68 Landmarks are Efficient for 3D Face Alignment: What about More? 3D Face Alignment Method Applied to Face Recognition

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

05-10-2021 / 20-10-2021

CITATION

Jabberi, Marwa; Wali, Ali; Chaudhuri, Bidyut Baran; Alimi, Adel (2021): 68 Landmarks are Efficient for 3D Face Alignment: What about More? 3D Face Alignment Method Applied to Face Recognition. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.16742077.v2>

DOI

[10.36227/techrxiv.16742077.v2](https://doi.org/10.36227/techrxiv.16742077.v2)

68 Landmarks are Efficient for 3D Face Alignment: What about More?

3D Face Alignment Method Applied to Face Recognition

Marwa Jabberi^{1,2} · Ali Wali² · Bidyut
Baran Chaudhuri³ · Adel M.Alimi^{2,4}

the date of receipt and acceptance should be inserted later

Abstract This paper proposes a 3D face alignment of 2D face images in the wild with noisy landmarks where the objective is to recognize individuals from their single profile image. We first proceed by extracting more than 68 landmarks using a bag of features. This allows us to obtain a bag of visible and invisible facial keypoints. Then, we reconstruct a 3D face model and get a triangular mesh by meshing the obtained keypoints. For each face, the number of keypoints is not the same which makes this step very challenging. Later, we process the 3D face using butterfly and BPA algorithms to make correlation and regularity between 3D face regions. Indeed, 2D-to-3D annotations give much higher quality to the 3D reconstructed face model without the need for any additional 3D Morphable models. Finally, alignment and pose correction steps are carried out to get frontal pose by fitting the rendered 3D reconstructed face to 2D face and performing pose normalization to achieve good rates in face recognition. The recognition step is based on deep learning and it is performed using DCNNs, which are very powerful and modern, for feature learning and face identification. To verify the proposed method, two popular benchmarks, YTF and LFW databases, are tested. Compared to the best recognition results reported on these two benchmarks, our proposed method achieves comparable or even better recognition performances.

Keywords 3D face recognition · 3D face alignment · Deep learning · DCNNs · Feature extraction · 3D mesh reconstruction · 3D mesh preprocessing

Marwa Jabberi

E-mail: marwa.jabberi@regim.usf.tn

¹ University of Sousse, ISITCom, 4011, Sousse, Tunisia;

² REsearch Groups in Intelligent Machines (REGIM Lab), University of Sfax, National Engineering School of Sfax (ENIS), BP 1173, Sfax, 3038, Tunisia;

³ Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata-700108, India;

⁴ Department of Electrical and Electronic Engineering Science, Faculty of Engineering and the Built Environment, University of Johannesburg, South Africa;

1 Introduction

Face Recognition (FR) is a common authentication tool in many applications. The face has a physical biometric characteristic that is non-invasive and it is accepted by users. There is no direct contact with the acquisition device as required when using the iris and the finger print. Nowadays, FR is the most widespread technique used in authentication [1]. The main architecture of FR is shown in the figure below:

Fig. 1 Classical face recognition pipeline

The classical FR pipeline consists of two phases: Online and Offline phases. In the Offline phase, the user is not logged in. Part of the dataset follows the preprocessing steps, such as face cropping, denoising, smoothing, alignment. The feature extraction step is carried out to compute the biometric signature of each face, then to classify these signatures to categorize the features. However, the Online phase is carried out at each interrogation of the dataset by the user, where a query face goes through the same steps; i.e., preprocessing and feature extraction. Then, a check is established to know the belonging rate for each class and to establish a related decision. The decision step is a 1:N problem that compares a query face image using its biometric computed signature against all the stored signatures to determine the identity of the query face. The run time of this phase must be reduced.

The challenges facing FR are the lighting conditions, pose variation, occlusions, facial expressions, and low resolution [2]. All these problems decrease the recognition rate. To solve them, the preprocessing step must be well-developed to strengthen the recognition system and to achieve good results in recognizing

the face as taken in the wild. Numerous approaches related to this field have been advanced; however, several challenges still persist.

In this paper, we present a method in the context of FR. The aim is to tackle the pose variation problem because recognizing and identifying a person from a single 2D image under pose variation remains a great challenge. To get the highest recognition rate, an alignment step has to be well-developed. It is an essential preprocessing step in face recognition. Thus, our work includes:

1. Feature extraction to add keypoints to the 68 traditional fiducial landmarks since these keypoints provide rich information about facial geometry.
2. 3D face reconstruction from 2D obtained keypoints of a single image under an arbitrary view to localize the self-occluded face parts in the case of large poses.
3. 3D face alignment by fitting 3D reconstructed face to 2D face image using keypoints marching to render the frontal view by pose normalization and correction.
4. Application of face recognition using Deep Convolutional Neural Networks (DCNNs) on the aligned faces.

Indeed, facial alignment and 3D reconstruction are two different tasks. Currently, the relationship between these two tasks has become known. Indeed, 2D face alignment has shown weakness consisting in its inability to address large poses. The relationship between 3D face reconstruction and face alignment consists essentially in mapping and estimating the 3D face geometry from a single 2D image. The main objective is to compute the visibility and position of 2D landmarks.

Recent methods have used hand-crafted features to improve performance, especially for the earliest contributions. In this paper, our approach is applied directly to RGB face images using compact features with engineered descriptors to achieve good performance. The power of CNNs, which are used to learn the features on large multi-identities datasets for 3D face alignment with application to face recognition, is therefore exploited.

2 Related works

2.1 Face Recognition

FR is currently a widely used biometric technique since the face has become the most attractive biometric. Also, the current COVID-19 pandemic has changed several statistics worldwide, including the biometric modalities. In the earliest results of FindBiometrics reported in a review survey [3], FR (Fig.2) has retained the top spot as the year's most used and exciting modality.

FR methods can be classified into three categories: global methods using the entire facial surface [4], local methods based on local regions or patches and not considering the whole face [5], and hybrid methods [6] consisting in combining global and local feature descriptors.

