An Investigation on Face Detection Applications

Mohammad Almasi
Department of Advanced Mathematics and Mathematical Engineering
University Polytechnic of Catalonia, Barcelona, Spain

ABSTRACT

Face detection in uncontrolled ambiance continues to be a challenge to traditional face detection methods due to the large difference in facial expressions. The task of alignment is particularly difficult when the face comes from an extremely unconditional environment. To overcome these problems, the present paper is a reliable and deeply considerable application that allows users to detect a face(s) in real-time and process them. Database of faces with bounding rectangles and facial landmark locations is collected, and simple discriminative classifiers are learned from each of them. Then methods for morphing, warping, swapping and averaging of faces are presented. As a result, faces can be very effectively detected, aligned/ oriented and used for the above methods. In addition, based on results this approach can detect faces and eyes in difficult conditions without explicitly simulating their variation. Evaluating the tests of the application, it can confidently said that this face detection method is accurate and effective and reaches the most modern level of performance. The same methodology can easily be generalized to other tasks, as well as the detection of a common object.

General Terms

Recognition, Security, Face detection, Image Processing

Keywords

Face detection, Facial landmark, detection and recognition

1. INTRODUCTION

The task of detecting, recognizing and processing faces has many applications in areas such as biometrics, computer vision systems in robotics, intelligent security, and access control systems. Face identification technology unlike the use of other biometric indicators (fingerprint, iris) does not require physical contact with the device and considering the rapid development of digital technology, it is the most acceptable for mass uses. Given that it is the least invasive and people use facial identification in their daily lives, the technology has been on the rise and its optimization is a growing field of interest [1] - [3].

Many different detecting, recognizing and processing faces algorithms have been developed in recent years, such as neural networks, personal faces, Markov chains, etc. All face recognition systems fall into two broad categories: systems using two-dimensional (2D) images and systems using three-dimensional (3D) images. When using 2D images, the recognition quality is affected by the position of the faces in the image and the light conditions, while the 3D images of persons

