Age Estimation Detection System for Facial Images

Jeff David [1]

[1] Department of Electronic, Electrical and Systems Engineering, School of Engineering, University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom
[*]talktojydii@gmail.com

Abstract: This paper presents the age classification of facial images using an anthropometric-based model. The proposed method consists of four main stages: image processing (image resizing and grayscale image), face detection, feature extraction, and age group classification. The feature extraction was performed with an understanding of face variations by age in humans. The age group classification was evaluated using the K-Nearest Neighbor (which will be referred to as "KNN" in this paper) classifier. The facial images are classified into two groups (Adult and Child), and the performance of the system is measured based on accuracy and mean absolute error (which will be referred to as "MAE" in this paper) of experiments performed on a subset of the all-aged-faces (which will be referred to as "AAF" in this paper) dataset and the face and gesture recognition research network (which will be referred to as "FG-NET" in this paper) dataset. The result shows that the test, train accuracy, and MAE are 91.43%, 92.77%, and 0.086, respectively, after using the proposed ninety-three (93) biometric ratios and applying principal component analysis (which will be referred to as "PCA" in this paper) transformation to reduce dimensionality.

Keywords – Age Estimation, Geometric Feature Extraction, Facial Anthropometry, K-Nearest Neighbor, Principal Component Analysis.

1. INTRODUCTION

Facial age estimation has been an area of interest to many researchers since the early 1900s because of its importance in application areas like in security control, law enforcement, and human-computer interaction. Facial images are used for age estimation, making its main objective to predict ages close to their appearance age [13,15].

Generally, every age estimation system needs to be able to extract useful facial features that will help predict the age of a person in an image. Age estimation can either be a classification or regression problem, both have their pros and cons, but in this study, age estimation will be considered as a two-class problem (adult or child).

An age estimation system is mainly categorized into models and algorithms. Age estimation models can be categorized into handcrafted-based models (the model used in this paper) and deep learning-based models. While the algorithms are either the classification or regression algorithm earlier mentioned [15].

Handcrafted-based models have the advantage of requiring a small dataset over the deep learning-based models. They achieve good results even though many essential features and facial data may be lost because of a lack of knowledge in extracting the facial features affected by age progression [15].

This paper's proposed method tries to extract useful facial features to classify facial images into two classes: adult and child. The dlib's HoG-based face detector alongside the 68 face landmark shape predictor is used to extract useful facial features. Then, PCA further reduces these extracted features to the important once and improves the classifier's performance, in this case, KNN.

The rest of this paper is ordered as follows: section 2: briefly reviews related works, section 3: explains how the data used in the proposed work is collected, section 4: presents the proposed method in details, in section 5: the experimental results are shown and discussed, in section 6: the paper is concluded, and a few recommendations are given, in section 7: a group of persons that have hugely impacted in the progress of this project are acknowledged and finally, section 8: provides all the sources of information cited in this paper.

2. LITERATURE REVIEW

This section reviews a couple of related works in the field of age estimation. It will be divided into three groups: algorithm papers, application papers, and review papers.

2.1 Algorithm papers

In [6], facial images are classified into three age groups: child, adult, and old. The paper aims to detect underage people. The Viola-Jones algorithm is used for face detection while Haar, HoG, geometrical, and wrinkle features are used to extract facial features. Three classifiers are used in the proposed work, namely: KNN, Artificial Neural Network (which will be referred to as "ANN" in this paper), and Support Vector Machine (which will be referred to as "SVM" in this paper) with accuracies 71.71%, 95.6%, and 78.95% respectively. The benchmark dataset of face photos from the Open University, Israel consisting of 120 images, was used.

In [4], three different modifications to KNN facial image classification are presented. The paper aims to develop an efficient exact age estimation system. Images from age 0 to 39 years from the FG-NET dataset are used and are divided into four classes. The 68 facial landmarks attached to the database are used as local features, and Local Binary Pattern

(which will be referred to as “LBP” in this paper) features are used as global features. These two features are normalized and then combined into one vector, after which the proposed modifications are applied.

In [5], facial images are classified into three classes, namely: first class (3 years to 7 years and 26 years to 30 years), second class (8 years to 25 years), and third class (31 years to 50 years). The FG-NET dataset and J48 classifier are used. The proposed system's accuracy is 89.13%.

In [7], a new system structure is proposed. A four-stage fusion framework is designed to improve the performance of facial age estimation under a constrained condition. The framework starts with gender recognition, which is the first stage, then onto gender-specific age grouping (second stage), age estimation within age groups (third stage), and finally, the fusion stage (fourth stage). Three datasets were used in this study, namely: MORPH-II, FG-NET, and CLAP2016.

In [3], facial images are classified into three classes, namely: class 1 (1 year to 10 years), class 2 (11 years to 23 years), and class 3 (24 years and above). The paper aims to optimize facial feature points by establishing a mathematical relationship among facial features and using them for age classification. Eight facial landmarks are marked manually on the facial images, and the Euclidean distance between the selected marked points are calculated using Matlab. Four distances are selected as the final features of interest. The SVM-Sequential Minimal Optimization (SMO) algorithm is used for classification in the proposed work, with approximately 95.97%. One hundred and fifty (150) faces for 50 persons are considered with an age range from 3 years to 45 years from the FG-NET dataset.

In [2], facial images are classified into five classes, namely: AG1(0-2), AG2(3-7), AG3(8-19), AG4(20-39), and AG5(40-60). The paper aims to use geometrical ratios calculated based on the distance and the size of selected facial features to distinguish age groups. It is unclear how the facial landmarks are formulated, but seventeen facial landmarks and ten facial measurements are used to formulate six geometric ratios in the proposed work. Three classifiers were used in the proposed work, namely: Neural Network Classifier (NNC), Support Vector Classifier (SVC), and normal densities-based linear classifier (LDC) with accuracies 98.21%, 98.52%, and 97.56% respectively for separating babes (0-2 years) from adults (3-60 years). Six hundred and fifty (650) frontal images from the FG-NET dataset and 204 frontal images from the Iranian Face Database (IFDB). Only frontal images without glasses, beard, and mustache from different ages were used.

In [1], a set of ratios used to classify baby facial images from two older groups are evaluated. These ratios only require the automatic localization of primary features: the eyes, nose, mouth, chin, and virtual top of the head. The approach taken in the proposed method is by first finding the initial oval and the eyes, then finding the chin and the sides of the face before the geometric ratios are computed.

2.2 Application papers

In [10], an approach for reconstructing a 3D human face from a single 2D frontal face image and a set of input facial landmarks is presented. The proposed method focuses on its use, mainly in forensic applications, where images can contain uneven lighting or partial facial occlusions. The method used in this paper does not rely on computing the depth from the input image; instead, it utilizes a database of precomputed depth images, making it suitable for 2D images that are not captured under perfect lighting conditions to be computed. The method also permits the use of images with partially occluded faces. The algorithm searches for the most suitable depth images, using a similarity metric based on Procrustes distance, age, and gender and uses them to reconstruct the human face's 3D mesh. Fifty (50) facial images; 25 males, 25 females, aged from 18 years to 44 years with an average of 25.3 years are used. These images were generated from 3D textured facial models extracted from the Extended FIDENTIS database.

In [8], age progression and regression are differentiated, and the processes are explained. It explains how these processes are mostly used as a forensic tool by law enforcement officers to show the likely current appearance of a missing person, identify fugitives, and assist in criminal investigations.

In [9], the process of typical facial growth trajectory within the categories of different age groups is described. 2D and 3D facial rejuvenation models are considered to digitally rejuvenate an adult face appearance down to its childhood based on statistical measurements. First, the geometrical model is applied to the face. The weighted sum calculation is applied to the geometrical results and the related reconstructive face template to adjust the texture to the target age.

In [11], an automatic face tracking application with an embedded age range estimation algorithm is designed. The Viola-Jones algorithm is used for face detection. The facial features are extracted by the automatic location of the optimal facial landmark points based on ideal frontal symmetry and proportion of the face using face image as an input. The proposed technique will recognize five different classes, namely: R1(0-7 years), R2(18-30 years), R3(31-40 years), R4(41-50 years), and R5(51 years and above). The Hessian based filter wrinkle analysis is also used to extract additional facial features. The database used is the FG-NET dataset, and the classifier used is the multi-SVM classifier. Experimental results show an accuracy of 92.6%.

2.3 Review papers

The study in [17] combines different age progression techniques for juvenile subjects. It also describes various researches based on longitudinal radiographic data: physical anthropometric measurements of the head and face, and digital image measurements in pixels.

In [16], the main aspects that can increase the age estimation system's performance are analyzed. It presents the handcrafted-based models and deep learning-based models and shows how the evaluations are being conducted. It discusses the proposed algorithms and models in the age estimation and shows the main limitations and challenges facing the age estimation process. Also, different aging databases are discussed.