Fig. 2 Results of the latest FindBiometrics until year 2020

EigenFace [7] is a global FR method which uses Principal Component Analysis (PCA) [8]. Eigen Vectors are measured to describe faces, which are computed by measuring the features of the nose tip, mouth, eye corners, and chin edges. Since global methods project face representation into a small subspace or a correlation plane, EigenFaces are projected onto a reduced face space by PCA. Eigenface has been used in several other works accompanied by modifications and improvements as summarized in [9].

Local FR methods focus on fiducial points and parts of the face to generate features. These techniques compute the local features through pixel parameters, face histograms, geometric shapes, and correlation planes between different regions. The most popular used techniques are based on different descriptors, such as Local Binary Pattern (LBP) and its derivatives, Histogram of Oriented Gradients (HOG), VanderLugt Correlator (VLC), Scale Invariant Feature Transform (SIFT). All these methods are summarized in [10].

Hybrid FR methods consist of a fusion of global and local methods. In fact, the global characteristics are combined with the local ones making this FR category the most efficient and robust [11].

2.2 Deep Face Recognition

With the advent of BigData and Data Mining, methods and approaches for FR have become numerous. In this work, our goal is to recognize individuals from their faces under pose variations using CNNs. This method proved to have impressive results. With the advent of CPU and GPU cores [12], CNNs and Deep CNNs have been used in a huge number of training data.

CNNs can be classified among the category of hybrid FR methods. They are adapted to feature learning and label prediction, as well as to mapping the input data to deep features, which are the output of the last hidden layer, and later to the predicted labels. Feature learning is carried out automatically and it is shared as weights between different layers. However, DCNNs achieve

superior performance since they are able to extract high level features ensured by the classification architecture [13]. Once deep features are extracted, most of the methods directly calculate the similarity between the two features using cosine, L2, or the nearest neighbor (NN) metrics, and therefore establish comparison for identification. Yet, deep networks which perform perfectly on benchmark datasets may fail in real world applications.

Most of the recent methods perform face image representation using hand-crafted local image descriptors, such as SIFT, LBP, HOG [14–16].

Contrary to the aforementioned methods, our method is applied to RGB pixels without combining other descriptors to improve performance.

Researchers have used CNNs and DCNNs in FR application, either for features learning, features extraction, or features classification. In CosFace [17], Large Margin Cosine Loss (LMCL), as a novel loss function, is performed to remove radial variations and to maximize the decision margin in the angular space. LMCL guides DCNNs to learn the highly discriminative face features. So, intra-class variance is minimized and inter-class variance is maximized. SphereFace [18] represents class centers in the angular space and penalizes the angles between deep features and their corresponding weights in a multiplicative way, since authors found out that linear transformation matrix in the last fully connected layer of the CNN is useful for this issue. Thus, an Additive Angular Margin Loss helps to obtain the highly discriminative features learned via DCNNs for FR. In the same context, RegularFace [19] uses intuitive geometric interpretation by penalizing the angle between an identity and its nearest neighbor by focusing on intra-class compactness. In [20], the authors focused on decreasing information redundancy in features learning and on maintaining the most informative components of spatial feature maps. This module, called attention, is added to the convolutional layer of a standard CNN.

FR methods based on deep CNNs are in full development. Indeed, to have a high recognition rate, it is absolutely necessary to focus on features, since CNNs perform feature learning in an automatic way. So, most methods add a module or an additional function to CNNs layers or focus on the preprocessing steps to keep only the salient features of the face.

2.3 Face Alignment

As mentioned in the previous section, the recognition rate is relative to the extracted and the learned face features. For this reason, the face must be well preprocessed before performing the recognition test.

The alignment process forms a part of the preprocessing steps and involves the placement of the face in a frontal position (pitch (ϕ)=0°, yaw (γ)=0° and roll (θ)=0°). More precisely, it is pose normalization since the frontal pose covers the canonical view of the face taken arbitrarily in the wild. Aligning poses make FR easier.

Fig. 3 Head poses close to the frontal pose

In the majority of papers, authors refer to face alignment as face detection while aligning faces consists in establishing a rotation in the plane and making the face in a frontal view. Moreover, a face image captured under pose variation presents missing data, which can degrade the recognition rate. Methods of face alignment are numerous and have shown impressive results with sophisticated techniques.

2D face alignment aims at establishing pose normalization if faces are in frontal or near-frontal poses as shown in Fig.3. However, this transformation fails due to out-of-plane rotation. So, 2D face alignment has difficulties [21] when addressing large poses (Fig.4). Yet, 3D face alignment consists in aligning faces despite the presence of out-of-plane rotations.

Fig. 4 Examples of large poses

2.3.1 3D Face Alignment methods based on fitting 3D generic models to 2D faces

The human face is characterized by 68 landmarks which can provide information about the head pose. The fitting process consists in pasting a 3D face model to the 2D face using landmarks as references. This is performed by minimizing the difference between the face image and the 3D face model appearance. The purpose of fitting lies in the possibility of rotating the face and performing the alignment to a frontal pose. Fitting is a method used for 3D face alignment, especially in medium poses. However, in large poses, it is very challenging because of the dramatic appearance variations when getting closer to the profile view.

In [22], the authors introduced a 3D Dense Face Alignment (3DDFA) which fits the 3D morphable model (3DMM)[23]. 3DDFA synthesizes face appearances by labeling invisible landmarks due to large poses. Its objective is to skip 2D landmark detection and start from 3DMM fitting. HPEN [24] aims at fitting the 3DMM to the 2D faces captured in the wild. The approximation method is also performed to avoid iterative visibility estimation of the masked landmarks in large poses. In addition, an identity-preserving normalization is carried out by correcting 3D transformation and anchoring adjustment in the meshed image. In the same context, a method proposed in [25] uses the Basel Face Model (BFM)[26] for 3D face alignment and keypoints locations. It consists of a deep evolutionary model integrating sparse 3D Diffusion Heat Maps (DHM) for pose assistance. A CNN is used for feature extraction and Recurrent Neural Network (RNN) is utilized for learning.

The methods already quoted have achieved the best results in FR framework, including face alignment. However, the big challenge is always evoked when dealing with large poses. Their main drawback is therefore related to the limited geometry of the 3D used models. On the other hand, the use of a 3D model, such as 3DMM or BFM to establish fitting always leaves a common signature in the extracted features.