Are designed to remove these limitations. However, the acquisition of 3D images is done using special devices. In addition, such systems require that the object is stationary for several seconds during scanning, which is unacceptable for systems operating in real-time mode (human-computer interaction, object-oriented compression of video data, video surveillance, etc.) [4] - [5]. The task of detecting and recognizing faces is complicated due to several reasons: a person is a dynamic object having a high degree of variability. For example, in the shape of the face and colour of the skin; various lighting conditions, determined by the type, direction and number of light sources; partial overlapping of faces by other objects of the scene; the necessity of localization and recognition of persons having arbitrary positions in space [6] - [8]. There are a lot of methods for detecting the face and the point or region of interest (ROI) and processing them. The most commonly used method is facial landmarks, which allows to localize and represent salient regions of the face. Facial landmarks have been successfully applied to face alignment, head pose estimation, face swapping, face morphing and much more. Detecting facial landmarks is a subset of the shape prediction problem. Given an input image, a shape predictor attempts to localize key points of interest along with the shape. In the context of facial landmarks, the goal is to create a Graphical User Interface application in python to detect the face(s) in real-time. The task of facial landmark localization is to obtain the coordinates of a set of pre-defined facial landmarks. These facial landmarks usually have specific semantic meanings, such as the nose tip and eye corners, and are instrumental in enabling the subsequent face image analysis. The task can be accomplished using a few different techniques, such as face and eye alignment. Many face alignment methods have been proposed in recent years. The Regression-based methods and deep-learning-based methods are the most popular methods. The regression-based methods regress the face landmark explicitly by learning the features in the image, e.g. [9] use random forest [10] to achieve good performance with local binary features, [11] regresses features with random ferns. Convolution networks have also been used in face alignment tasks successfully, [12] detects face landmarks with three-level networks. Multi-task learning [12] has been used in many computer vision tasks successfully. In [13] proposes the method of TCDCN which first introduced multi-task learning into facial landmark detection. But TCDCN only works for five points with the auxiliary task of head pose estimation, gender classification and age estimation. Some tasks are not closely related to face alignment, in other words, the age doesn’t have much relationship with where the facial points located in. Besides this, the most related task pose estimation only output right, left, or frontal but not the real angle. Focus on these problems removed the irrelevant tasks, and describe the pose by using three float numbers, which represent roll, yaw, and pitch of ahead to build the network. In addition, many studies are concentrated on the initial problem. As discussed in [14] to use different initializations and take the mean results as the final pose. Also [15] solved the initialization problem by predicting the rough estimation from a global image patch. In [16] was also presented a unified model for face detection, pose estimation, and landmark estimation in real-world, cluttered images. Their model is based on a mixture of trees with a shared pool of parts; they model every facial landmark as a part and use global mixtures to capture topological changes due to viewpoint. They show that tree-structured models are surprisingly effective at capturing global elastic deformation while being easy to optimize, unlike dense graph structures. Face detection is responsible for segmenting the face, or more specifically a face area, from the background. By aligning the faces, the aim is to more precisely pinpoint face location and also to normalize the faces as data for the next stages. As such, detection and alignment work in tandem to provide estimates of the location, and scale, of faces, detected in the input data. Based on this it is possible to extract facial components, including the mouth, nose, eyes, and the jawline. The facial landmark points make up the identifying points of a face, for a computer, allowing for the use of faces as data. Z. Zhang in 2014 describes the importance and challenges of facial landmark detection, an integral part of facial analysis tasks, face verification, and face recognition. Even though there has been performed considerable amounts of work in facial landmark detection, Zhang et al. argue that a robust solution still remains to surface. Some of the challenges include partial face occlusion and considerable head pose variations [17]. Historically, there have been two main categories for detection, this being regression-based, and template fitting methods. Where regression-based methods rely solely on landmark estimation by regression using image features, template fitting methods build face templates to fit the input images in to. Another approach is by using cascaded Convolutional Neural Networks (CNNs). The cascaded CNNs requires faces to be divided into separate parts, where each part is handled in turn by its own deep CNN. Outputs are averaged and passed on to cascaded layers where every facial landmark is estimated individually [17]. The average face is a concept that has been of interest in several disciplines. It has been subject to much debate within psychology, where several studies have shown that computationally averaged faces are generally regarded as more aesthetically pleasing [18], [19]. This phenomenon is often credited to the fact that through averaging, individual imperfections and asymmetry are watered down. Koinophilia, an evolutionary hypothesis claims that an average looking individual is more often preferred as a mate as it is less likely to be subject to undesirable mutations within a species [20]. The first average face dates back to 1878 when Francis Galton created a new technique for compositing faces in the development of photography. By aligning the eyes of several face images and exposing them on the same photography plate, Galton managed to create a new face, the composite face, which combined all the original faces ([21]; Galton 1878[22]). The composite technique had its resurgence in the 1990s when computers could take over these operations [23]. Face morphing is the process of creating a fluid transition between two faces. This transition is actually a series of images, comparable to the frames of a video, of differing alpha blending. This blend determines the relationship of pixel intensity between the two images, and by parameterizing this alpha value it is possible to decide which face is to be more dominant in the end result. The process of face morphing is very similar to the process of creating average faces and uses several of the same operations. By using Delaunay Triangulation, it is possible to create corresponding triangles that can be transformed and warped from one face onto the other using the concept of affine transformation as mentioned previously. Finally, the warped faces can be alpha blended using the alpha blend parameter. The result will then be a morphed face which is a combination of the two faces, where the given alpha value decides which face is more dominant [23]. The concept of face swapping also uses facial landmark detection, face alignment, Delaunay Triangulation, and affine warping. Given the detected landmarks, the convex hull (the smallest convex set of points that contains all other points) of one face is aligned on top of the other, and potentially vice versa. By using Delaunay Triangulation and affine transform the triangles of the faces are warped to match their destination face. However, the process is not finished here, as an essential operation remains. Seamless cloning is an implementation based on the findings from the paper "Poisson Image Editing" by Perez, Gannet and Blake from 2003. The paper argues that it is beneficial to work with image gradients as opposed to image intensities as a means to achieve more realistic results when performing cloning. Seamless cloning makes the warped face blend with the destination face by altering aspects of the face like texture, illumination, and color. This entire process will result in the destination face now having a different facial appearance, but approximately the same photometrical qualities as before the swap [23] - [25].

