08	0.95
LM9	2.33	1.55	1.21	0.84
LM10	1.59	1.21	0.85	0.72
LM11	1.54	1.24	0.83	0.70

Table 6: The average distance and standard deviation on body images (96×96)

Table 5, 6, and 7 show the average distances and standard deviations on each landmark on pronotum, body, and head part of beelte in the dataset with the size of 96×96. In this dataset, we have just considered the fine-tuning process without freezing because we do not see the significantly different between two fine-tuning processes in the case of the previous dataset. Like previous case of dataset 192×256, the average distances and standard deviations with fine-tuning have been improved than the results when training from scratch.

In addition, when we compare the results on two datasets, the distances have been reduced in the case of small resolution (96×96). Even in the case of training from the scratch, the distances are still smaller than the results of fine-tuning in the large resolution (192×256).

Landmarks	From scratch		With fine-tuning	
	Average	SD	Average	SD
LM1	**2.32**	**1.40**	**0.98**	**0.80**
LM2	2.00	1.36	0.88	0.62
LM3	2.18	1.41	1.00	0.86
LM4	1.90	1.40	0.91	0.68
LM5	1.93	1.26	1.19	0.96
LM6	**1.63**	**1.31**	**1.07**	**0.85**
LM7	1.77	1.18	0.89	0.86
LM8	1.57	1.15	0.85	0.76
LM9	1.92	1.22	0.96	0.91
LM10	1.83	1.20	0.92	0.83

Table 7: The average distance and standard deviation on head images (96×96)

6 Conclusions

In this study, we have proposed a CNN to predict the landmarks on beetle species. The model includes 3-repeated of an elementary block followed by 3 full-connected layers and a dropout layer. Each elementary block includes a convolutional layer, a maximum pooling layer, and a dropout layer.

The model has been trained and tested on two dataset with different size of images (256×192 and 96×96) by applying two stategies: training from scratch and fine-tuning. The results show that the model had a good prediction on beelte species by training from scratch. However, the quality of predictions have been improved when we have applied fine-tuning. In addition, when we consider the size of the input, the model has given the better result with the small size of images than the larger.

References

[1] Anne Sonnenschein, David VanderZee, William R Pitchers, Sudarshan Chari, and Ian Dworkin. An image database of drosophila melanogaster wings for phenomic and biometric analysis. *GigaScience*, 4(1):25, 2015.

[2] Celia Cintas, Mirsha Quinto-Sánchez, Victor Acuña, Carolina Paschetta, Soledad de Azevedo, Caio Cesar Silva de Cerqueira, Virginia Ramallo, Carla Gallo, Giovanni Poletti, Maria Catira Bortolini, et al. Automatic ear detection and feature extraction using geometric morphometrics and convolutional neural networks. *IET Biometrics*, 6(3):211–223, 2016.

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

[4] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 253–256. IEEE, 2010.

[5] Sander Dieleman, Jan Schlter, Colin Raffel, Eben Olson, Sren Kaae Snderby, Daniel Nouri, et al. Lasagne: First release., August 2015.

[6] Sasirekha Palaniswamy, Neil A Thacker, and Christian Peter Klingenberg. Automatic identification of landmarks in digital images. *IET Computer Vision*, 4(4):247–260, 2010.

[7] Van Linh LE, Marie BEURTON-AIMAR, Adrien KRÄHENBÜHL, and Nicolas PARISEY. MAELab: a framework to automatize landmark estimation. In *WSCG 2017*, Plzen, Czech Republic, May 2017.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[10] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[11] Tomáš Mikolov, Anoop Deoras, Daniel Povey, Lukáš Burget, and Jan Černockỳ. Strategies for training large scale neural network language models. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pages 196–201. IEEE, 2011.

[12] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*, 2014.

[13] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

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

[15] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[16] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3476–3483, 2013.

[17] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision*, pages 94–108. Springer, 2014.