2.3.2 3D Face Alignment methods based on 3D face model reconstruction

This process consists in reconstructing a 3D face model from a 2D face to have its own model for each input 2D face image without the need for a 3D model, such as 3DMM, BFM or any external data. Indeed, each 3D reconstructed model has its own characteristics and parameters. Thereafter, the 3D reconstructed model and the 2D landmarks are correlated by a specific technique.

DeepFace [27] modelizes a 3D face based on the extracted 67 fiducial points. Thus, this method consists in wrapping the detected facial crop to a 3D frontal model after mesh reconstruction by Delaunay triangulation. Also, the 67 anchor points are fitted to the obtained 3D shape to get correspondence between the 67 detected fiducial points and their 3D references. In the same context, another work used the Iterative Closest Points (ICP) [28] algorithm to perform

correspondence between each reconstructed 3D face and the ground truth point cloud. Then, normalized mean error (NME) is calculated by the face bounding box size.

Feng *et al.* [29] proposed a new approach for 3D face reconstruction using UV space as a position map [30]. The UV position map represents the full 3D plot of facial structure with alignment information. It is a 2D image recording 3D positions of all the points in UV space. So, the full facial geometry is reconstructed along with the semantic meaning and it is regressed to get aligned faces.

The previously cited works have used methods establishing alignment with 3D face reconstruction without 3D model basis, which is challenging but had good results. So, no 3D model shape or template restriction was present. In our method, reconstruction of 3D models for each 2D face is carried out as explained in the following section.

3 Proposed method

Conventional pipeline consists of face detection, face alignment to get frontal pose, face representation that has to be trained in the DCNN, and finally face classification to establish identification. Face detection and face alignment are preprocessing steps. In the figure below (Fig.5), our global pipeline is presented.

Fig. 5 Overview of the proposed method

We also present the following pseudo code to get an overview of the proposed method. The different steps are detailed in the following subsections.

Overall Pseudo Code (1):

1. Read input images from dataset
2. For # images
 - $2D_{Face} \leftarrow$ Face Detection and Cropping from input images
 - $2D_{Landmarks} \leftarrow$ Landmarks Detection and Extraction ($2D_{Face}$)
 - $3D_{Face}, 3D_{Landmarks} \leftarrow$ 3D Face Reconstruction ($2D_{Landmarks}$)
 - Aligned Face $\leftarrow$ 3D Alignment ($2D_{Face}, 3D_{Face}, 2D_{Landmarks}, 3D_{Landmarks}$)
3. Train DCNN (AlexNet)
4. Recognize faces and establish verification

3.1 Face Detection and Cropping

Before detecting faces in the images, we eliminate the duplicated images and check the labels. For face detection, Modified Viola–Jones algorithm is used. The pseudo code is summarized in [31].

Fig. 6 Face detection using Modified Viola–Jones algorithm

When it first appeared [32], this method was effective in the detection of faces in a frontal position; however, following certain modifications, it has become sophisticated in all scenarios. So, the face, which is our region of interest (ROI), can be detected under various poses, various illumination conditions, different skin colors, and complex backgrounds while maintaining considerable speedup by parallelizing the training. Once the face is detected, bounding boxes are randomly generated around the detected window (Fig.6).

When facial detection is established, all images are resized in the same scale. In case of images having multiple faces, each detected face is labeled manually and assigned to the appropriate class.

3.2 3D Face reconstruction

In this paper, we revisit the alignment step which consists in searching landmarks based on global shape or texture models to configure landmarks loca-

tions. However, under some view angles, landmarks are invisible. So, performance decreases for non-frontal faces and invisible landmarks are considered as self occlusions. It is for this reason that face reconstruction is required. The difference between using a 3D generic model and a 3D reconstructed model is that each 2D face has its own 3D model which preserves texture, shape, and other features. The use of a generic model, such as BFM or 3DMM, causes a common signature between all faces, which increases the error rate afterwards.

Fig. 7 68 traditional facial landmarks

3D reconstruction is established by keypoints detection which is added to the traditional 68 landmarks (Fig.7). Indeed, the addition of supplementary keypoints to face features is helpful in the reconstruction stage because the 68 landmarks are not enough for 3D mesh creation.

3.2.1 Facial keypoints detection and extraction

First, we start by locating the 68 fiducial points (Pseudo Code (2) Line 1) using the facial landmark detector included in the dlib library and OpenCV presented by [33]. The 68 (x, y) extracted landmarks allow to delineate the facial surface in the face image as shown in Fig.8. Thus, our new ROI is delimited by the jaws and eyebrows keypoints. This method is tested under large poses and this step is successfully performed.

Our choice of the 68 facial landmarks detector was made following a series of tests and experiments that proved robustness against large poses. They are detailed in the self-evaluation section.

According to the state-of-the-art studies, the presence of out of plane or invisible landmarks is noted in large poses. So, keypoints are added since the 68 landmarks are not enough for 3D face reconstruction. Indeed, this is our basic contribution.

Fig. 8 68 landmarks detection using dlib library: (a) input face image, (b) numbered 68 landmarks

Keypoints Detection and Extraction Pseudo Code (2):

Input: 2DFace

Output: $2D_{Landmarks}$

1. $68_{Landmarks} \leftarrow$ Estimate the location of 68 (x, y)-coordinates that map the facial structures on the face (2DFace) using dlib library
2. $Edges_{Canny} \leftarrow$ Edge Detection (2DFace, 'Canny')
3. $Edges_{Prewitt} \leftarrow$ Edge Detection (2DFace, 'Prewitt')
4. $Regions \leftarrow$ MSER Features Detection (2DFace)
5. $2D_{keypoints} \leftarrow$ Reshape ($68_{Landmarks}$, $Edges_{Canny}$, $Edges_{Prewitt}$, $Regions$)

The edges in the face image are detected using Canny and Prewitt edge detection algorithms [34] (Pseudo Code (2), Line 2 and 3). Only the features in the delimited ROI are kept.

