Comparison of Facial Landmark Detection Methods for Micro-Expressions Analysis

Alexander V. Savin, Victoria A. Sablina, and Michael B. Nikiforov
Department of Electronic Computers
RSREU
Ryazan, Russia
sablina.v.a@evm.rsreu.ru

2021 10th Mediterranean Conference on Embedded Computing (MECO) | 978-1-6654-3912-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/MECO52532.2021.9460191

Abstract—**In the paper the problem of the facial landmark detection is investigated. This task is considered as the stage of the pipeline for micro-expression analysis. The classification of the known facial landmark detection methods is represented. The most promising methods are selected for the experimental comparison. These methods are based on the cascaded regression and on the deep learning. The known open source software implementations for them are chosen, viz. the OpenFace library and the MediaPipe framework. The experiments are carried out on the images from the Spontaneous Actions and Micro-Movements (SAMM) dataset. To evaluate and compare the obtained results for the two tested methods the reference image is constructed manually. Average deviations of the detected landmarks from the referenced are estimated. The comparison shows that a little better precision of the facial landmark detection is achieved using the deep learning based method.**

Keywords-facial landmarks; micro-expressions; Spontaneous Actions and Micro-Movements (SAMM) dataset; cascaded regression; deep learning; OpenFace; MediaPipe.

I. INTRODUCTION

Micro-expression analysis gained a lot of attention from the researchers all over the world recent ten years. Nevertheless, it was known in psychology many decades ago at most from the works of the eminent scientist Paul Ekman [1]. But only at present time the humanity obtained the tools for performing the micro-expression analysis automatically [2]. It is becoming possible due to the rapid development of the computer sciences disciplines and also the artificial intelligence. But still we can observe the lack of suitable methods and algorithms to solve the tasks of micro-expression analysis. There are also not many datasets for research work in this field. The possible advancement in understanding the human micro-expression could find its applications in many different areas. To name just a few they are psychology, criminology, security, business, education, entertainment, and even the everyday life. It is because micro-expression analysis could help to reveal the genuine emotions of the people.

According to the Paul Ekman researches there are seven main human emotions: happiness, fear, anger, sadness, disgust, contempt, and surprise. Also the neutral state without any emotion could be determined. When a person feels an emotion it becomes apparent on the face through macro- or micro-expression. The difference that the macro-expressions are more obvious and last longer but the micro-expressions are very subtle and typically lasts less than a half of a second. Because of that the usual macro-expressions are much more researched and the micro-expression analysis has many issues. Its methods are only developing nowadays. So it is indubitably drawn the research interest to this field. One of the best dataset from a very few available is the Spontaneous Actions and Micro-Movements (SAMM) dataset [3]. So it is decided to carry out the experiments on the images from this dataset.

Micro-expression analysis has two main branches: spotting and recognition. The first is to spot when the micro-expression occurs in the image sequence and the second is to recognize what emotion appears in the image sequence. The used version of the SAMM dataset is for experiments for micro-expression recognition. In either cases the solution could be represented by a pipeline consisted of several main stages. E.g. for micro-expression recognition such stages could be: face detection, facial landmark detection, feature extraction, classification of action units, and emotion recognition. In this research work we considered the facial landmark detection methods. The precise landmark detection is very important because the result of this stage could influence a lot on the correctness of the finally recognized emotion. In our previous research works we considered and compared the approaches to the face detection and simple facial landmark detection methods [4, 5]. In continuation of the research in this paper we investigate and compare more advanced facial landmark detection methods.

II. RELATED WORK

The known facial landmark detection methods could be divided into three following main categories: holistic methods, Constrained Local Model (CLM) methods, and the regression-based methods [6]. These categories differ in the ways of using the facial appearance and shape information. The holistic methods explicitly represent the global facial appearance and shape. The CLM methods explicitly use the global shape model but build the local appearance models. The regression based methods implicitly describe the facial appearance and shape.

Formally, given a facial image denoted as I, a landmark detection algorithm predicts the location of D landmarks $\mathbf{x} = \{x_1, y_1, x_2, y_2, \dots, x_D, y_D\}$, where x_i and y_i, $i = \overline{1, D}$, represent the image coordinates of the facial landmarks. Facial landmark detection is still a challenging problem though many methods and algorithms are proposed.

978-0-7381-3361-4/21/$31.00 ©2021 IEEE

The classical holistic method is the Active Appearance Model (AAM) [7]. It is a statistical model that fits the facial images with a small number of coefficients. In this method the global facial shape and appearance models are based on Principal Component Analysis (PCA). Most of the holistic methods focus on improving the fitting algorithm.