In [14], earlier techniques proposed by researchers for facial based age estimation are analyzed. It discusses different feature extraction and estimator learning methods. Also, different aging databases are discussed.

In [15], a review of the existing literature and approaches used for facial recognition and age estimation is discussed. It discusses research challenges and forward recommendations for future research in face detection and age estimation

techniques. Its primary focus is on face recognition, age estimation, and facial features.

In [12], a brief description of age estimation and facial changes during growth and aging is given. It focused on analyzing handcrafted-based models, namely: anthropometric model, active appearance model, aging pattern subspace, and age manifold. It also compared various algorithms against their MAE's.

From the relevant papers reviewed above, a few research gaps were noted. We find that both the handcrafted and deep learning-based methods were tested on age estimation and have shown promising results. However, the facial features extracted for some of the proposed methods have shown some inconsistencies in performance. For this concern, a more robust facial feature set is investigated in this paper.

3. DATA COLLECTION

This paper used a subset of two datasets, namely the Face and Gesture Recognition Research Network (which will be referred to as "FG-NET" in this paper) [21] and the All-Age-Faces (which will be referred to as "AAF" in this paper) [22]. The FG-NET dataset is publicly available, and it consists of 1,002 facial images of 82 individuals. The age range is from 0 – 69. The AAF dataset is also publicly available, and it consists of 13,322 facial images (mostly Asian). The age range is from 2 to 80 years (including 7,381 females and 5,941 males).

In all of these, the facial images of interest are the frontal face images, which excludes facial images with large variations of pose and expression. Also, these exclude frontal face images that suffer from false facial landmark detection.

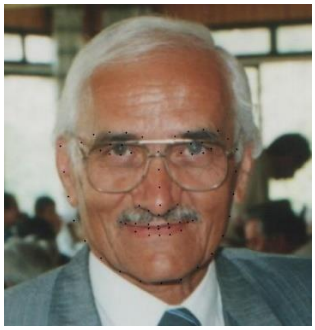
(a)  (b) 

Fig. 1: (a) example of a facial image with a considerable variation of pose and expression. (b) example of a frontal facial image with false facial landmark detection. Source: [21]

As a result, 227 facial images were passed for frontal facial images of interest from the FG-NET dataset, and 1,356 facial images were passed for frontal facial images of interest from the AAF dataset. Altogether, the dataset used in all the experiments performed in the upcoming sections contains 1,583 frontal facial images, and the age range is from 0 – 80 years. The sample sizes for each age contained in the dataset in the order, "Age (in years) – Sample size" is given as follows: 00 year – 10 images, 01 year – 6 images, 02 years – 23 images, 03 years – 17 images, 04 years – 15 images, 05 years – 15 images, 06 years – 18 images, 07 years – 21 images, 08 years – 22 images, 09 years – 14 images, 10 years – 21 images, 11 years – 26 images, 12 years – 27 images, 13 years – 24 images, 14 years – 16 images, 15 years – 14 images, 16 years – 21 images, 17 years – 17 images, 18 years – 29 images, 19 years – 26 images, 20 years – 21 images, 22 years – 11 images, 23 years – 29 images, 24 years – 23 images, 25 years – 13 images, 26 years – 15 images, 27 years – 24 images, 28 years – 32 images, 29 years – 38 images, 30 years – 22 images, 31 years – 31 images, 32 years – 35 images, 33 years – 39 images, 34 years – 35 images, 35 years – 38 images, 36 years – 38 images, 37 years – 48 images, 38 years – 32 images, 39 years – 19 images, 40 years – 15 images, 41 years – 25 images, 42 years – 30 images, 43 years – 31 images, 44 years – 24 images, 45 years – 36 images, 46 years – 12 images, 47 years – 24 images, 48 years – 17 images, 49 years – 47 images, 50 years – 9 images, 51 years – 24 images, 52 years – 20 images, 53 years – 17 images, 54 years – 20 images, 55 years – 23 images, 56 years – 37 images, 57 years – 27 images, 58 years – 10 images, 59 years – 19 images, 60 years – 10 images, 61 years – 15 images, 62 years – 9 images, 63 years – 14 images, 64 years – 4 images, 65 years – 6 images, 66 years – 5 images, 67 years – 12 images, 68 years – 8 images, 69 years – 11 images, 70 years – 4 images, 71 years – 5 images, 72 years – 5 images, 73 years – 8 images, 74 years – 6 images, 75 years – 4 images, 76 years – 9 images, 77 years – 8 images, 78 years – 8 images, 79 years – 4 images, 80 years – 9 images.

4. PROPOSED WORK & METHODOLOGY

The proposed method, as seen in figure 2, first preprocesses the face image. This helps in the face detection and location of the facial landmarks. Ninety-two different distances are calculated between thirty-one selected facial landmarks, and this information is used to calculate ninety-three biometric ratios. In the next phase, PCA is applied to create new features from the most essential extracted features to improve the classification performance by the classifier, K-Nearest Neighbor (which will be referred to as "KNN" in this paper).

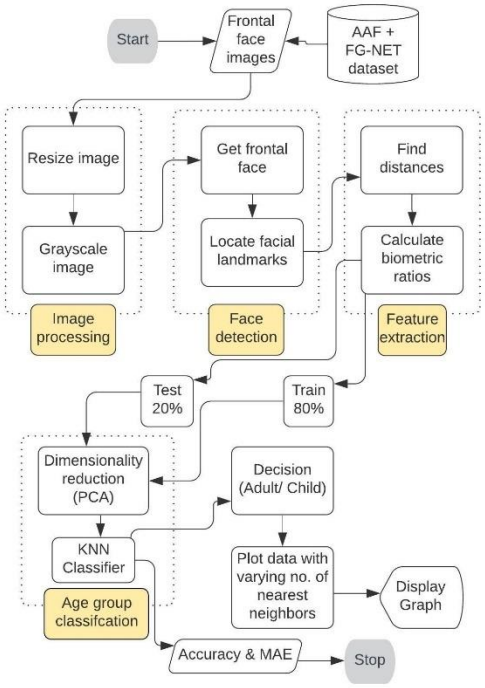


Fig. 2: The flowchart for the proposed age estimation system

4.1 Image Processing

4.1.1 Resize Image

The resize function from the imutils library was used. It provides the keyword arguments, width, and height to resize the image to the expected width/height while (1) preserving aspect ratio and (2) ensuring that the developer does not directly measure the dimensions of the image [24].

In this paper, the width is set to 500 because the Histogram of Gradients (which will be referred to as "HoG" in this paper) face detector in dlib is trained for a minimum face size of 80 x 80 and to make it large enough to view [23].

4.1.2 Grayscale Image

A pixel color in an image is a combination of three colors: Red, Green, and Blue (which will be referred to as "RGB" respectively in this paper). If RGB occupies 8 bit, then the combination of RGB occupies 24 bit and supports 16,777,216 different colors. The 24 bit represents the color of a pixel in the color image. The grayscale image (which will be referred to as "GY" in this paper) is represented by luminance using 8 bits. The luminance of a pixel value of a grayscale image ranges from 0 to 255 [18]. Transforming a color picture into a grayscale image is converting the RGB values (24 bit) into grayscale value (8 bit) as shown in the equation below: -

$$GY = ((0.3xR) + (0.59xG) + (0.11xB))$$

In GY, the contributions are as follows; the red color is decreased by 30%, the green color is increased by 59%, and the blue color is reduced by 11% [26].

4.2 Face detection

4.2.1 Get frontal face

The function used in the dlib library is the 'get_frontal_face_detector,' and this gets the frontal face of the grayscale image, as shown in figure 3. It is an HoG-based face detector that uses a Support Vector Machine (which is to be referred to as "SVM" in this paper) classifier. It is built out of five HoG filters, namely, front looking, left looking, right looking, front looking but rotated left, and a front looking but rotated right. This model comes embedded in the header file itself.

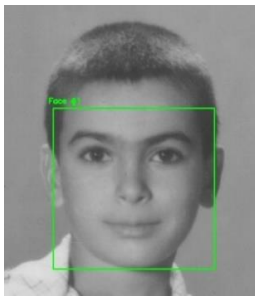


Fig. 3: Example of a face detected by the dlib's face detector. Source: [21]

The dataset used for training this function consists of 2,825 images obtained from the Labelled Faces in the Wild (which is to be referred to as "LFW" in this paper) dataset and manually annotated by Davis King, the author of dlib [23].

The face detector is an implementation of the paper by Dalal and Triggs [17]. However, the method used was slightly augmented to use the version of HoG features from the paper by Felzenswalb et al. [19].

In the paper by Dalal and Triggs [17], the process was implemented for human body detection, and the detection is as seen in figure 4. The explanation of this process is as follows:

(a) Preprocessing the data: - The data needs to be preprocessed, and the width to height ratio brought down to 1:2. The image size should preferably be 64x128. This process is done because the image will be divided into 8x8 and 16x16 patches to extract the features.