The Canny method consists in finding edges by looking for local maxima in the image gradient. The edge function calculates the gradient using the derivative of a Gaussian filter. This method uses two thresholds to detect the strong and weak edges, including the weak edges in the output if they are connected to the strong ones. By using two thresholds, the Canny method helps to detect the true weak edges which can represent wrinkles in the face (Fig.9 (c)).

On the other hand, the Prewitt method aims at finding the edges at the points where the image gradient is maximum using the Prewitt approximation to the derivative (Fig.9 (d)).

Since the output is a binary image, pixels with 0 values are found and they are extracted to be added to the other keypoints. We notice that the number of keypoints is variable for each given face.

Fig. 9 Facial keypoints detection: (a) Input face image, (b) 68 Landmarks, (c) Detected edges using Canny, (d) Detected edges using Prewitt, (e) Detected regions using MSER, and (g) All detected keypoints

In addition to edge detection, Maximally Stable Extremal Regions (MSER) features [35] are added (Pseudo Code (2) Line 4). Indeed, using this descriptor (Fig.9 (e)) allows to obtain good identification of significant image parts, usually combined with high repeatability under typical image distortions. It also allows to get highlighting boundaries of the ROI, which are maximally stable extremal regions. Moreover, MSER helps to find correspondences between the image elements from two images with different viewpoints.

For each input 2D image, the detected keypoints number is not the same (Fig.9 (f)). Once keypoints are detected, they are extracted (Pseudo Code (2) Line 5) and saved under the same label as the image in order to be used in the 3D reconstruction process. In the following table, we present the number of extracted keypoints of two query face images.

Table 1 Examples of the number of the extracted keypoints of two query face images

Example	Fiducial landmarks	Canny	Prewitt	MSER	Total
marwa.jpg	68	641	307	301	1317
roua.jpg	68	724	389	625	1806

3.2.2 Face Meshing

Once the keypoints are extracted, we start meshing the ROI using Delaunay triangulation [36] (Pseudo Code (3), Line 1 and 2). The latter creates triangulations of a set of points and ensures that the circumcircle associated with each triangle contains no other points in its interior that depends on its neighborhood. Delaunay triangulation derived from the extracted facial keypoints is shown in Fig.10.

Fig. 10 3D face reconstruction from single 2D face image: (a) Input face image, (b) Cropped face, (c) Extracted keypoints, (d) Delaunay triangulation, (e) 3D Mesh creation

2D Face Meshing Pseudo Code (3):

Input: $2D_{keypoints}$

Output: $3DFace$, $3D_{keypoints}$

1. $3D_{keypoints} \leftarrow \text{Delaunay Triangulation } (2D_{keypoints})$
2. $3DFace \leftarrow \text{3D Mesh Generation } (3D_{keypoints})$
3. Write off ($3DFace$)

After the triangulation process, we obtain facial points in 3D domain, derived from the facial keypoints in 2D domain using n , which is the number of extracted landmarks. It is worth noting that n is not the same for each given face (P_0 : Initial Points, P_m : Meshed Points).

$$P_0 = [x_1, y_1, x_2, y_2, \dots, x_n, y_n]^T \in R^{2.n*1} \quad (1)$$

$$P_m = [x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_n, y_n, z_n]^T \in R^{3.n*1} \quad (2)$$

As previously mentioned, face cropping is performed to extract the face from the image, but we notice that a part of the background is still there. This part is useful in the alignment step; however, in the reconstruction of the 3D face, it should be ignored because we only need the salient part of the face. If the background in the 3D reconstruction is left, this will be very demanding in terms of time and complexity.

3.2.3 3D Face Preprocessing

This step is very important since the obtained mesh is not in good quality due to several factors, such as mesh regularity, holes coming from self occlusions. Vertices with no connections can also be found. In the following pseudo code, we present the steps to be followed to perform 3D face preprocessing.

3D Face Preprocessing Pseudo Code (4):

Input: 3DFace, 3D_{keypoints}, 2D_{keypoints}, 68_{landmarks}

Output: 3D_{Processed} Face, 3D_{keypoints}

1. 3D_{patch} Face, 3D_{keypoints} $\leftarrow$ Region Growing (3DFace, seed, radius, distance)
2. Diagonal_{Keypoints} $\leftarrow$ Diagonal Keypoints Extraction (3D_{patch} Face, 3D_{keypoints}, 2D_{keypoints}, 68_{landmarks})
3. 3D_{update} Face $\leftarrow$ 3D Mesh Update (3D_{keypoints}, Diagonal_{Keypoints})
4. 3D_{Processed}Face $\leftarrow$ Butterfly Mesh Subdivision(3D_{update}Face, iterations, edge threshold)
5. 3D_{Processed}Face $\leftarrow$ Ball Pivoting Algorithm (3D_{update} Face, radius, angle)

First of all, we extract the facial surface using Region Growing [37] (Pseudo Code (4), Line 1), which is a segmentation algorithm suitable for 3D mesh. The nose tip is used as a seed point and several tests are performed to determine the extraction radius suitable for any face shape ($r = \text{length of Bounding Box} * 0.6$). Then, the geodesic distance is used to obtain an oval shape as shown in Figure Fig. 11. Indeed, the keypoints residing around the jaws and their neighborhoods are taken into consideration.

Fig. 11 3D facial surface extraction: (a) localization of the nose tip, (b) Facial patch detection, (c) Extracted face

Once the suitable facial region (patch) is extracted from the initial generated mesh, we locate the diagonal of the face (Pseudo Code (4), Line 2) from the annotated landmarks (28, 29, 30, 31, 34, 52, 63, 67, 58, 9), as shown in Fig. 12 (b). We also extract other facial diagonal keypoints having the same coordinates on the y axis as the last ones. Then, we start generating symmetrical vertices to the y axis of each facial landmark while considering x and z axes (Pseudo Code (4), Line 3), as shown in Figure 12 (c). This allows to solve the

problem of missing parts or self occlusions caused by large poses and profile views (Fig. 12 (a)).