2. METHOD

It all started with Francis Galton who, back in 1878, came up with a new photographic technique for compositing faces by aligning the eyes. He thought that by averaging faces of criminals he could create the prototypical criminal face which in turn would help predict if someone is a criminal based on their facial features. His hypothesis turned out to be wrong; you cannot predict if a person is a criminal by looking at their photos. However, he noted that the average face was always more attractive than the faces it was the average of several researchers in the 1990s showed that people find facial averages much more attractive than individual faces. In one amusing experiment researchers averaged the faces of 22 Germany finalists of 2002. People rated the average face to be much more attractive than every one of the 22 contestants, including Berlin who won the competition shouldn't an average be mediocre by definition. The question here is, what the importance of finding an average face and why it's attractive. According to an evolutionary hypothesis called Koinophilia, sexually reproducing animals seek mates with average features because deviations from the average could indicate disadvantageous mutations. An average face is also symmetric because the variations in the left side and the right side of the face are averaged out [26]. Face detection can be regarded as a specific case of object class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belongs to a given class. In other words, face detection is the process of detect features of a face and locating it in a digital image/video using predefined features. OpenCV uses classifiers to test the images based on features. These features are extracted by training the classifier with a large set of positive as well as negative images. Positive images are those that actually contain the object that is needed to detect. Once the classifier training finishes, features will be extracted from it. The OpenCV face detection algorithm uses the concept of Cascade of Classifiers. Because of the fact that the face in the image takes up only a small region, and most of the image region is the non-face region, this concept saves up a lot of processing. Instead of applying the large set of all features on a window, it is done in different stages. If the window fails the first stage, it will stop testing it for the next stage. If the window passes all the stage tests, it is identified as a face in OpenCV face detection. For each facial image 68 facial landmarks calculation using Dlib has done. Facial landmarks can be used to align facial images to mean face shape so that after alignment the location of facial landmarks in all images is approximately the same [27]. The input facial images can be of very different sizes. Therefore, it will need a way to normalize the faces and bring them to the same reference frame. To achieve this, warping the faces to an image such that the left corner of the left eye is at pixel location and the right corner of the right eye is at the pixel location. Let us call this coordinate system the output coordinate system and the coordinates of the original images of the input coordinate systems. How were the above points chosen? It will be wanted to make sure the points were on a horizontal line, and the face was centered at about a third of the height from the top of the image. So the corners of the eyes have chosen to be at (0.3 x width, height / 3) and (0.7 x width, height / 3). It will also know the location of the corners of the eyes in the original images; they are landmarks 36 and 45 respectively. That can be used to calculate a similarity transform (rotation, translation, and scale) that transforms the points from the input coordinate systems to the output coordinate system. What is a similarity transform? A similarity transform is a 2×3 matrix that can be used to transform the location of points or an entire image. The first two columns of this matrix encode rotation and scale, and the last column encodes translation (i.e. shift). Let's say you want to transform (move) the four corners of a square so that the square is scaled in the x and y direction by and respectively. At the same time, it is rotated by an angle and translated (moved) by and in the x and y directions. The similarity transforms for this can be written as [28].

$$S = \begin{bmatrix} s_x cos\,\alpha & \sin\alpha & tx \\ -\sin\alpha & s_y \cos\alpha & t_y \end{bmatrix}$$

Given, a point, the above similarity transform, move it to point using the equation given below:

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} s_x \cos\alpha & \sin\alpha \\ -\sin\alpha & s_y \cos\alpha \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

To find the similarity transform that will transform the points from the input image coordinates to the output image coordinates, estimate Rigid Transform has been used. There is one little problem though. OpenCV requires you to supply at least 3 points pairs. This is unnecessary because you can calculate a similarity transform using just two points. The good news is that it can simply hallucinated a third point such that it forms an equilateral triangle with the two known points and then use estimate Rigid Transform as if it had three points' pairs. Once a similarity transform is calculated, it can be used to transform the input image and the landmarks to the output coordinates. The image is transformed using warp Affine and the points are transformed using transform. From the previous step were able to transform all the images and the landmarks to the output image coordinates. All the images are now of the same size, and the two corners of the eyes are aligned. It may be tempting to obtain the average image by averaging the pixel values of these aligned images. However, if you did this, you will end up with an image shown on the left. Sure, the eyes are aligned, but other facial features are misaligned. If it was known, which point in one input image corresponded to which point in another input image it could be easy to align the two images perfectly. However, it's not possible. And the locations of 68 corresponding points in the input images is known. S will use these 68 points will used to divide the images into triangular regions. And align these regions before averaging pixel values. Now for Calculation the Mean Face Points: To calculate the average face where the features are aligned, first it needs to calculate the average of all transformed landmarks in the output image coordinates. This is done by simply averaging the x and y values of the landmarks in the output image coordinates.

Also for calculating Delaunay Triangulation: In the previous step it will obtain the landmark locations for the average face in the output image coordinates. Delaunay triangulation allows breaking the image into triangles. The result of Delaunay triangulation is a list of triangles represented by the indices of points in the 76 points (68 face points + 8 boundary points) array. In the previous step, the average location of facial landmarks has been calculated and used these locations to calculate a Delaunay triangulation to divide the image into triangles [29].

3. RESULTS AND DISCUSSION

As demonstrated above this project is the overall summary of the main important and useful features and application of face detection. Each section will illustrate a specific state of the art application of using computer vision in terms of face detection that can be used in security or any other type of technologies related to face understanding [31, 32]. The Most Important and under developing Ideas in face detection world, that is used and described in this project can be named as, face detection, eye detection, face swapping, face morphing and face averaging also motion detection can be categorized in the face detection usage and applications [33]. As discussed above in this paper Python has been used. And also designing and appearance of the Software has been evolved by QT creator. As shown in figure 1 the main page of the software will give the abstract information about the main features and rules in software. Also as can be seen the important parts of software will be shown and a brief idea of the main features and applications in the software [34]. It should be noticed that the "Help" section is directly accessible in the top left corner. As can be seen in Second Tab five main applications are set up for different approaches. Each Section will demonstrate its subsections.