The CLM methods are more robust to illumination and occlusion in comparison with the holistic methods. The most popular is the Active Shape Model (ASM) [8]. Our result of using the ASM method for the SAMM dataset is presented in [5]. It is noticed that for the micro-expression analysis pipelines the more precise facial landmark detection is needed.

The regression-based methods directly learn the mapping from the image appearance to the landmark locations and do not explicitly build any global shape model. These methods are considered more promising compared to the methods of the previous two categories. The regression-based methods could be classified into direct regression methods, cascaded regression methods, and deep learning based regression methods [9, 10]. Cascaded regression methods achieve better results than direct regression. Cascaded regression with deep learning can be even more effective but required a large dataset for training. So in this work we decided to compare the two methods reported as most promising: cascaded regression based and deep learning based pre-trained for facial images.

III. METHODS

The facial landmark detection algorithm based on the cascaded regression method is described in [11]. This algorithm is implemented in the dlib library. We selected this implementation for experiments on the SAMM dataset.

A cascade of regression functions could be utilized for the facial landmark detection. In the considered algorithm each regression function in the cascade efficiently estimates the shape from an initial estimate and the intensities of a sparse set of pixels indexed relative to this initial estimate. Each regressor is leant via gradient boosting with a squared error loss function. The sparse pixel set, used as the regressor's input, is selected via a combination of the gradient boosting algorithm and a prior probability on the distance between pairs of input pixels. The cascade of such regressors can detect the facial landmarks when initialized with the mean face pose. The initial shape can be chosen as the mean shape of the training data centered and scaled according to the bounding box output of a generic face detector. This cascaded regression algorithm is a machine learning algorithm. The algorithm is already pre-trained.

The facial landmark detection model based on the deep learning method is described in [12]. This model is implemented in the TensorFlow library. We also selected this implementation for experiments on the SAMM dataset.

The input to the model is an image, and the output is a face mesh prediction which is comprised of the detected facial landmarks. The residual neural network architecture is used for the mesh prediction model. The most of the computations are performed by its shallow part. The neurons receptive fields start covering large areas of the input image relatively early. When such a receptive field reaches the image boundary its relative location in the input image becomes implicitly available for the model to rely on. Then the neurons for the deeper layers are likely to differentiate, e.g., between mouth-relevant and eye-relevant features. This model is based on the deep learning architecture and also already pre-trained.

IV. EXPERIMENTAL RESULTS

A. Software implementations used in the experiments

The main problem considered in this research work is the facial landmark detection. The experiments on the basis of the two described methods are carried out. And then the obtained results are compared. The first method is based on OpenFace and the second on MediaPipe. Both are open-source software.

OpenFace is a set of libraries, models and methods for the detection, tracking the position, identification of the faces on images using the deep neural networks [13]. OpenFace provides a set of pre-trained models of neural networks. To solve the facial landmark detection problem OpenFace uses the OpenCV library to detect the face. Further the search for the facial landmarks on the detected face is performed using the dlib library implementing the method based on the cascaded regression [11]. OpenFace makes it possible to detect 68 facial landmarks. These landmarks are presented in Figure 1. In OpenFace they are used for face alignment as preprocessing to prepare the image to input into the neural network.

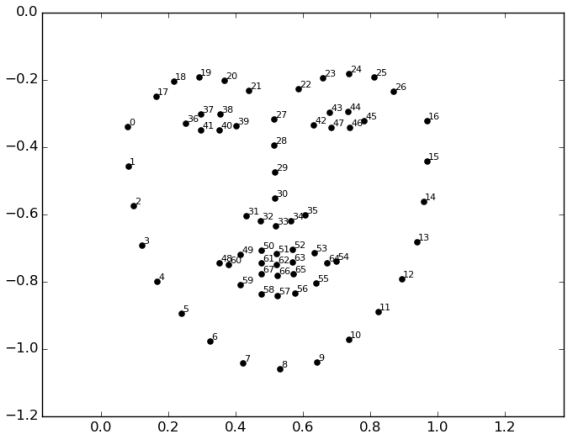

Figure 1. 68 landmarks which could be detected using OpenFace.

MediaPipe is also a set of libraries, pre-trained models and methods for solving different kinds of problems. It could be such problems as face identification in the image, facial landmark detection, body pose tracking, and object recognition. If a problem requires machine learning to be solved then MediaPipe uses TensorFlow. The solution of the problem by MediaPipe leads to the construction of the pipeline for media information (video flow) processing. There are the ready pipelines to solve the widespread video processing problems, e.g., the facial landmark detection problem. MediaPipe makes it possible to detect 468 facial landmarks which are arranged in fixed quads and represented by their coordinates (x, y, z) [14]. The mesh topology comprised by these landmarks is presented in Figure 2.