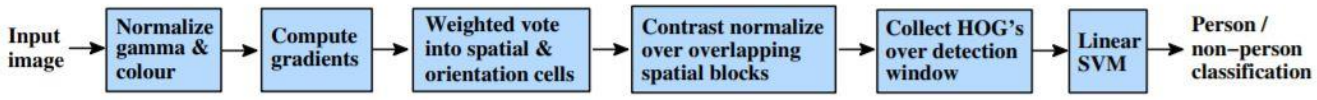


Fig. 4: An overview of the object detection chain in [17]

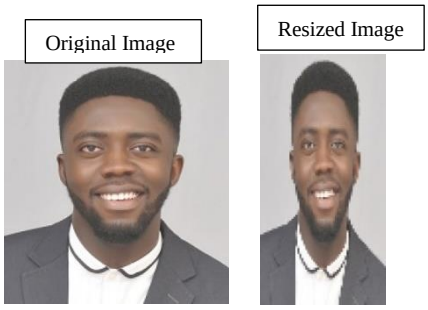


Fig. 5: Illustration of preprocessing the data in [17]

(b) Calculating gradients (direction x and y): - The gradient of every pixel in the image will be calculated next. Below, a small patch from the image is taken and used to calculate the gradients:

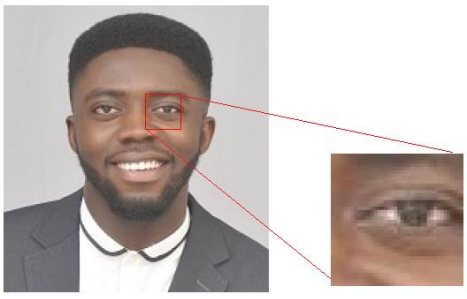


Fig. 6: Illustration of a small patch taken from the image

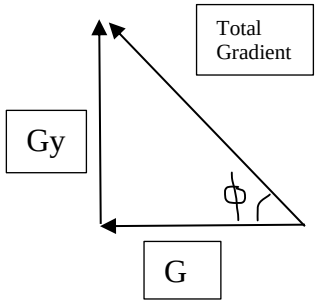
The pixel values will be gotten from this patch. For instance, if the pixel matrix is generated as seen below.

207	67	18	47	96
63	87	85	68	111
40	66	22	78	89
154	125	20	56	7
67	145	186	78	10

The pixel value, 22 has been highlighted. To determine the gradient in the x-direction, the pixel value is subtracted on the left from the one on the right. Similarly, for the y-direction, the pixel value is subtracted below from the one above.
The result of the determination is given below:

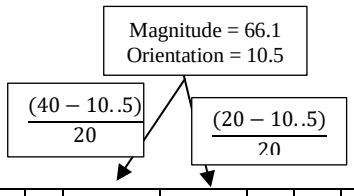
- Change in X direction (Gx) = 78 – 22 = 12
- Change in Y direction (Gy) = 85 – 20 = 65

(c) Calculate the magnitude and orientation: -
Total Gradient Magnitude =
$\sqrt{[(Gx)^2 + (Gy)^2]} = \sqrt{(12^2 + 65^2)} = 66.1$
Calculating the orientation for the same pixel,
$\tan(\phi) = {}^{Gy}/_{Gx}$,
$\phi = \tan^{-1}({}^{Gy}/_{Gx}) = \tan^{-1}({}^{12}/_{65}) = 10.5°$



So, now for every pixel value, there is a total gradient and orientation, and these will be used to generate the histogram.

(d) Create histogram using Gradients and orientation: - Histogram features are created for higher bin values to generate the histogram. A bin size of 20 is used. So, the number of buckets gotten here is 9.



Magnitude		(29.5/20) x 66.1	(9.5/20) x 66.1						
Bin	0	20	40	60	80	100	120	140	160

The HOG feature descriptor is not generated for the whole image; instead, the image is divided into 8x8 cells, and the histogram of oriented gradients is computed for each cell. If the image is divided into 8×8 cells and the histograms are generated, a 9 x 1 matrix for each cell will be gotten.

(e) Normalize gradients in 16x16 cell (36x1): - Once the HoG for 8x8 patches in the image is generated, the next step is to normalize the histogram.
This process is done by combining four 8x8 cells to create a 16x16 block. Therefore, four 9x1 matrices will be available, and each of these values is divided by each values' sum of squares' square root.
Mathematically,
Given, vector V

V = [a1, a2, a3, . . . a36]

The root of the sum of squares will be calculated as follows:

$$k = \sqrt{(a1)^2 + (a2)^2 + (a3)^2 + \cdots (a36)^2}$$

& divide all vector, V quantities with this value k:

Normalized Vector = $(\frac{a1}{k}, \frac{a2}{k}, \frac{a3}{k}, \ldots \frac{a36}{k})$

The result will be a normalized vector of size 36x1.

(f) Features for the complete image: - 105 (7x15) blocks of 16x16 for a single 64x128 image will be obtained. Each of the 105 blocks has a vector of 36x1 as features. Therefore, the total features for the image would be 105x36x1 = 3780 features.

(g) Decision (Person/ no person): - After the HOG features are generated, these features are then fed into a classifier (in this case, Support Vector Machine) that will assess whether a face is present in the image or not.

4.2.2 Locate facial landmarks

In locating facial landmarks, the dlib function: dlib shape_predictor and dlib model, dlib.shape_predictor_68_face_landmarks.dat are used.

- dlib shape_predictor [23]: This object is a tool that takes an image region containing some object and generates a collection of point locations that define the pose of the object. The classic example of this is the human face prediction. A human face's image is taken as input, and the model is expected to identify the locations of critical facial landmarks such as the corners of the mouth and eyes, and the tip of the nose. This process is shown in figure 7.
- dlib shape_predictor_68_face_landmarks.dat [25]: It was trained on the ibug 300-W dataset. It is designed for use with dlib's HOG face detector. This model is an implementation of the method used by Kazemi and Sullivan [20].

In the paper by Kazemi and Sullivan [20], they presented an algorithm to precisely estimate the facial landmarks' position in a computationally efficient way. This method utilizes a cascade of regressors.

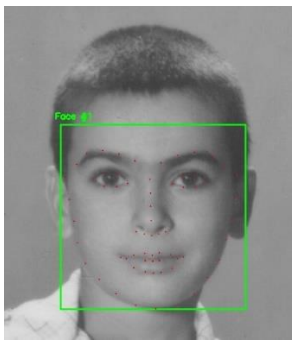


Fig. 7: Example of a face with facial landmarks detected by the dlib's 68 landmarks shape predictor model. Source: [21]

Let $\mathbf{x}_i \in R1$ be the *x, y*-coordinates of the *i*th facial landmark in an image *I*,
Then the vector $\mathbf{S} = (\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_p^T)^T \in R^{1p}$ denotes the coordinates of all the *p* facial landmarks in *I*,
Also, the vector **S** as the shape,
$\hat{\mathbf{S}}^{(t)}$ represents the current S estimate.

In the cascade, each regressor, $r_t(\cdot, \cdot)$ predicts an update vector from the image and $\hat{\mathbf{S}}^{(t)}$ that is added to the current shape estimate $\hat{\mathbf{S}}^{(t)}$ to improve the estimate:

$$\hat{\mathbf{S}}^{(t+1)} = \hat{\mathbf{S}}^{(t)} + r_t(I, \hat{\mathbf{S}}^{(t)})$$

Predictions made by the regressor, r_t are based on features, such as the values of pixel intensity, derived from I and indexed in relation to the current shape estimate $\hat{\mathbf{S}}^{(t)}$. Some form of geometric invariance is introduced into the process. One can be more confident that a precise semantic location on the face is being indexed as the cascade proceeds.
The set of outputs expanded by the ensemble is ensured to lie in a linear subspace of training data if the initial estimate $\hat{\mathbf{S}}^{(0)}$ belongs to this space. Therefore, there is no need to enforce additional constraints on the predictions, which greatly simplifies this method. The initial shape can be selected as the mean shape of the training data-centered and scaled according to a generic face detector (in my case, dlib's HOG face detector).
The gradient tree-boosting algorithm with a sum of square error loss is used
to train each regressor rt.