In first section the face and Eyes and motion detection will be presented, as shown in figure number 2, the main appearance of Detection application has two main figures for showing the results and setting the references, as can be seen here the left one is for setting the reference for motion detection and face Detection, and the right figure is responsible for presenting the results. It should be attentional, any of these sections need an installed camera on your own machine. So before everything please set your camera up.

In detecting section changing the Target between Eyes, and face is handled by checkboxes that are used in middle menu of detection part view as shown. Also in this section four pushbuttons are responsible for changing the algorithms and tasks namely as Webcam start and stop for running and stopping the camera as the main image acquisition hardware. Setting the reference that will be used for motion detection approach, by setting the reference position and gesture, this application will be used in liable button for motion detection, motion detection is mainly used for detecting any movements and changing between frames regarded to their pixels [35]. And the Detect button will be used for detection from any checkboxes that selected between eyes or face. In this project, faces are detected by the haar cascade method [36]. Furthermore, for detecting motions there will be no difference between faces and any other type of moving object, on the other hand for Eyes, face and landmarks detection, any faces should be detected firstly by the camera. This means that it has a priority to a task.

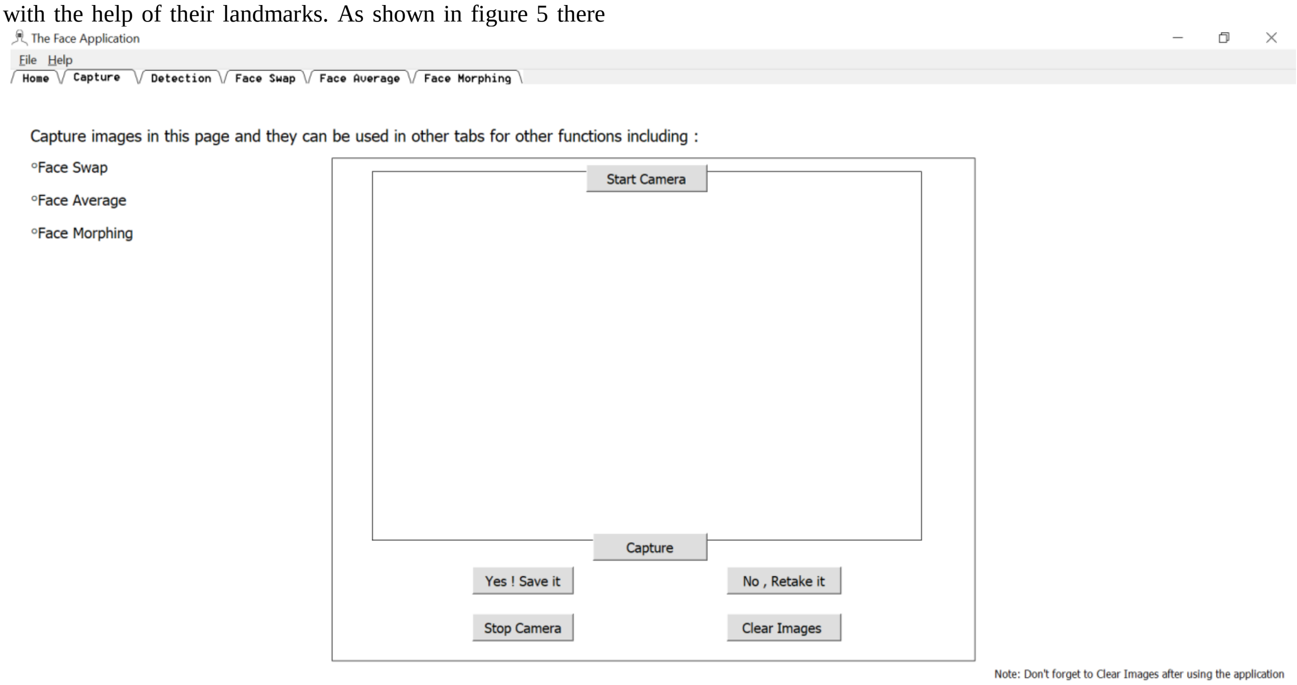
For developing and swapping the faces 68 landmarks shape detection have been used [37]. The results of 68 point algorithm can be done. It should be noticed that this method is different from static face detection [37]. This part is used for swapping faces. As can be seen in figure 3 the two upload push buttons are responsible for loading the original screenshot captured in the previous step. Continuing the results of swapping between faces will be displayed on the pined figure and the results will show the swapping of the first face and the second one. It means each vector that is responsible for each value in landmarks will be swapped by the other one [37].