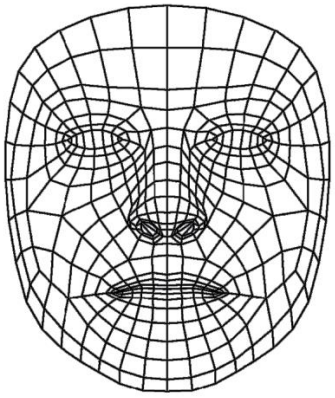

Figure 2. 468 landmarks in the mesh topology detected by MediaPipe.

B. *Description of the experiments*

To carry out the experiments the SAMM dataset is used. It consists of the image sequences of the faces with the spontaneous micro-expressions. The set contains 159 spontaneous facial micro-expressions image sequences of people belonging to different gender, race, and nationality groups. This set is prepared by the research group from the Manchester Metropolitan University [3]. The examples of the images from the SAMM dataset are presented in Figure 3.

Figure 3. Example images from the SAMM dataset.

The special software is developed for the experiments. In general the main stages of the experimental research are represented in Figure 4.

The input image is provided as the input for the developed subroutines. These subroutines use OpenFace and MediaPipe. The result of this stage is an array of the found landmarks, i.e. their coordinates. Besides that, the result of this stage is an image with the found landmarks. This image serves for visualizing the landmark disposition on the face for the further analysis. Examples of this stage visualization results are shown in Figure 5. The result for OpenFace is in Figure 5, a. The result for MediaPipe is in Figure 5, b.

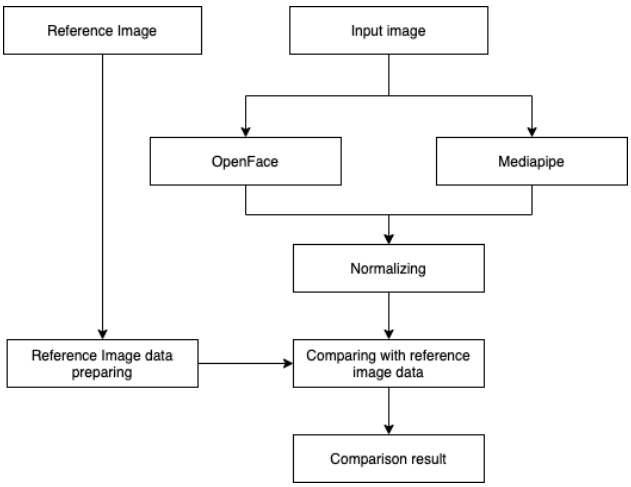

Figure 4. The main stages of the experiments.

a b

Figure 5. Examples of the output results: a – for OpenFace; b – for MediaPipe.

The next stage is normalizing the obtained results. The process of normalizing is required to transform the data to the uniform standard representation for the further comparison. As OpenFace initially detects 68 facial landmarks and MediaPipe detects 468 facial landmarks, normalizing of the data is required to compare these two methods. So the landmarks detected by MediaPipe appropriate for the comparison with the landmarks detected by OpenFace are selected from the set of 468 landmarks. To compare the obtained data some etalon is required. The same input image which is initially processed could serve as such etalon after the manually labeling the facial landmarks on it. This labeled image could be called the reference image. These landmarks on the reference image are manually disposed very close to each other. Also the labeled landmarks are connected by lines. So these lines highlight the significant parts of the face such as the facial contour, the nose line, the eyes, and the eyebrows line. An example of the reference image labeling is presented in Figure 6.

Figure 6. An example of the reference image labeling.

The reference image is loaded by the developed program. Then the reference image data preparing is performed. All the points of the lines highlighting the facial parts on the reference image are taken.

The data comparison is carried out as follows. The next landmark obtained by OpenFace is selected. Its coordinates are compared with the taken points of the reference image. For all such reference image points p_r with the coordinates (x_r, y_r) and the selected landmark p_i with the coordinates (x_i, y_i) the distance is calculated by the formula (1).

$$S_{ir} = \sqrt{(x_r - x_i)^2 - (y_r - y_i)^2}. \quad (1)$$

The minimal distance from the calculated distances is chosen by the formula (2).

$$S_{imin} = \Delta_i = \min_r(S_{ir}). \quad (2)$$

This minimal distance is assumed as the deviation indicator between the landmarks detected using OpenFace and the points from the reference image. After the deviation indicator is calculated for each landmark detected on the source image the average deviation is found by the formula (3).

$$\Delta = \frac{\sum_{i=0}^{n} \Delta_i}{n}. \quad (3)$$

The same comparison is done for the MediaPipe landmarks.