Furthermore, this process will be explained in detail as follows:

1. Learning each regressor in the cascade
   Assume the training data $(J_1, S_1), \ldots, (J_n, S_n)$ where each J_i is a face image and S_i, its shape vector. To learn the initial regression function r_0 in the cascade, an initial shape estimate and the target update step is created from the training data triplets of a face image, that is, $(J_{\pi i}, \hat{\mathbf{S}}_i^{(0)}, \Delta\mathbf{S}_i^{(0)})$
   Where $\pi_i \in \{1, \ldots, n\}$
   $\hat{\mathbf{S}}_i^{(0)} \in \{\mathbf{S}_1, \ldots, \mathbf{S}_n\} \setminus \mathbf{S}_{\pi i}$
   $\Delta\mathbf{S}_i^{(0)} = \mathbf{S}_{\pi i} - \hat{\mathbf{S}}_i^{(0)}$

   for $i = 1, \ldots, N$. The total number of these triplets to $N = nR$ is set, where R is the number of initializations used per image Ji. For an image, each initial shape estimate is sampled consistently from $\{\mathbf{S}_1, \ldots, \mathbf{S}_n\}$ without substitution. From this data, the regression function r_0, using gradient tree boosting with a sum of square error loss, is learned. The training triplets' set is then updated to provide the training data, $(J_{\pi i}, \hat{\mathbf{S}}_i^{(1)}, \Delta\mathbf{S}_i^{(1)})$, for the next regressor r_1 in the cascade by setting (with $t = 0$)

$$\hat{\mathbf{S}}_i^{(t+1)} = \hat{\mathbf{S}}_i^{(t)} + r_t(J_{\pi i}, \hat{\mathbf{S}}_i^{(t)})$$
$$\Delta\mathbf{S}_i^{(t+1)} = \mathbf{S}_{\pi i} - \hat{\mathbf{S}}_i^{(t+1)}$$

   This process is repeated until a cascade of T regressors $r_0, r_1, \ldots r_{T-1}$ is learned, which, when combined, give a sufficient level of accuracy.

2. Tree-based regressor
   The essence of each regression function rt is the tree-based regressors fit to the residual targets during the gradient boosting algorithm.

   (a) Shape invariant split tests
   In the regression tree at each split node, a decision based on thresholding the difference between two pixels' intensities is made. When these pixels are defined in the coordinate system of the mean shape, their positions are at, u, and v in the test. For a face image with a random shape, we would like to index the points with the same position in relation to its shape as u and v have to the mean shape. Before extracting the features to achieve this, the image can be deformed to the mean form based on the current shape estimate. Since a very sparse representation of the image is used, distorting the location of points instead of the whole image is much more effective. In addition, a crude warping approximation can be made using only a transformation of global similarity and local translations.

   (b) Choosing the node splits
   For each regression tree, where a constant vector is fit to each leaf node, the underlying function is approximated with a piecewise constant function. To train the regression tree, a set of candidate splits is randomly generated, which is θ's, at each node. Then the $\theta *$ is greedily chosen from these candidates, which minimizes the sum of square error. If Q is the set of the training examples' indices at a node, this corresponds to minimizing.

$$E(\mathcal{Q}, \boldsymbol{\theta}) = \sum_{s \in \{l,r\}} \sum_{i \in \mathcal{Q}_{\boldsymbol{\theta},s}} \|\mathbf{r}_i - \boldsymbol{\mu}_{\boldsymbol{\theta},s}\|^2$$

   where $Q_{\theta,l}$ is the examples' indices, sent to the left node as a result of the decision induced by θ, r_i - in the gradient boosting algorithm, is the vector of all the residuals computed for the image, i and

$$\mu_{\boldsymbol{\theta},s} = \frac{1}{|\mathcal{Q}_{\boldsymbol{\theta},s}|} \sum_{i \in \mathcal{Q}_{\boldsymbol{\theta},s}} \mathbf{r}_i, \quad \text{for } s \in \{l, r\}$$

(c) Feature selection
The decision is based on thresholding the difference in intensity values at a pair of pixels at each node. This process is a relatively simple test, but it is much more potent than single intensity thresholding because of its relative insensitivity to global lighting changes.

$$P(\mathbf{u}, \mathbf{v}) \propto e^{-\lambda \|\mathbf{u}-\mathbf{v}\|}$$

3. Handling missing labels
The objective of equation E(Q,θ) is easily extended to handle cases where some of the landmarks are not labeled in the training images (or we have an uncertainty measure for each landmark). Introduce variables $w_{i,j} \in [0, 1]$ for the training image, i, and each landmark j. Setting $w_{i,j}$ to 1, which indicates that the landmark j is labeled in the ith image while setting it to 0 indicates that it is.

To account for these weight factors, the gradient boosting algorithm must be modified. Simply initializing the ensemble model with the weighted average of targets and fitting regression trees into the weighted residuals can do this.

4.3 Feature extraction

4.3.1 Finding distances

The aim of the facial feature extraction in this paper is to calculate biometric ratios, and to do so, distances between important facial landmarks need to be calculated.

In this study, a mesh of facial landmarks around the eyes, eyebrows, nose, mouth, and jaw has been created, as shown in figure 8.

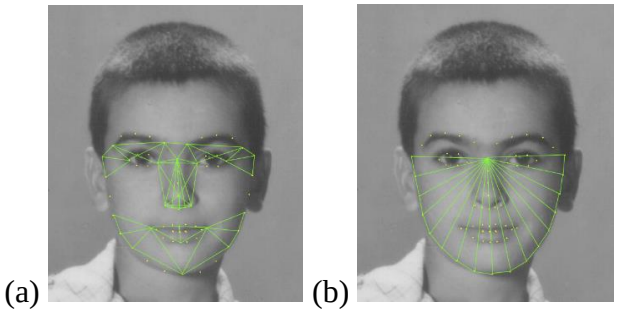


Fig. 8: (a) Illustration of the mesh created around the eyes, eyebrows, nose, mouth, and jaw (b) Illustration of the 17 radiuses and mesh defining the jaw shape and scale. Source:[21]

The mesh of facial landmarks is inspired by the initial polar mesh used in [10]. Also, the idea of facial radiuses is inspired by the twenty radiuses used in [9]. The eye grows rapidly from age zero to three years and continues to grow until almost adulthood at thirteen years. The eyebrows grow along with the eyes. The nose grows downwards and forward from childhood to early adulthood. The lip grows at a gradual rate from childhood until age 15 years. The upper lip grows away from the palate, and the lower lip grows away from the chin.

It should be noted that the upper lip grows slightly with age. The face of a child is broad and less in height. Facial growth is mostly seen more in the vertical direction and the horizontal direction, and this is because of mandibular, maxillary, and zygomatic growth. Studies have shown that facial growth occurs during these years: between three and four years, between seven and eleven years, and between sixteen and nineteen years [9].

In this paper, ninety-two different distances are calculated between thirty-one selected facial landmark points, and this information is used to calculate ninety-three (93) biometric ratios.

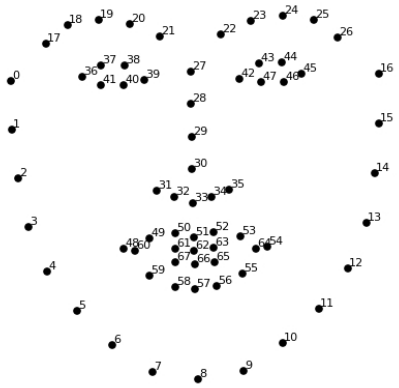


Fig. 9: Illustration of dlib's 68 facial landmarks with their landmark numbers. Source: [31]

To find the distances, the Euclidean distance (which will be referred to as "ED" in this paper) between the thirty-one selected facial landmark points (which will be referred to as "L_PT" in this paper) will be calculated and used to find the biometric ratios.

4.3.2 Calculate biometric ratios

Refer to figure 9 as a guide to understand the ninety-three biometric ratios calculated and used in the proposed system. The ratios are given as follows:

(a) Ratios associated with the right eye and right eyebrow

$$\text{Ratio } 1 = \frac{ED \text{ between } L\_PT[0] \text{ and } L\_PT[36]}{ED \text{ between } L\_PT[17] \text{ and } L\_PT[36]}$$

$$\text{Ratio } 2 = \frac{ED \text{ between } L\_PT[0] \text{ and } L\_PT[17]}{ED \text{ between } L\_PT[0] \text{ and } L\_PT[36]}$$

$$\text{Ratio } 3 = \frac{ED \text{ between } L\_PT[0] \text{ and } L\_PT[36]}{ED \text{ between } L\_PT[1] \text{ and } L\_PT[36]}$$

$$\text{Ratio } 4 = \frac{ED \text{ between } L\_PT[0] \text{ and } L\_PT[1]}{ED \text{ between } L\_PT[1] \text{ and } L\_PT[36]}$$

$$\text{Ratio } 5 = \frac{ED \text{ between } L\_PT[1] \text{ and } L\_PT[36]}{ED \text{ between } L\_PT[2] \text{ and } L\_PT[36]}$$

$$\text{Ratio } 6 = \frac{ED \text{ between } L\_PT[1] \text{ and } L\_PT[2]}{ED \text{ between } L\_PT[2] \text{ and } L\_PT[36]}$$

$$\text{Ratio } 7 = \frac{ED\ between\ L\_PT[17]\ and\ L\_PT[36]}{ED\ between\ L\_PT[17]\ and\ L\_PT[21]}$$

$$\text{Ratio } 8 = \frac{ED\ between\ L\_PT[36]\ and\ L\_PT[21]}{ED\ between\ L\_PT[17]\ and\ L\_PT[36]}$$

$$\text{Ratio } 9 = \frac{ED\ between\ L\_PT[39]\ and\ L\_PT[17]}{ED\ between\ L\_PT[39]\ and\ L\_PT[21]}$$

$$\text{Ratio } 10 = \frac{ED\ between\ L\_PT[39]\ and\ L\_PT[21]}{ED\ between\ L\_PT[17]\ and\ L\_PT[21]}$$