The average part is used to show one of the most useable applications in face detection, as shown in figure 4, in this approach basically, from the values of each landmark for each face the average face will be calculated [38]. One of the best approaches for this applicant is principal components analysis (PCA) in this method the average can be calculated using Eigenfaces from input faces [38]. Same as face swapping, here 68 landmark shape detection will be used, but for calculating the average of all faces in the images. It should be noticed this part has been conducted both with images and in real-time, which means that the faces landmarks and the average will be calculated and shown by the camera recording online and for preprocessing for removing blur noises, the Laplacian filter is the best choose to decline the dimness and make face and results in more sharpness [39].

The morphing part will changed two faces chronologically with the help of their landmarks. As shown in figure 5 there will be 3 figures for presenting results, the two left ones will be used for capturing the faces and as the inputs or references, and the right one is for the results. It should be noticed that face morphing with this approach cannot be done in real time. The main difference between face swapping and face morphing is in their approaches to use landmarks. In first method the whole face will be changed, while in the second approach the values of each vector in landmarks will be changed one by one [40].

4. CONCLUSION

The aims of this paper is to demonstrate the hot topics in face detection algorithm and application. From the many research that has been carried out, it is worth pointing out that incorporating among face detection in terms of morphing, averaging, and swapping not only can be done in entertainment approaches but also has many usage in security or other serious application, so face detection and recognition is being under developing. In this paper python was used for developing the codes and QT creator has been evolved for designing the interface of the software. Furthermore as discussed above the experimental results have illustrated the effectiveness of potential face detection algorithms, but still developing is continued likewise noise removal for face averaging section with namely Laplacian filter that can be used for improving the results. Also, it is too obvious the average face is so exciting but average facial configurations can be done too. in addition in all section it so important to distinguish faces while they have overlapped to each other.



Capture images in this page and they can be used in other tabs for other functions including :

°Face Swap

°Face Average

°Face Morphing

Note: Don't forget to Clear Images after using the application

Figure1. First page of Application. As can be seen above in the top taskbar "File" and "Help" scroll menus are used, also five main tab are trying to show each part.

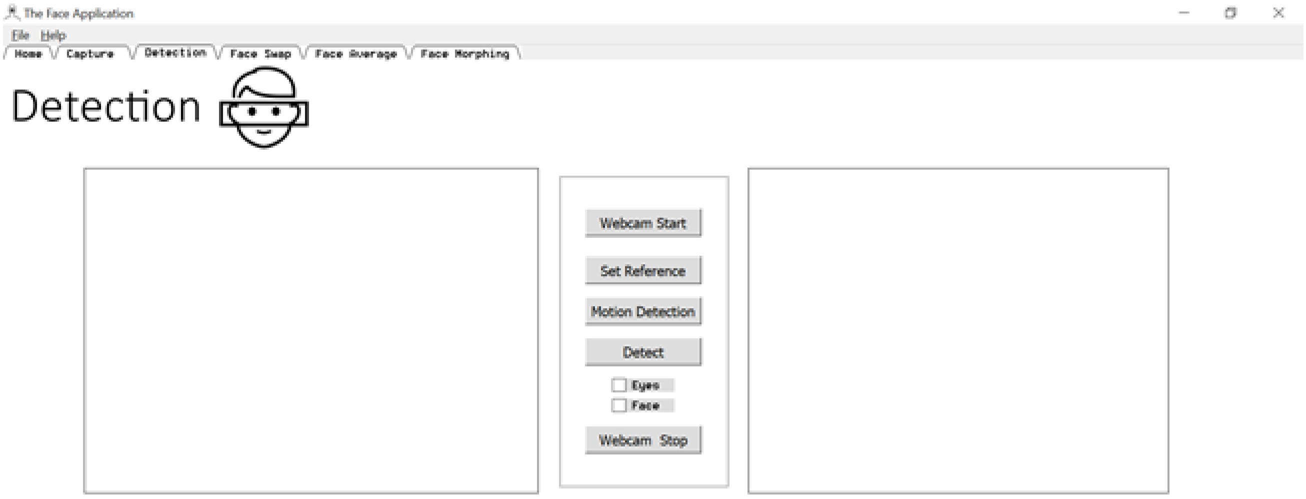


Figure2. The appearance of detection part. Two check boxes are used to make change between eyes and face detection.