Fig. 12 3D mesh reconstruction (Update) by adding the missing parts: (a) Missing parts of the face caused by large poses, (b) Localization of facial diagonal keypoints referencing to the 68 annotated landmarks: The red points represent the number of landmarks: 28, 29, 30, 31, 34, 52, 63, 67, 58 and 9. The blue ones represent other detected keypoints having the same y coordinates, (c) Missing parts reconstruction

After adding the missing parts of the 3D face, the quality of the preprocessed mesh in the context of good reconstruction is improved for the pose normalization task. Remeshing to connect the new vertices and the facial surface subdivisions of the mesh is performed using the Butterfly subdivision algorithm [38] and the Ball Pivoting Algorithm (BPA) [39] (Pseudo Code (4), Line 4 and 5) for triangular interpolation (Fig.13) .

The Butterfly algorithm is used for mesh subdivisions and vertices connections. This process is very essential in 3D reconstruction to produce other vertices whose purpose is to achieve mesh regularity controlled by BPA to preserve the facial shape.

Using the butterfly algorithm, we normalize all the 3D reconstructed faces to a defined number of vertices and facets. Indeed, the original meshes do not have the same parameters since the number of extracted landmarks is variable from one face to another.

For each facet consisting of 3 vertices (3 coordinates in the 3D space x,y, and z), BPA pivots around an edge (which connects two vertices) until it touches another vertex, forming another triangle. So, BPA builds relationships between vertices having no connections, which improves the mesh regularity.

Fig. 13 Remeshing process: 3 mesh subdivision iterations using Butterfly algorithm. Red circles show interpolating triangulation using BPA

This process is iterated until connecting all the vertices in the mesh.

BPA is a very used and efficient technique for mesh interpolation. It exhibits linear time complexity and robustness in the given 3D meshes. Although these two techniques are old, they are very efficient. In the experimental part, we justify our choice using some discriminating values.

3.3 3D Face Alignment

3.3.1 Pose Normalization

After 3D mesh reconstruction and preprocessing, we wrap all the detected 2D facial keypoints by projecting the 3D reconstructed face onto the image plane using the Weak Perspective Projection [40], based on the 2D positions of the 3D points on the image plane (Pseudo Code (5), Line 1). The following pseudo code summarizes the different steps.

3D Alignment Pseudo code (5):

Input: 2DFace, 2D_{keypoints}, 3D_{Processed} Face, 3D_{Keypoints}

Output: Aligned Face

1. $f, R, t_{3d} \leftarrow \text{Project Vertex (2DFace, 2D}_{keypoints}, 3D_{Processed} \text{ Face, 3D}_{keypoints})$
2. $f, R, t_{3d}, 3D_{keypoints} \leftarrow \text{Wrap Reconstructed 3D Model (2DFace, 2D}_{keypoints}, 3D_{Processed} \text{ Face, 3D}_{keypoints})$
3. $\text{Aligned Face} \leftarrow \text{Pose Normalization (f, R, } t_{3d}, 2DFace, 2D_{keypoints}, 3D_{keypoints}, 3D_{Processed} \text{ Face)}$
4. $\text{Aligned Face} \leftarrow \text{Image Cleaning (Aligned Face)}$

Then, we fit the 3D obtained face by minimizing the difference between the 2D extracted landmarks and their references in our 3D reconstructed model while considering the rotation parameters (R is 3×3 matrix constructed with pitch (ϕ), yaw(γ), and roll(θ)), the translation vector t_{3d} , and the scale factor f given by the normalization process (Pseudo Code (5), Line 2).

$$argf_{f,R,t3d} = \min ||P_m - P_0|| \quad (3)$$

The rotation matrix is obtained by multiplying the following three matrices:

$$R_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{pmatrix} \quad (4)$$

$$R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad (5)$$

$$R_z(\gamma) = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6)$$

So, what happens to the vertices added during subdivision and remeshing when fitting is performed using initial 3D keypoints?

Indeed, the vertices added at the level of subdivision and remeshing are useful for different reasons. First, they serve for the termination of the missing parts of the face during large poses. Secondly, our fitting algorithm takes into consideration the referenced keypoint and its neighborhood since the facets present relationships between the 3 vertices in the space.

Fig. 14 Fitting results using our 3D reconstructed models

In the figure above (Fig.14), we present the results of the fitting process when using our 3D reconstructed models. The advantage of 3D reconstruction is that each identity has a specific 3D model which is useful for alignment. This makes it unique and original. In fact, there is no common factor between

the different identities. Indeed, this is useful for the recognition task. Later, we perform pose correction for the alignment step (Pseudo Code (5), Line 3). So, the 3D face designed by P_m in equation 7 is rotated by normalizing with R^{-1} to the frontal pose with 0° view centered by the nose tip and considering the pose map of the 2D extracted keypoints. This step is iterated until the face is aligned (P_a) to the desired view according to the pitch (θ), yaw (γ), and roll (ϕ) values of the frontal pose.

$$P_a = R^{-1}P_m \quad (7)$$

Once the 3D face is normalized to the frontal pose, correspondence between 3D and 2D keypoints is redone to refine the new 2D keypoints location.

Following a bibliographic study we performed, we notice that face alignment methods using generic 3D models have a problem of breaking correspondence, especially in cases of large poses. Indeed, the keypoints on the face contour boundary are not consistent. In addition, the shape of the 3D generic model is always existent. This implies that after the fitting process, all the faces will have a common touch despite the different identities, simply because they are fitted with the same 3D generic model. For this reason, full reconstruction of the 3D face of each given 2D face is efficient and recommended. So, each 2D face will have its own 3D modeling which makes it truly original following the fitting and the alignment steps.

3.3.2 Aligned image cleaning

After the fitting and the alignment processes, we notice that the images obtained are not in good condition and they contain holes and missing parts due to alignment.

Some preprocessing operations are performed to clean the resulting images and to increase the recognition rate. It is not possible to generate a reasoned face image just like the one taken in the frontal view. So, artifacts are treated using the mirroring method [41] whose purpose is to fill the holes and the missing parts caused by alignment.

In Figure 15, the graphical results of 3D face alignment when applying our method are presented. The blue circles show our method robustness and justify our contribution at the level of keypoints addition, which serves to detect more regions and to wrap all the visible parts of the face. In fact, more keypoints extraction involves good 3D face reconstruction, which leads to fit the whole face region to get better face alignment. The purpose of such face alignment is to increase face recognition results, no matter how challenging the conditions are.