As a result, for each analyzed image the average deviation from the reference image is found for the methods using OpenFace and MediaPipe. These resulted average deviations for several images are represented in Table 1.

TABLE I. THE RESULTED AVERAGE DEVIATIONS.

№	OpenFace	MediaPipe
1	3.3257	3.2608
2	5.1518	3.6937
3	3.2096	2.8792
4	3.5296	4.1308
5	4.0357	3.9769

After analyzing the obtained results it could be concluded that in the most cases MediaPipe gives the less deviation from the reference disposition of the landmarks than OpenFace. Moreover, MediaPipe framework detects 468 landmarks as opposed to OpenFace which detects only 68 landmarks. The more landmarks make it possible to analyze more data when considering the problems of micro-expressions spotting and recognition. MediaPipe is developed and supported by forces of the engineers from the Google company and a huge community of developers. At the same time the development and support of OpenFace at present is on the way out that could be observed from the OpenFace GitHub page [15].

V. CONCLUSION

The comparison of the experimental results leads to the following conclusion. The two methods of the facial landmark detection are examined: based on OpenFace and MediaPipe.

OpenFace uses the cascaded regression method, and MediaPipe uses the deep learning method. The experiments carried out for the images from the SAMM dataset show that in average MediaPipe detects facial landmarks more precise than OpenFace. So in the future work it is preferable to apply the deep learning method at the facial landmark detection stage of the overall pipeline for micro-expression analysis. The more precise results at this stage could increase ultimately the correctness of the human hidden emotion recognition.

REFERENCES

[1] Paul Ekman, Emotion in the Human Face, 2nd Edition, Malor Books, 2013, 456 p.

[2] Yee-Hui Oh, John See, Anh Cat Le Ngo, Raphael C.-W. Phan, and Vishnu M. Baskaran, "A Survey of Automatic Facial Micro-Expression Analysis: Databases, Methods, and Challenges," Frontiers in Psychology Journal, Volume 9, Article 1128, 2018, 21 p.

[3] Adrian K. Davison, Cliff Lansley, Nicholas Costen, Kevin Tan, and Moi Hoon Yap, "SAMM: A Spontaneous Micro-Facial Movement Dataset," in IEEE Transactions on Affective Computing, Volume 9, No. 1, 2018, pp. 116-129.

[4] Anna D. Sergeeva, Alexander V. Savin, Victoria A. Sablina and Olga V. Melnik, "Emotion Recognition from Micro-Expressions: Search for the Face and Eyes," 8th Mediterranean Conference on Embedded Computing (MECO) Proceedings. Budva, Montenegro, 2019, pp. 632-635.

[5] Anna D. Sergeeva and Victoria A. Sablina, Eye Landmarks Detection Technology for Facial Micro-Expressions Analysis, 9th Mediterranean Conference on Embedded Computing (MECO) Proceedings. Budva, Montenegro, 2020. Pp. 448-451.

[6] Yue Wu and Qiang Ji, "Facial Landmark Detection: A Literature Survey," International Journal of Computer Vision, Volume 127, Issue 2, February 2019, pp.115-142.

[7] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor, "Active Appearance Models," IEEE Trans. PAMI 23, 2001, pp. 681-685.

[8] Timothy F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim Graham, "Active Shape Models – Their Training and Application," Computer Vision, Graphics and Image Understanding, 1995, pp. 38–59.

[9] Yi Sun, Xiaogang Wang, and Xiaoou Tang, "Deep convolutional network cascade for facial point detection," In IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3476-3483.

[10] Olga V. Melnik, Victoria A. Sablina, Alexander V. Savin, and Alexey B. Borschev, "Detection of Anthropometric Face Points Based on Deep Learning Methods to Recognize Emotions," Biomedical Radioelectronics Journal, Volume 23, Issue 3, Moscow, "Radiotekhnika" Publishing House, 2020, pp. 45-52 (in Russian).

[11] Vahid Kazemi and Josephine Sullivan, "One millisecond face alignment with an ensemble of regression trees," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1867-1874.

[12] Yury Kartynnik, Artsiom Ablavatski, Ivan Grishchenko, and Matthias Grundmann, "Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs," CVPR Workshop on Computer Vision for Augmented and Virtual Reality 2019, IEEE, Long Beach, CA, 4 p. [accessed 2021 March 12], https://arxiv.org/abs/1907.06724.

[13] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan, "Openface: A General-Purpose Face Recognition Library with Mobile Applications", CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016, 18 p.

[14] MediaPipe on GitHub, [accessed 2021 March 12], https://google.github.io/mediapipe.

[15] OpenFace on GitHub, [accessed 2021 March 12], https://github.com/cmusatyalab/openface.