(b) Ratios associated with the left eye and left eyebrow

$$\text{Ratio } 11 = \frac{ED\ between\ L\_PT[45]\ and\ L\_PT[16]}{ED\ between\ L\_PT[45]\ and\ L\_PT[26]}$$

$$\text{Ratio } 12 = \frac{ED\ between\ L\_PT[16]\ and\ L\_PT[26]}{ED\ between\ L\_PT[26]\ and\ L\_PT[45]}$$

$$\text{Ratio } 13 = \frac{ED\ between\ L\_PT[45]\ and\ L\_PT[16]}{ED\ between\ L\_PT[45]\ and\ L\_PT[15]}$$

$$\text{Ratio } 14 = \frac{ED\ between\ L\_PT[45]\ and\ L\_PT[15]}{ED\ between\ L\_PT[15]\ and\ L\_PT[16]}$$

$$\text{Ratio } 15 = \frac{ED\ between\ L\_PT[45]\ and\ L\_PT[15]}{ED\ between\ L\_PT[45]\ and\ L\_PT[14]}$$

$$\text{Ratio } 16 = \frac{ED\ between\ L\_PT[45]\ and\ L\_PT[14]}{ED\ between\ L\_PT[14]\ and\ L\_PT[15]}$$

$$\text{Ratio } 17 = \frac{ED\ between\ L\_PT[45]\ and\ L\_PT[26]}{ED\ between\ L\_PT[22]\ and\ L\_PT[45]}$$

$$\text{Ratio } 18 = \frac{ED\ between\ L\_PT[22]\ and\ L\_PT[26]}{ED\ between\ L\_PT[22]\ and\ L\_PT[45]}$$

$$\text{Ratio } 19 = \frac{ED\ between\ L\_PT[42]\ and\ L\_PT[22]}{ED\ between\ L\_PT[42]\ and\ L\_PT[26]}$$

$$\text{Ratio } 20 = \frac{ED\ between\ L\_PT[42]\ and\ L\_PT[26]}{ED\ between\ L\_PT[22]\ and\ L\_PT[26]}$$

(c) Ratios associated with eyes, eyebrows, and nose

$$\text{Ratio } 21 = \frac{ED\ between\ L\_PT[39]\ and\ L\_PT[21]}{ED\ between\ L\_PT[39]\ and\ L\_PT[27]}$$

$$\text{Ratio } 22 = \frac{ED\ between\ L\_PT[39]\ and\ L\_PT[27]}{ED\ between\ L\_PT[21]\ and\ L\_PT[27]}$$

$$\text{Ratio } 23 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[22]}{ED\ between\ L\_PT[27]\ and\ L\_PT[42]}$$

$$\text{Ratio } 24 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[42]}{ED\ between\ L\_PT[42]\ and\ L\_PT[22]}$$

$$\text{Ratio } 25 = \frac{ED\ between\ L\_PT[39]\ and\ L\_PT[42]}{ED\ between\ L\_PT[27]\ and\ L\_PT[39]}$$

$$\text{Ratio } 26 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[42]}{ED\ between\ L\_PT[39]\ and\ L\_PT[42]}$$

$$\text{Ratio } 27 = \frac{ED\ between\ L\_PT[39]\ and\ L\_PT[42]}{ED\ between\ L\_PT[39]\ and\ L\_PT[30]}$$

$$\text{Ratio } 28 = \frac{ED\ between\ L\_PT[39]\ and\ L\_PT[42]}{ED\ between\ L\_PT[42]\ and\ L\_PT[30]}$$

$$\text{Ratio } 29 = \frac{ED\ between\ L\_PT[21]\ and\ L\_PT[22]}{ED\ between\ L\_PT[21]\ and\ L\_PT[27]}$$

$$\text{Ratio } 30 = \frac{ED\ between\ L\_PT[21]\ and\ L\_PT[22]}{ED\ between\ L\_PT[22]\ and\ L\_PT[27]}$$

(d) Ratios associated with the nose

$$\text{Ratio } 31 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[30]}{ED\ between\ L\_PT[27]\ and\ L\_PT[35]}$$

$$\text{Ratio } 32 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[30]}{ED\ between\ L\_PT[30]\ and\ L\_PT[35]}$$

$$\text{Ratio } 33 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[30]}{ED\ between\ L\_PT[27]\ and\ L\_PT[31]}$$

$$\text{Ratio } 34 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[30]}{ED\ between\ L\_PT[30]\ and\ L\_PT[31]}$$

$$\text{Ratio } 35 = \frac{ED\ between\ L\_PT[31]\ and\ L\_PT[35]}{ED\ between\ L\_PT[30]\ and\ L\_PT[35]}$$

$$\text{Ratio } 36 = \frac{ED\ between\ L\_PT[31]\ and\ L\_PT[35]}{ED\ between\ L\_PT[30]\ and\ L\_PT[31]}$$

$$\text{Ratio } 37 = \frac{ED\ between\ L\_PT[31]\ and\ L\_PT[35]}{ED\ between\ L\_PT[31]\ and\ L\_PT[33]}$$

$$\text{Ratio } 38 = \frac{ED\ between\ L\_PT[31]\ and\ L\_PT[35]}{ED\ between\ L\_PT[33]\ and\ L\_PT[35]}$$

$$\text{Ratio } 39 = \frac{ED\ between\ L\_PT[32]\ and\ L\_PT[34]}{ED\ between\ L\_PT[27]\ and\ L\_PT[32]}$$

$$\text{Ratio } 40 = \frac{ED\ between\ L\_PT[32]\ and\ L\_PT[34]}{ED\ between\ L\_PT[27]\ and\ L\_PT[34]}$$

(e) Ratios associated with the nose and the inner eye points

$$\text{Ratio } 41 = \frac{ED\ between\ L\_PT[30]\ and\ L\_PT[31]}{ED\ between\ L\_PT[39]\ and\ L\_PT[30]}$$

$$\text{Ratio } 42 = \frac{ED\ between\ L\_PT[30]\ and\ L\_PT[31]}{ED\ between\ L\_PT[39]\ and\ L\_PT[31]}$$

$$\text{Ratio } 43 = \frac{ED\ between\ L\_PT[30]\ and\ L\_PT[35]}{ED\ between\ L\_PT[42]\ and\ L\_PT[30]}$$

$$\text{Ratio } 44 = \frac{ED\ between\ L\_PT[30]\ and\ L\_PT[35]}{ED\ between\ L\_PT[42]\ and\ L\_PT[35]}$$

$$\text{Ratio } 45 = \frac{ED\ between\ L\_PT[48]\ and\ L\_PT[4]}{ED\ between\ L\_PT[3]\ and\ L\_PT[4]}$$

$$\text{Ratio } 46 = \frac{ED\ between\ L\_PT[48]\ and\ L\_PT[4]}{ED\ between\ L\_PT[48]\ and\ L\_PT[3]}$$

$$\text{Ratio } 47 = \frac{ED\ between\ L\_PT[48]\ and\ L\_PT[4]}{ED\ between\ L\_PT[48]\ and\ L\_PT[5]}$$

$$\text{Ratio } 48 = \frac{ED\ between\ L\_PT[4]\ and\ L\_PT[5]}{ED\ between\ L\_PT[48]\ and\ L\_PT[5]}$$

$$\text{Ratio } 49 = \frac{ED\ between\ L\_PT[54]\ and\ L\_PT[12]}{ED\ between\ L\_PT[54]\ and\ L\_PT[13]}$$

$$\text{Ratio } 50 = \frac{ED\ between\ L\_PT[54]\ and\ L\_PT[12]}{ED\ between\ L\_PT[12]\ and\ L\_PT[13]}$$

$$\text{Ratio } 51 = \frac{ED\ between\ L\_PT[54]\ and\ L\_PT[12]}{ED\ between\ L\_PT[54]\ and\ L\_PT[11]}$$

$$\text{Ratio } 52 = \frac{ED\ between\ L\_PT[11]\ and\ L\_PT[12]}{ED\ between\ L\_PT[54]\ and\ L\_PT[11]}$$

$$\text{Ratio } 53 = \frac{ED\ between\ L\_PT[48]\ and\ L\_PT[51]}{ED\ between\ L\_PT[48]\ and\ L\_PT[57]}$$

$$\text{Ratio } 54 = \frac{ED\ between\ L\_PT[48]\ and\ L\_PT[51]}{ED\ between\ L\_PT[57]\ and\ L\_PT[51]}$$

$$\text{Ratio } 55 = \frac{ED\ between\ L\_PT[54]\ and\ L\_PT[51]}{ED\ between\ L\_PT[57]\ and\ L\_PT[51]}$$

$$\text{Ratio } 56 = \frac{ED\ between\ L\_PT[54]\ and\ L\_PT[51]}{ED\ between\ L\_PT[54]\ and\ L\_PT[57]}$$

$$\text{Ratio } 57 = \frac{ED\ between\ L\_PT[48]\ and\ L\_PT[8]}{ED\ between\ L\_PT[48]\ and\ L\_PT[5]}$$

$$\text{Ratio } 58 = \frac{ED\ between\ L\_PT[48]\ and\ L\_PT[8]}{ED\ between\ L\_PT[5]\ and\ L\_PT[8]}$$

$$\text{Ratio } 59 = \frac{ED\ between\ L\_PT[54]\ and\ L\_PT[8]}{ED\ between\ L\_PT[54]\ and\ L\_PT[11]}$$

$$\text{Ratio } 60 = \frac{ED\ between\ L\_PT[54]\ and\ L\_PT[8]}{ED\ between\ L\_PT[11]\ and\ L\_PT[8]}$$