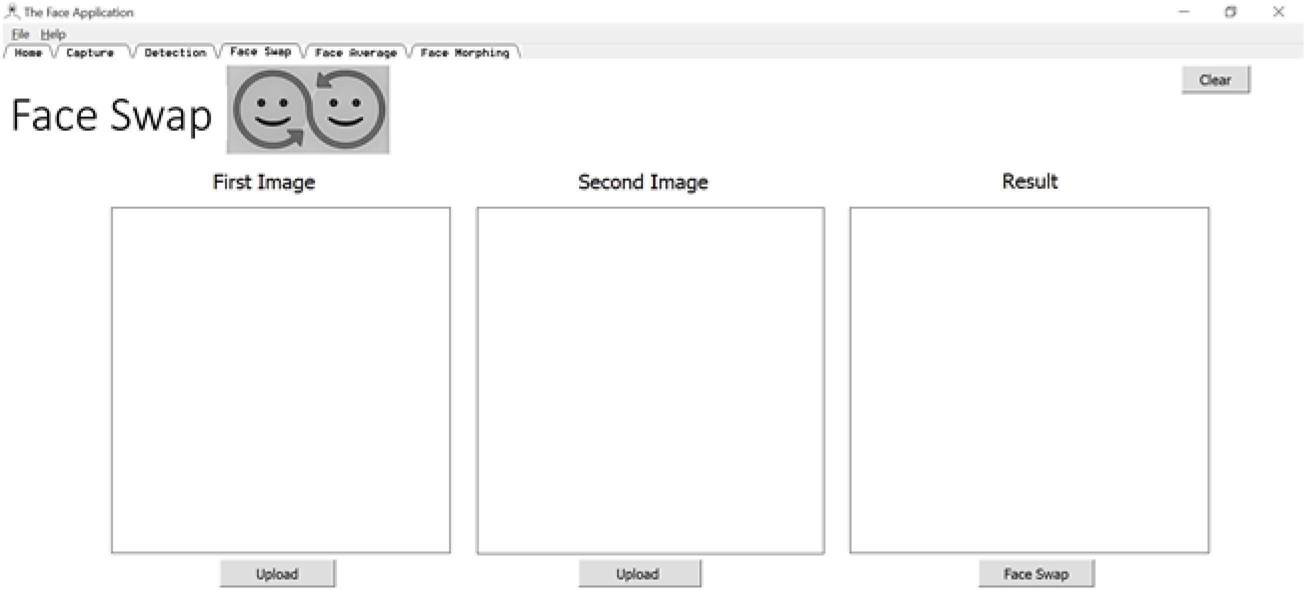


Figure3. The appearance of the swapping part with three main figures to handle the procedure.