Fig. 15 Graphical results of face cleaning and alignment: (a) Input image, (b) 3D Face alignment, (c) Results of image cleaning and alignment

3.4 Deep face recognition

After face frontalization and preprocessing, we move to face verification using DCNNs which eliminate the need for manual features extraction. Otherwise, the features are learned directly. We train our DCNN on a multi-class face dataset. To establish this operation, our main objectives are fast GPU-implementation of a DCNN to win a face image recognition contest and to search for successful DCNN applications for such big datasets. Applying DCNN to aligned facial images makes the network more robust to small registration errors.

In our work, we tried several DCNNs and for each dataset we kept the best recognition rates obtained. Our DCNN is therefore trained on an aligned RGB face image. The image size is adapted to the input layer of each tested DCNN. Our input consists of an RGB image of the aligned face which is given to a convolutional layer (CL) and resized according to the CL characteristic of each tested DCNN. Indeed, AlexNet [42] has given the best recognition rates that will be detailed in the experimental part of this paper.

4 Experimental results

Using our method, we present the experiments conducted on YTF and LFW datasets, which are well-known benchmarks for face recognition. Our implementation is based on the dlib library, using Python 3, MatConvNet, Image Processing, and Graph MATLAB toolbox for 3D mesh processing. Indeed, MeshLab linked to the NVIDIA packages is used to accelerate training. All

our experiments were carried out using NVIDIA CUDA development 9.2 and were run on intel (R) Core (TM) i7-7500U, 2.70 GHz and 2.90 GHz with 8GO RAM.

4.1 Experimentation and results on YTF dataset

YTF dataset [43] includes 3,425 YouTube videos of 1,595 different subjects. The used classes are the same as in LFW (a subset of the celebrities presented in the LFW dataset [44]). The videos were taken by professional photographers and were divided into 5,000 video pairs and 10 splits. They were used to evaluate the video-level face verification. The images of this dataset are not in good quality due to acquisition problems. So, a preprocessing step, including smoothing and other filters was primordial.

In this paper, we performed our experiments employing the restricted protocol, which limits the information available for training to the same/not-same labels in the training splits. Before performing 3D alignment, FR was tested via different DCNNs to check if alignment increases the recognition rate. Using AlexNet, the recognition rate was **99.14%**. In the following table (Tab.2), a comparison with some related works is presented.

Table 2 Comparison with the state-of-the-art on YTF dataset

Method	Accuracy
Aligned faces provided by YTF	78.19%
Deepface [27]	91.4%
CosFace [17]	97.6%
SphereFace [18]	95.0%
RegularFace [19]	96.7%
Our method	99.14%

4.2 Experimentation and results on LFW dataset

LFW is a big dataset for face verification testing in unconstrained domains (lighting, poses, facial expressions). It contains 13,233 face images of 5,749 different identities collected from the web. It includes 1,680 people having two or more images against 4,069 people having only single image in the dataset. In our experiments, we used the configuration described in paper [44] related to the dataset, and we only used the LFW samples. No outside data were used. After testing several DCNNs, AlexNet showed the best recognition rate. Following this experimentation, the best recognition results are once again obtained using AlexNet. They were **98.37%** with the restricted protocol and **97.28%** when using the unrestricted protocol.

The following table (Tab.3) presents a comparison between our results and

those of the existing methods using different alignment methods as described in the previous sections.

Table 3 Face recognition rates on LFW dataset, testing restricted and unrestricted protocols

Method	Restricted	Unrestricted
HPEN [24]	92.80 %	95.25%
DeepFace [27]	97.15 %	97.35%
Our method	98.37%	97.28%

4.3 Self evaluation

We carried out a series of tests to justify our qualitative and quantitative choices the different parameter and techniques. Apart from highlighting the robustness of our contribution through the rates obtained, we would also like to emphasize the quality of our work.

Fig. 16 68 facial landmarks detection: (a) Input face image, (b) 68 landmarks detection using Chow-Liu algorithm, (c) 68 landmarks detection using the Gaussian-Newton method, (d) 68 landmarks detection using dlib and OpenCV

First of all, we start by justifying the use of the 68 landmark detector. As it was mentioned in the proposed method, we used dlib and OpenCV

through Python3. This technique gave the best results for face annotation compared to Chow-Liu algorithm [46], which is widely used in recent face landmarks detection methods although it is an old technique (Fig.16 (b)), and compared to the Gaussian-Newton method [47], which is also widely used in face alignment (Fig.16 (c)). Comparison can also be made through the graphic results in figure Fig.16.

The used technique established landmarks detection in almost all pose variations. Contrarily to some other techniques, we obtained errors of landmarks detection in critical scenarios or bad locations that would be disturbing during mesh reconstruction.

Our first contribution consists in adding more keypoints to the traditional 68 facial landmarks. This is useful for 3D model reconstruction which is used in the alignment process. **So, is 3D reconstruction perfect?**

To answer this question, an experiment was carried out. An evaluation of 3D reconstruction was made based on the BU3DFE dataset [48], which contains 3D meshes accompanied by 2D images just to make sure that our reconstruction is perfect and close to the 3D faces as taken by 3D acquisition devices. We used Mean Absolute Error (MAE) evaluation metric, which measures the average magnitude of errors between prediction (3D reconstructed faces) and real 3D faces.

Fig. 17 MAE of the proposed 3D reconstruction method on BU3DF

The average MAE of 3D reconstructed faces decreases with each addition of other keypoints (Fig.17). This justifies the addition of other points to accomplish this task of reconstruction. However, the rates obtained are not within the standards. For this reason, 3D mesh preprocessing was performed to perform mesh regularization and to further decrease MAE, which would guarantee the alignment phase, as shown in Fig. 18.

Fig. 18 MAE of the proposed processed 3D reconstruction method on BU3DF using butterfly and BPA algorithms

The following histogram presents a quantitative study of the number of vertices and facets during the 3D reconstruction phase. We perform 3 iterations of mesh subdivision using Butterfly algorithm in the remeshing step. This choice is established after a series of tests. For interpolating triangulation using BPA, pivoting ball radius=3.3231 and the angle threshold= 90° .