(f) Ratios associated with the jaw radiuses with L_PT [27] as the center point

$$\text{Ratio } 61 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[0]}{ED\ between\ L\_PT[0]\ and\ L\_PT[1]}$$

$$\text{Ratio } 62 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[0]}{ED\ between\ L\_PT[27]\ and\ L\_PT[1]}$$

$$\text{Ratio } 63 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[1]}{ED\ between\ L\_PT[1]\ and\ L\_PT[2]}$$

$$\text{Ratio } 64 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[1]}{ED\ between\ L\_PT[27]\ and\ L\_PT[2]}$$

$$\text{Ratio } 65 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[2]}{ED\ between\ L\_PT[2]\ and\ L\_PT[3]}$$

$$\text{Ratio } 66 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[2]}{ED\ between\ L\_PT[27]\ and\ L\_PT[3]}$$

$$\text{Ratio } 67 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[3]}{ED\ between\ L\_PT[3]\ and\ L\_PT[4]}$$

$$\text{Ratio } 68 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[3]}{ED\ between\ L\_PT[27]\ and\ L\_PT[4]}$$

$$\text{Ratio } 69 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[4]}{ED\ between\ L\_PT[4]\ and\ L\_PT[5]}$$

$$\text{Ratio } 70 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[4]}{ED\ between\ L\_PT[27]\ and\ L\_PT[5]}$$

$$\text{Ratio } 71 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[5]}{ED\ between\ L\_PT[5]\ and\ L\_PT[6]}$$

$$\text{Ratio } 72 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[5]}{ED\ between\ L\_PT[27]\ and\ L\_PT[6]}$$

$$\text{Ratio } 73 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[6]}{ED\ between\ L\_PT[6]\ and\ L\_PT[7]}$$

$$\text{Ratio } 74 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[6]}{ED\ between\ L\_PT[27]\ and\ L\_PT[7]}$$

$$\text{Ratio } 75 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[7]}{ED\ between\ L\_PT[7]\ and\ L\_PT[8]}$$

$$\text{Ratio } 76 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[7]}{ED\ between\ L\_PT[27]\ and\ L\_PT[8]}$$

$$\text{Ratio } 77 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[8]}{ED\ between\ L\_PT[8]\ and\ L\_PT[9]}$$

$$\text{Ratio } 78 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[8]}{ED\ between\ L\_PT[27]\ and\ L\_PT[9]}$$

$$\text{Ratio } 79 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[9]}{ED\ between\ L\_PT[9]\ and\ L\_PT[10]}$$

$$\text{Ratio } 80 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[9]}{ED\ between\ L\_PT[27]\ and\ L\_PT[10]}$$

$$\text{Ratio } 81 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[10]}{ED\ between\ L\_PT[10]\ and\ L\_PT[11]}$$

$$\text{Ratio } 82 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[10]}{ED\ between\ L\_PT[27]\ and\ L\_PT[11]}$$

$$\text{Ratio } 83 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[11]}{ED\ between\ L\_PT[11]\ and\ L\_PT[12]}$$

$$\text{Ratio } 84 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[11]}{ED\ between\ L\_PT[27]\ and\ L\_PT[12]}$$

$$\text{Ratio } 85 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[12]}{ED\ between\ L\_PT[12]\ and\ L\_PT[13]}$$

$$\text{Ratio } 86 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[12]}{ED\ between\ L\_PT[27]\ and\ L\_PT[13]}$$

$$\text{Ratio } 87 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[13]}{ED\ between\ L\_PT[13]\ and\ L\_PT[14]}$$

$$\text{Ratio } 88 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[13]}{ED\ between\ L\_PT[27]\ and\ L\_PT[14]}$$

$$\text{Ratio } 89 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[14]}{ED\ between\ L\_PT[14]\ and\ L\_PT[15]}$$

$$\text{Ratio } 90 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[14]}{ED\ between\ L\_PT[27]\ and\ L\_PT[15]}$$

$$\text{Ratio } 91 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[15]}{ED\ between\ L\_PT[15]\ and\ L\_PT[16]}$$

$$\text{Ratio } 92 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[15]}{ED\ between\ L\_PT[27]\ and\ L\_PT[16]}$$

$$\text{Ratio } 93 = \frac{ED\ between\ L\_PT[27]\ and\ L\_PT[16]}{ED\ between\ L\_PT[15]\ and\ L\_PT[16]}$$

After the facial features are extracted, each facial image's facial features alongside its ground truth age classification (that is, either adult or child) is saved in a pickle file.

4.4 Age group classification

The pickle file with the facial features extracted is now loaded into the classifier; in this case, KNN and the ground-truth age is converted to a binary problem (that is, child = 0 and adult = 1). Then, this data is divided into 80% for training and 20% for testing with the scikit-learn library's help.

4.4.1 Dimensionality reduction (PCA)

It is essential to reduce the number of extracted features to only the ones useful for classification; doing so tends to improve the proposed system's performance. This process is called dimensionality reduction. The PCA technique is used in this paper. Furthermore, I would shed more light on dimensionality reduction, PCA, and how PCA improves classification performance.

Dimensionality reduction can be defined as reducing the number of random variables being considered by obtaining a set of principal variables. It is divided into feature extraction and feature selection.

- The best features are selected from a set of features provided in the data set in feature selection. In this case, new features are not created; instead, the essential features from the given set of features are chosen. Therefore, reduced dimensions or features will be available to work with.
- Unlike in feature selection, feature extraction creates new features from the most important features, and the newly created features are not present in the original feature set.

Principal Component Analysis (PCA) is mainly the rotation of coordinate axes, chosen such that each successful axis captures or preserves as much variance as possible. It is a feature extraction technique of dimensionality reduction. It also is an unsupervised technique that does not consider labels.

Mathematically, PCA performs a linear transformation moving the original set of features to a principal component's new space.

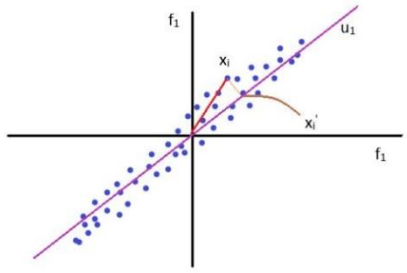


Fig. 10: Showing points plotted in the f_1 and f_2 feature spaces. Also, showing coordinate axes rotated to find unit vector, u_1. Source: [28]

u_1 = unit vector, $|u_1| = 1$

Projection of x_i on u_1 is x_1'

$D = \{x_i\}_{i=1}^n$, $D' = \{x_i'\}_{i=1}^n$

$x_i' = (u_1 . x_i) / |u_1|^2$ (projecting x_i on u_1)

since $|u_1|^2 = 1, x_1' = u_1^T . x_i$

The projection of the mean on the new feature space will be the mean in this feature space.

if $x_i = x_{mean}$

$x_{mean}' = u_1^T x_{mean}$

We have to find u_1 such that var{projection(x_i)} on u_1 is maximum.

$$var(u_1^T x_i) = \frac{1}{n}\sum_{i=1}^{n}(u_1^T x_i - u_1^T x_{mean})^2$$

Since X is column standardized so x_{mean} is 0, and ($u_1^T x_{mean}$) is also 0.

$$\text{So, } \max\left\{var\{x_i'\} = \frac{1}{n}\sum_{i=1}^{n}(u_1^T x_i)^2\right\} \text{ is the objective function}$$

Such that, $u_1^T u_1 = 1 = |u_1|^2$

Eigenvectors represent the directions in which most of the data is spread, given a standardized data matrix. Eigenvalues represent the amount of spread. The covariance matrix from our data matrix is calculated and using that covariance matrix, the eigenvectors and their corresponding eigenvalues are calculated.