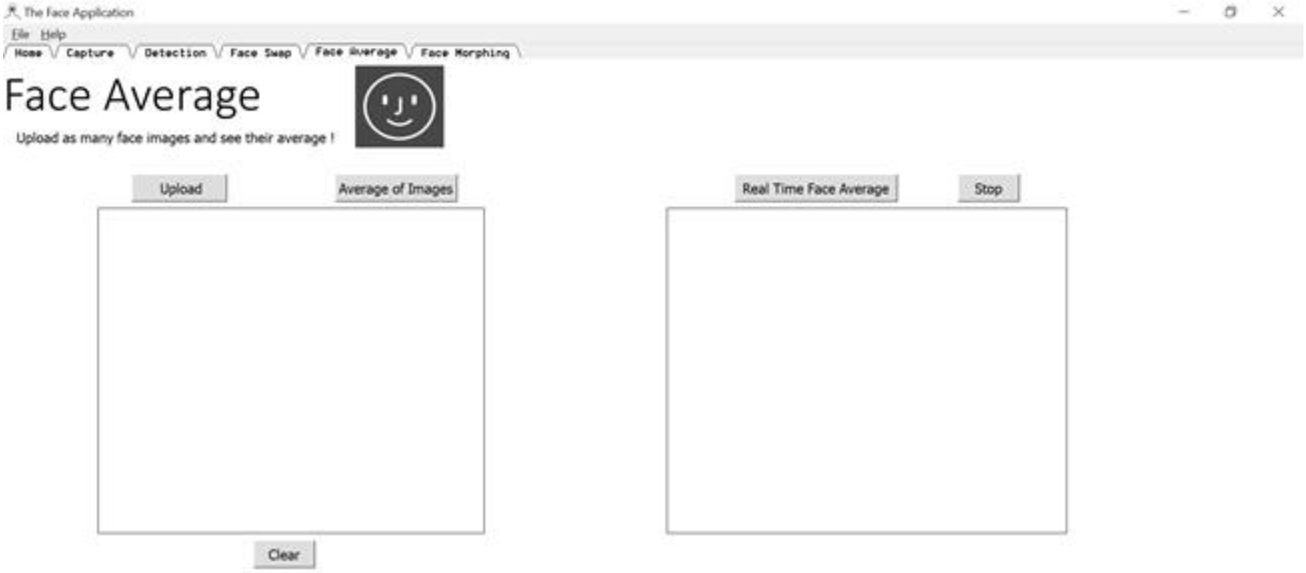


Figure5. The appearance of swap part of software

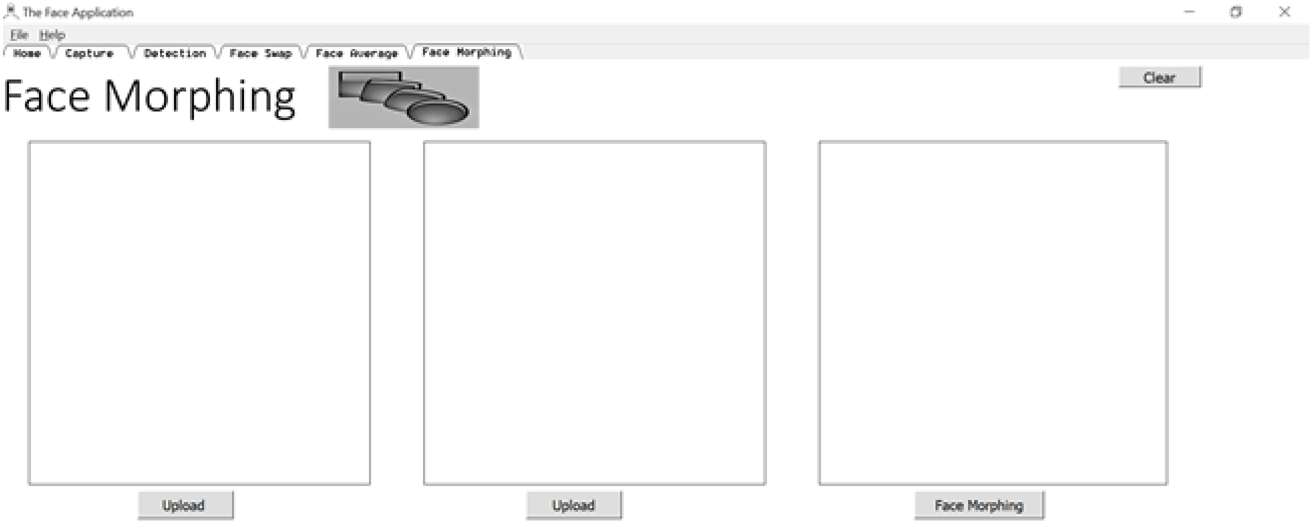


Figure5. The appearance of face morphing part of software

5. REFERENCES

[1] A. Pakhirka, Thesis for a scientific degree: Computer method of face localization in images in difficult lighting conditions; p.p. 8-10, 2011.

[2] A.K. Jain, R. Bolle, and S. Pankanti, eds., Biometrics: Personal Identification in a Networked Society, Kluwer Academic Publishers, 1999.

[3] A Krenker, J Bešter and A Kos," Introduction to the Artificial Neural Networks", Edited Kenji Suzuki, Published by InTech,, Janeza Trdine, Croatia, p.p. 3-18, 2011.

[4] H. A. Rowley, S. Baluja and T. Kanade, "Neural Network-based Face detection", IEEE Trans. On Pattern Recognition and Machine Intelligence, 20(1),23-28, 1998.

[5] W.Zhao, R.Chellapa, P.J.Phillips, and A.Rosenfeld, "Face Recognition: A Literature Survey," Technical Report CART-TR948. University of Maryland, Aug.2002.

[6] W. Zhao, R. Chellappa, A. Rosenfeld, and P. Phillips. Face recognition: A literature survey. ACM Computing Surveys; p.p. 399–458, 2003.

[7] Er.P. Kumar, Er.P. Sharma, "ARTIFICIAL NEURAL NETWORKS- A Study", International Journal of Emerging Engineering Research and Technology, vol. 2, no. 2, p.p. 143-148, 2014.

[8] N Jindal, V Kumar," Enhanced Face Recognition Algorithm using PCA with Artificial Neural Networks", International Journal of Advanced Research in Computer Science and Software Engineering, vol 3, no. 6, p.p. 864-872, 2013.

[9] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In CVPR, 2014

[10] L. Breiman. Random forests. Machine learning, 45(1): p.p. 5–32, 2001.

[11] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. International Journal of Computer Vision, 107(2): p.p. 177–190, 2014.

[12] Caruana, R.: Multitask learning. Machine learning 28(1), p.p. 41–75, 1997.

[13] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In ECCV, p.p. 94– 108, 2014.

[14] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine autoencoder networks (CFAN) for realtime face alignment. In ECCV, p.p. 1–16, 2014.

[15] Sun, Y., Wang, X., Tang, X.: Deep convolutional network cascade for facial point detection. In: CVPR: p.p. 3476–3483, 2013.

[16] X. Zhu, D. Ramanan. "Face detection, pose estimation and landmark localization in the wild" Computer Vision and Pattern Recognition (CVPR) Providence, Rhode Island, June 2012.

[17] Z. Zhang. "Facial Landmark Detection by Deep Multi-task Learning". In: Computer Vision – ECCV 2014 Lecture Notes in Computer Science, p.p. 94– 108, 2014.