Fig. 19 Number of vertices and facets during the 3D reconstruction process of query face image

Once 3D model reconstruction is performed for each given 2D face, fitting to wrap all the detected 2D facial landmarks is conducted by projecting the 3D reconstructed faces onto the 2D ones.

As a self-evaluation, the fitting process was tested using two widely-used existing models in face alignment, in addition to the model we generated. So, we noticed that the alignment process with fitting BFM (Fig.20 (b)) is not well-adapted to the 2D face due to projection errors. The shift is very remarkable. Apart from the cases of large poses, many images are missed because projection is unreachable.

Fig. 20 Fitting results: (a) Input images, (b) Fitting process using BFM, (c) Fitting process using 3DMM, (d) Fitting process using our reconstructed model

When using 3DMM (Fig.20 (c)), the fitting process was successful under wide poses. We also noticed that facial expressions are well-illustrated on the obtained model. This is due to the reconstruction of this generic model learned from 10000 faces in the wild. However, using this model has one drawback, consisting in image meshing each time the shape of the 3DMM is present in all faces. This implies that all the identities have the same signature, which degrades face frontalization.

Performing fitting with an appropriate 3D face model as shown in Fig. 20 (d) aids in preserving identity at the level of pose correction. All the 2D keypoints undergo this change of plane while referencing to the 3D ones.

Moreover, quantitative tests were carried out to justify and highlight our contribution. A recognition test was therefore established after having carried

out the alignment process using the previously mentioned fitting methods. We used the same technique of keypoints projection and keypoints marching. The recognition rates are summarized in the following tables (Table 4 and 5).

Table 4 Face recognition rates of aligned faces on YTF dataset

Method	Accuracy
Alignment using BFM	73.68%
Alignment using 3DMM	92.03%
Our method	99.14%

Table 5 Face recognition rates of aligned faces on LFW dataset testing restricted and unrestricted protocols

Method	Restricted	Unrestricted
Alignment using BFM	81.97%	80.6%
Alignment using 3DMM	74.64%	79.15%
Our method	98.37%	97.28%

Indeed, BFM and 3DMM are two different generic models used in the fitting process. Yet, for pose normalization, both image cleaning and image classification were carried out in the same way to be able to establish comparisons between results.

To ensure that our approach is efficient and effective, the time factor is considered. The following curve shows the consumed time in each step.

Fig. 21 Consumed time computing during the preprocessing step of the FR pipeline per a query image

4.4 Discussion

Our contribution consists in applying 3D face alignment to FR. The results obtained are among the best ones thanks essentially to the efficiency of our 3D face alignment method.

Adding keypoints consists in covering the cropped facial surface, which reduces the number and the size of regions hidden by poses. This guarantees a sophisticated 3D mesh reconstruction from a single input face image. The aim of 3D reconstruction lies in the possibility of wrapping maximum keypoints when the fitting process is established. This process facilitates face rotation with a slight damage to the 2D face image.

5 Conclusion

The paper presents a work on face recognition using DCNNs with appropriate training. We added keypoints to the 68 traditional fiducial landmarks using MSER, Canny, and Prewitt techniques.

We reconstructed 3D meshes based on Delaunay triangulation followed by facial surface extraction using Region Growing algorithm, mesh subdivision, and remeshing using Butterfly and BPA algorithms.

Then, we projected the obtained 3D mesh onto the 2D image plane and wrapped it. This step was followed by pose correction whose purpose was face alignment.

The recognition rate we found is justified by several factors. They include the well-developed preprocessing steps and efficient addition of more keypoints. This proves that 3D mesh reconstruction was very carefully conducted. So, the resulting images of faces were directly given to AlexNet without any intervention. The results obtained are comparable to those in the state-of-the-art. In the near future, we are preparing other experiments on other existing benchmarks, such as LFPW and WLFW, using our proposed method.

6 Acknowledgments

The research leading to these results has received funding from the Ministry of Higher Education and Scientific Research of Tunisia under the grant agreement number LR11ES48.

References

1. Ali, W., Tian, W., Din, S. U., Iradukunda, D., Khan, A. A. (2020) Classical and modern face recognition approaches: a complete review. *Multimedia Tools and Applications*, 1-56.
2. Oloyede, M. O., Hancke, G. P., Myburgh, H. C. (2020) A review on face recognition systems: recent approaches and challenges. *Multimedia Tools and Applications*, 79(37), 27891-27922.
3. Guides, F. R. Articles, FindBiometrics. com.