- Standardization: It is the process of centering and scaling of our data. We can do row and column standardization, but we perform column standardization in general. It helps in dealing with different scale issues present in the dataset generally.

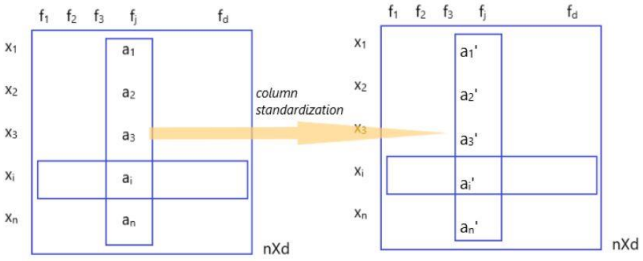


Fig. 11: Illustration of column standardization. Source: [28]

$$a_{mean} = \text{mean}\{a_i\}_{i=1}^n$$

$$\sigma = \text{std\_dev}\{a_i\}_{i=1}^n$$

$$a_i' = \frac{(a_i - a_{mean})}{\sigma}$$

The mean of the new points of the feature f_j is 0, and the variance is 1. So, the standardization moves the mean vector to the origin and expands the feature variance to 1.

- The covariance of the data matrix: It tells how two features vary together.

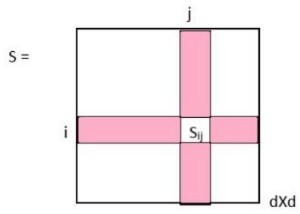


Fig. 12: Showing the covariance matrix, S of matrix X. Source: [28]

Where S is the covariance matrix of the matrix, X.

S_{ij} is the covariance of the ith feature and the jth feature

.

The formula given below is used to calculate the covariance between the two features, X and Y:

$$\text{cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu_x) * (y_i-\mu_y)$$

Where x_i and y_i are the values of the features X and Y, respectively

μ is the mean of the features

In the case, the matrix is standardized, μ = 0, and the formula becomes:

$$\text{cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i) * (y_i)$$

$$\text{cov}(f_i, f_j) = \frac{1}{n} f_i^T f_j$$

$$S_{dXd} = \frac{1}{n} X_{standardized}^T \; X_{standardized}$$

S is a square symmetric matrix. The covariance matrix's diagonal is the covariance of the feature with itself, which is nothing but variance. If the matrix was standardized, then the diagonal values of the covariance matrix become 1. The dXd shape of the matrix is as follows:

$$\text{shape}(X_{standardized}) = n \text{ X } d$$

$$\text{shape}(X_{standardized}^T) = d \text{ X } n$$

$$\text{shape}(X^T X) = d \text{ X } n \text{ X } n \text{ X } d = d \text{ X } d$$

We can calculate the eigenvalue and eigenvector using the covariance matrix as follows:

$$\mathbf{S}.\mathbf{v} = \lambda.\mathbf{v}$$

Where **S** is an n-by-n matrix,

v is a non-zero n-by-1 vector (eigenvector) and

λ is a scalar (eigenvalue of matrix **S**)

The equation can be rewritten as

$$\mathbf{S}.\mathbf{v}-\lambda.\mathbf{v} = 0$$

$$\mathbf{S}.\mathbf{v}-\lambda.\mathbf{I}.\mathbf{v} = 0$$

$$(\mathbf{S}.\mathbf{v}-\lambda.\mathbf{I}).\mathbf{v} = 0$$

Therefore,

Eigenvalues of S (covariance matrix) = $\lambda_1, \lambda_2, \ldots \lambda_d$

Eigenvectors of S (covariance matrix) = $v_1, v_2, \ldots v_d$

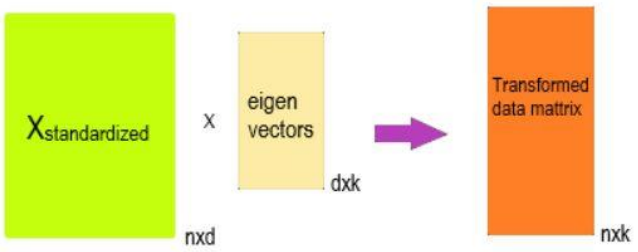


Fig. 13: Showing the transformed matrix with fewer dimensions (k) than the original matrix. Source: [28]

Once the eigenvectors, which are of the shape dx1, are found, the standardized data matrix is now converted into the new form (i.e., project the standardized points in the new feature space. For instance, out of d eigenvectors, top eigenvectors k are chosen. Each eigenvector is of shape, dx1, so when we put all the eigenvectors in one matrix, our matrix becomes the shape dxk. Now, we perform matrix multiplication of the X and the matrix of eigenvectors. We end up having a new matrix of shape nxk with the points projected on the new feature space.

To decide the best k when PCA is performed only for dimensionality reduction, The cumulative variance explained is checked by the number of vectors selected (k). This is also known as the percentage of variance explained, and its formula is given below:

$$\text{Percentage of variance explained} = \frac{\lambda i}{\sum_{i=1}^{d}\lambda i}$$

4.4.1 KNN classifier

To explain how the K – Nearest Neighbour (KNN) works, I would like to define some parameters like *x* to denote a *feature* (predictor) and *y* to denote the *target* (label) we are trying to predict. In this paper, *x* is ninety-three features, and y is either an adult or a child.

In KNN, a labeled dataset consisting of training observations (*x, y*) is given, and the relationship between *x* and *y* is captured. The primary goal is to learn a function *h: X → Y* so that given as unseen observation *x*, *h(x)* can confidently forecast the corresponding output *y*. Therefore, KNN is a supervised learning algorithm.

The KNN classifier is also a non-parametric and instance-based learning algorithm.

- It is non-parametric in the sense that it does not make explicit assumptions about the functional form of *h*, therefore, avoiding the dangers of misclassifying the underlying distribution of the data.
- It is also an instance-based learning algorithm because it does not explicitly learn a model; instead, it chooses to memorize the training instances used as 'knowledge' for the prediction phase.

In classification, the KNN algorithm forms a majority vote between the K most similar instances to a given 'unseen' observation. This similarity is calculated using a distance metric between two data points, mostly the Euclidean distance, which is given below:

$$d(x,x') = \sqrt{[(x1 - x'1)^2 + \cdots + (xn - x'n)^2]}$$

Other distance measures include the Manhattan and Hamming distance.

KNN classifier performs the following two steps given a positive integer K, an unseen observation *x*, and a metric similarity d:

- It runs through the entire dataset and computes *d* between *x* and each observation of the training. Let us call the K points in the training data closest to *x* the set A. To prevent tie situations

, K is usually odd.

- The conditional probability for each class is then estimated: the fraction of points in A with that given label. *I(x)* is the indicator function which evaluates to 1 when the argument *x* is true and 0 otherwise.

$$P\ (y = j \mid X = x) = \frac{1}{K}\sum_{i\epsilon A} I\ (y^{(i)} = j)$$

Finally, our input x gets assigned to the class with the most considerable probability.

*Note: The K in KNN is a hyperparameter that is intuitively picked to get the best possible fit for the dataset [27].

5. RESULTS AND DISCUSSION

The performance of the age estimation system is measured by accuracy and MAE. MAE is the measure of error between the estimated age and the ground truth age. Mathematically,

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{n} |\theta i' - \theta i|$$

Where θi is the ground truth age for the test image, i.

$\theta i'$ is the estimated age.

N is the total number of test images.

Accuracy is the frequency of the correctness of the classifier. Mathematically,

$$\text{Accuracy} = (\text{TP} + \text{TN}) / \text{T}$$

Where TP is the true positive rate of the system

TN is the true negative rate of the system.

T is the total test images.

The proposed system's age estimation results are shown in table 1, figures 14, 15, 16, and 17. The experiments were carried out using a robust method for trying to spot the pattern of the system. For each time the system is run, the best K for KNN with and without PCA, its accuracies, MAE, and best K-component for PCA are recorded. A pattern can be observed from the graphs in figures 16 and 17. The accuracy gradually tends to drop as the age progresses until 38 years (without PCA) and 35 years (with PCA), where it begins to rise again. Another thing that can be observed from the graphs is that PCA increases the system's accuracy and reduces its average MAE.

Now, this papers' primary objective is to classify facial images into two classes (adult/ child), and the best accuracy for this classification from table 1 is threshold age, 12 (that is, 0 – 11 years as a child and 12 – 80 years as an adult).

```
Confusion Matrix:
[[ 22  25]
 [  2 266]]
```

Fig. 14: Showing the confusion matrix of the proposed system at threshold age, 12.

	precision	recall	f1-score	support
0.0	0.92	0.47	0.62	47
1.0	0.91	0.99	0.95	268
accuracy			0.91	315
macro avg	0.92	0.73	0.79	315
weighted avg	0.91	0.91	0.90	315

Fig. 15: Showing the classification report of the proposed system at threshold age, 12 where 0 is child, and 1 is adult.