[18] J. Halberstadt, G. Rhodes. "The Attractiveness of Non-face Averages: Implications for an Evolutionary Explanation of the Attractiveness of Average Faces", July 2000.

[19] J. Halberstadt, G. Rhodes. "It's not just average faces that are attractive: Computer-manipulated averageness makes birds, fish, and automobiles attractive", 2003

[20] J.H. Koeslag. "Koinophilia: Groups sexual creatures into species, promotes stasis, and stabilizes social behaviour, Journal of Theoretical Biology", Vol.144, No.1, p.p.15–35, 1990.

[21] P. Benson, D. Perrett. 'Computer averaging and manipulations of faces'. In: Photovideo: Photography in the age of the computer, p.p. 32–38, 1991.

[22] F. Galton. 'Composite portraits'. Journal of the Anthropological Institute of Great Britain and Ireland 8, p.p. 132–142, 1878.

[23] P. Perez, M. Gangnet, A. Blake. "Poisson Image Editing", Journal of ACM Transactions on Graphics (TOG), Vol. 22 - 3, p.p. 313-318, 2003.

[24] S. Mallick. Average Face: OpenCV (C++ / Python)Tutorial.url:http://www.learnopencv.com/average-face-opencv-c-python-tutorial, 2016.

[25] M. Turk. “A random walk through Eigen space. IEICE Transactions on Information and Systems”, E84-D(12): p.p. 1586–1595, December 2001.

[26] Average Face: OpenCv (Python, C++) Tutorial. url:https://www.learnopencv.com/average-face-opencv-c-python-tutorial

[27] Recognition of individual faces and average. URL: psych.mcmaster.ca/maurerlab/Publications/deHaanPrototype.pdf.

[28] The ‘Averageness Hypothesis’ of attractiveness. URL: http://faceresearch.org/students/averageness.

[29] Average Face – Creative AI. URL: www.creativeai.net/posts/Fods4e9m8nGiy3F5a.

[30] Face Alignment with OpenCV and Python. URL: www.creativeai.net/posts/Fods4e9m8nGiy3F5a.

[31] Face Alignment with OpenCV and Python. URL: www.creativeai.net/posts/Fods4e9m8nGiy3F5a.

[32] Harry Wechsler, Andeep S. Toor, Modern art challenges face detection, Pattern Recognition Letters (2018), doi: 10.1016/j.patrec.2018.02.014

[33] Hong Ai, Xuebin Cheng, Research on embedded access control security system and face recognition system, Measurement (2018),doi: 10.1016/j.measurement.2018.04.005.

[34] Hazar Mliki, Mohamed Hammami, Hanêne Ben-Abdallah, Real time face detection based on motion and skin color information, Parallel and Distributed Processing with Applications, (2012), 10th IEEE International Symposium on Parallel and Distributed Processing with Applications, doi: 10.1109/ISPA.2012.117.

[35] Fehmi Jaafar, Angela Lozano, Yann-Ga¬ el Gu«eh«eneuc, Kim Mens, Analyzing Software Evolution and Quality by Extracting Asynchrony Change patterns, The Journal of Systems & Software (2017), doi: 10.1016/j.jss.2017.05.047.

[36] Fotini Patrona, Anargyros Chatzitofis, Dimitrios Zarpalas, Petros Daras, motion Analysis: Action Detection, Recognition and Evaluation based on motion capture data, Pattern Recognition (2017), doi: 10.1016/j.patcog.2017.12.007.

[37] Li Cuimei, Qi Zhiliang, Jia Nan, Wu Jianhua, Human face detection algorithm via Haar cascade classifier combined with three additional classifiers, Electronic Measurement & Instruments (ICEMI), 2017 13th IEEE International Conference, doi: 10.1109/ICEMI.2017.8265863.

[38] Dhananjay Rathod, Vinay A, Shylaja SS and SNatarajan, Facial Landmark Localization -A Literature Survey, International Journal of Current Engineering and Technology(2014), inpressco.com/category/ijcet.

[39] Margaret Collins, The Attractiveness of the Average Face, Seminars in Orthodontics, Volume 18, Issue3, September 2012, Pages 217-228, doi: 10.1053/j.sodo.2012.04.007.

[40] He, Xiaofe; Yan, Shuicheng; Niyogi, Partha and Zhang, HongJiang (2005). Face Recognition Using Laplacianfaces. IEEE Trans. Pattern Anal. Mach. Intell. 27(3): 328-340.

[41] D.R.Kasat, Sanjeev Jain, V. M. Thakare, A Survey of Face Morphing Techniques, nternational Journal of Computer Applications, Second National Conference on Recent Trends in Information Security, GHRCE, Nagpur, India, Jan-2014.