4. Anwarul, S., Dahiya, S. (2020). A comprehensive review on face recognition methods and factors affecting facial recognition accuracy. *Proceedings of ICRIC 2019*, 495-514.
5. Kortli, Y., Jridi, M., Al Falou, A., Atri, M. (2020). Face recognition systems: A survey. *Sensors*, 20(2), 342.
6. Sharif, M., Naz, F., Yasmin, M., Shahid, M. A., Rehman, A. (2017). Face Recognition: A Survey. *Journal of Engineering Science Technology Review*, 10(2).
7. Turk, M. (2005). Eigenfaces and beyond. *Face Processing: Advanced Modeling and Methods*, 55-86.
8. Shah, J. H., Sharif, M., Raza, M., Azeem, A. (2013). A Survey: Linear and Nonlinear PCA Based Face Recognition Techniques. *Int. Arab J. Inf. Technol.*, 10(6), 536-545.
9. Pandya, J. M., Rathod, D., Jadav, J. J. (2013). A survey of face recognition approach. *International Journal of Engineering Research and Applications (IJERA)*, 3(1), 632-635.
10. Soltanpour, S., Boufama, B., Wu, Q. J. (2017). A survey of local feature methods for 3D face recognition. *Pattern Recognition*, 72, 391-406.
11. Ning, X., Nan, F., Xu, S., Yu, L., Zhang, L. (2020). Multi-view frontal face image generation: a survey. *Concurrency and Computation: Practice and Experience*, e6147.
12. Li, D., Chen, X., Becchi, M., Zong, Z. (2016, October) Evaluating the energy efficiency of deep convolutional neural networks on CPUs and GPUs. In 2016 IEEE international conferences on big data and cloud computing (BDCloud), social computing and networking (SocialCom), sustainable computing and communications (SustainCom)(BDCloud-SocialCom-SustainCom) (pp. 477-484). IEEE.
13. Wang, YongXiong, Guangjun Li, and Li Ma. (2021) A sparse focus framework for visual fine-grained classification. *Multimedia Tools and Applications*, 1-19.
14. Moujahid, A., Dornaika, F., Arganda-Carreras, I., Reta, J. (2021) Efficient and compact face descriptor for driver drowsiness detection. *Expert Systems with Applications*, 168, 114334.
15. Bhople, A. R., Prakash, S. (2021) Learning similarity and dissimilarity in 3D faces with triplet network. *Multimedia Tools and Applications*, 1-19.
16. Kas, M., El-merabet, Y., Ruichek, Y., Messoussi, R. (2020) A comprehensive comparative study of handcrafted methods for face recognition LBP-like and non LBP operators. *Multimedia Tools and Applications*, 79(1), 375-413.
17. WANG, Hao, WANG, Yitong, ZHOU, Zheng, et al. (2018) Cosface: Large margin cosine loss for deep face recognition. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. p. 5265-5274.
18. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L. (2017) Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* pp. 212-220.
19. ZHAO, Kai, XU, Jingyi, et CHENG, Ming-Ming. (2019) Regularface: Deep face recognition via exclusive regularization. In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. p. 1136-1144.
20. Ling, H., Wu, J., Huang, J., Chen, J., Li, P. (2020) Attention-based convolutional neural network for deep face recognition. *Multimedia Tools and Applications*, 79(9), 5595-5616.
21. Jeni, L. A., Tulyakov, S., Yin, L., Sebe, N., Cohn, J. F. (2016, October) The first 3d face alignment in the wild (3dfaw) challenge. In *European Conference on Computer Vision* (pp. 511-520). Springer, Cham.
22. Zhu, X., Lei, Z., Liu, X., Shi, H., Li, S. Z. (2016) Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 146-155).
23. Booth, J., Roussos, A., Zafeiriou, S., Ponniah, A., Dunaway, D. (2016) A 3d morphable model learnt from 10,000 faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5543-5552).
24. Zhu, X., Lei, Z., Yan, J., Yi, D., Li, S. Z. (2015) High-fidelity pose and expression normalization for face recognition in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 787-796).
25. Sun, B., Shao, M., Xia, S., Fu, Y. (2019) Real-time Memory Efficient Large-pose Face Alignment via Deep Evolutionary Network. *arXiv preprint arXiv:1910.11818*.
26. Huber, P., Hu, G., Tena, R., Mortazavian, P., Koppen, P., Christmas, W. J., ... Kittler, J. (2016, February) A multiresolution 3d morphable face model and fitting framework. In

- Proceedings of the 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications.
27. Taigman, Y., Yang, M., Ranzato, M. A., Wolf, L. (2014) Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1701-1708).
 28. Kapoutsis, C. A., Vavoulidis, C. P., Pitas, I. (1999) Morphological iterative closest point algorithm. *IEEE Transactions on Image Processing*, 8(11), 1644-1646.
 29. Feng, Y., Wu, F., Shao, X., Wang, Y., Zhou, X. (2018) Joint 3d face reconstruction and dense alignment with position map regression network. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 534-551).
 30. Tarini, M. (2016) Volume-encoded UV-maps. *ACM Transactions on Graphics (TOG)*, 35(4), 1-13.
 31. Mutneja, V., Singh, S. (2019) Modified Viola-Jones algorithm with GPU accelerated training and parallelized skin color filtering-based face detection. *Journal of Real-Time Image Processing*, 16(5), 1573-1593.
 32. Wang, Y. Q. (2014). An analysis of the Viola-Jones face detection algorithm. *Image Processing On Line*, 4, 128-148.
 33. Kazemi, V., Sullivan, J. (2014) One millisecond face alignment with an ensemble of regression trees. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1867-1874).
 - Shoba VBT, Sam IS (2020) A hybrid features extraction on face for efficient face recognition. *Multimed Tools Appl* 79:22595–22616.
 34. Vit, P. (2016) Comparison of various edge detection technique. *Int. J. Signal Process. Image Process. Pattern Recognit*, 9, 143-158.
 35. Śluzek, A. (2016, October) Improving performances of MSER features in matching and retrieval tasks. In *European Conference on Computer Vision* (pp. 759-770). Springer, Cham.
 36. Boissonnat, J. D., Dyer, R., Ghosh, A. (2018) Delaunay triangulation of manifolds. *Foundations of Computational Mathematics*, 18(2), 399-431.
 37. Beksi, W. J., Papanikolopoulos, N. (2016, October) 3D region segmentation using topological persistence. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1079-1084). IEEE.
 38. Mandal, C., Qin, H., Vemuri, B. C. (2000) Dynamic modeling of butterfly subdivision surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 6(3), 265-287.
 39. Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G. (1999) The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4), 349-359.
 40. Bruckstein, A. M., Holt, R. J., Huang, T. S., Netravali, A. N. (1999) Optimum fiducials under weak perspective projection. *International Journal of Computer Vision*, 35(3), 223-244.
 41. Ding, L., Ding, X., Fang, C. (2012) Continuous pose normalization for pose-robust face recognition. *IEEE Signal Processing Letters*, 19(11), 721-724.
 42. Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012) Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.
 43. Wolf, L., Hassner, T., Maoz, I. (2011, June) Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011* (pp. 529-534). IEEE.
 44. Huang, G. B., Jain, V., Learned-Miller, E. (2007, October) Unsupervised joint alignment of complex images. In *2007 IEEE 11th international conference on computer vision* (pp. 1-8). IEEE.
 45. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015) Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).
 46. Chow, C. K. C. N., Liu, C. (1968) Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14(3), 462-467.
 47. Tzimiropoulos, G., Pantic, M. (2014) Gauss-newton deformable part models for face alignment in-the-wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1851-1858).
 48. Tang, H., Huang, T. S. (2008, June) 3D facial expression recognition based on automatically selected features. In *2008 IEEE computer society conference on computer vision and pattern recognition workshops* (pp. 1-8). IEEE.