From figure 14, we see that the proposed system made 315 predictions at threshold age, 12. Out of these 315 test facial images, it predicted 24 facial images to be child faces and 291 facial images to be adult faces. Meanwhile, the ground-truth age classification is 47 child facial images and 268 adult facial images. Therefore, 25 child facial images were misclassified as adult faces, and 2 adult facial images were misclassified as child faces by the proposed system at threshold age, 12.

The misclassification of the facial images can be because of the irregularities and differences in the growth rate of the face and its features in males and females [16]. Other factors that could have caused misclassification in the proposed system are lifestyle, health, and genetics.

In figure 15, we see that the proposed system is less likely to misclassify an adult face from the precision score of the two classes. Therefore, the system is better off finding adult faces, as seen in the two classes' recall scores.

Below in table 1 are the experimental results where Ts, Tr, and M represent test accuracy, train accuracy, and MAE, respectively.

Threshold Age (in years)	Best K for KNN without PCA	Accuracy (%) without PCA and MAE	Best K-component for PCA	Best K for KNN with PCA	Accuracy (%) with PCA and MAE
0	-	Ts: 100% Tr: 100% M: 0.0	-	-	Ts: 100% Tr: 100% M: 0.0
1	3	Ts: 99.37 Tr: 99.52 M: 0.006	30	3	Ts: 99.05 Tr: 99.52 M: 0.009
2	5	Ts: 99.05 Tr: 99.13 M: 0.009	10	3	Ts: 99.05 Tr: 99.28 M: 0.009
3	3	Ts: 97.78 Tr: 98.89 M: 0.022	10	5	Ts: 98.10 Tr: 98.10 M: 0.019
4	3	Ts: 97.78 Tr: 98.89 M: 0.022	10	5	Ts: 97.46 Tr: 97.46 M: 0.025
5	9	Ts: 95.57 Tr: 96.2 M: 0.044	6	7	Ts: 96.52 Tr: 97.46 M: 0.035
6	9	Ts: 94.30 Tr: 95.40 M: 0.057	15	5	Ts: 95.25 Tr: 95.64 M: 0.047
7	5	Ts: 93.35 Tr: 95.09 M: 0.066	19	7	Ts: 94.62 Tr: 94.69 M: 0.054
8	5	Ts: 91.77 Tr: 94.45 M: 0.082	42	5	Ts: 93.35 Tr: 94.45 M: 0.066
9	5	Ts: 90.51 Tr: 93.11 M: 0.095	40	5	Ts: 91.77 Tr: 93.66 M: 0.082
10	5	Ts: 89.56 Tr: 92.79 M: 0.104	39	3	Ts: 90.51 Tr: 94.24 M: 0.095
11	3	Ts: 89.24 Tr: 94.06 M: 0.108	27	5	Ts: 90.51 Tr: 93.19 M: 0.095
12	5	Ts: 89.24 Tr: 92.08 M: 0.108	21	5	Ts: 91.43 Tr: 92.77 M: 0.086
13	5	Ts: 87.97 Tr: 90.97 M: 0.120	20	5	Ts: 88.92 Tr: 93.03 M: 0.111
14	5	Ts: 86.71 Tr: 90.25 M: 0.133	30	3	Ts: 88.61 Tr: 93.19 M: 0.114
15	7	Ts: 86.71 Tr: 89.38 M: 0.133	30	3	Ts: 88.92 Tr: 92.95 M: 0.111

16	5	Ts: 85.44 Tr: 89.38 M: 0.146	30	3	Ts: 87.34 Tr: 92.03 M: 0.127
17	7	Ts: 83.86 Tr: 87.8 M: 0.161	40	5	Ts: 86.71 Tr: 90.41 M: 0.133
18	7	Ts: 82.59 Tr: 87.24 M: 0.174	30	3	Ts: 85.44 Tr: 91.84 M: 0.146
19	9	Ts: 82.28 Tr: 87.24 M: 0.177	40	5	Ts: 85.44 Tr: 89.38 M: 0.146
20	9	Ts: 81.33 Tr: 85.18 M: 0.187	40	5	Ts: 84.18 Tr: 88.43 M: 0.158
21	9	Ts: 79.75 Tr: 83.84 M: 0.203	40	5	Ts: 82.28 Tr: 87.32 M: 0.177
22	9	Ts: 79.11 Tr: 83.6 M: 0.209	40	5	Ts: 81.96 Tr: 86.69 M: 0.180
23	9	Ts: 78.16 Tr: 83.52 M: 0.218	40	5	Ts: 81.01 Tr: 85.58 M: 0.19
24	9	Ts: 77.85 Tr: 82.17 M: 0.222	40	5	Ts: 80.38 Tr: 84.47 M: 0.196
25	9	Ts: 75.95 Tr: 81.22 M: 0.241	40	5	Ts: 78.48 Tr: 83.91 M: 0.215
26	3	Ts: 75 Tr: 86.93 M: 0.25	30	5	Ts: 78.80 Tr: 83.60 M: 0.212
27	9	Ts: 75.63 Tr: 81.30 M: 0.244	30	5	Ts: 78.16 Tr: 83.60 M: 0.218
28	9	Ts: 75.63 Tr: 80.74 M: 0.244	30	5	Ts: 77.85 Tr: 83.12 M: 0.222
29	9	Ts: 75.63 Tr: 79.40 M: 0.244	31	9	Ts: 77.22 Tr: 79.64 M: 0.228
30	7	Ts: 75.32 Tr: 80.19 M: 0.247	21	3	Ts: 76.58 Tr: 84.94 M: 0.234
31	9	Ts: 74.68 Tr: 78.53 M: 0.253	21	3	Ts: 75.95 Tr: 84.39 M: 0.241
32	5	Ts: 74.05 Tr: 80.98 M: 0.259	21	3	Ts: 74.68 Tr: 83.91 M: 0.253
33	5	Ts: 70.57 Tr: 80.43 M: 0.294	21	3	Ts: 74.05 Tr: 84.31 M: 0.259
34	7	Ts: 70.57 Tr: 78.29 M: 0.294	23	7	Ts: 73.65 Tr: 81.02 M: 0.263
35	3	Ts: 69.62 Tr: 83.36 M: 0.304	23	3	Ts: 70.25 Tr: 84.07 M: 0.297
36	7	Ts: 69.94 Tr: 77.42 M: 0.301	19	9	Ts: 74.37 Tr: 78.53 M: 0.256
37	3	Ts: 69.94 Tr: 82.49 M: 0.301	19	9	Ts: 75 Tr: 78.21 M: 0.25
38	9	Ts: 68.99 Tr: 76.86 M: 0.310	19	9	Ts: 75 Tr: 78.61 M: 0.25
39	9	Ts: 71.20 Tr: 76.47 M: 0.288	19	9	Ts: 76.27 Tr: 78.68 M: 0.237
40	9	Ts: 73.10 Tr: 77.26 M: 0.269	19	9	Ts: 77.53 Tr: 79.08 M: 0.225
41	9	Ts: 74.68 Tr: 78.05 M: 0.253	19	9	Ts: 78.80 Tr: 79.40 M: 0.212
42	9	Ts: 74.05 Tr: 77.89 M: 0.259	19	9	Ts: 77.22 Tr: 79.32 M: 0.228
43	9	Ts: 75 Tr: 78.05 M: 0.25	15	9	Ts: 76.27 Tr: 77.42 M: 0.237
44	9	Ts: 75.95 Tr: 79.08 M: 0.241	24	7	Ts: 78.16 Tr: 81.06 M: 0.218
45	5	Ts: 78.80 Tr: 81.22 M: 0.212	25	9	Ts: 79.11 Tr: 80.43 M: 0.209
46	5	Ts: 79.43 Tr: 80.98 M: 0.206	16	7	Ts: 79.11 Tr: 81.30 M: 0.209
47	3	Ts: 78.16 Tr: 85.97 M: 0.218	22	5	Ts: 80.06 Tr: 81.77 M: 0.199
48	3	Ts: 80.70 Tr: 86.69 M: 0.195	22	5	Ts: 80.06 Tr: 82.09 M: 0.199
49	5	Ts: 81.01 Tr: 82.65 M: 0.19	22	5	Ts: 81.33 Tr: 82.49 M: 0.187
50	3	Ts: 82.28 Tr: 86.77 M: 0.177	26	3	Ts: 83.23 Tr: 85.26 M: 0.168
51	3	Ts: 81.33 Tr: 86.93 M: 0.187	15	3	Ts: 84.18 Tr: 86.29 M: 0.158
52	3	Ts: 81.65 Tr: 87.27 M: 0.184	15	3	Ts: 84.81 Tr: 86.53 M: 0.152
53	3	Ts: 82.28 Tr: 87.80 M: 0.177	21	3	Ts: 85.13 Tr: 85.74 M: 0.149
54	3	Ts: 83.23 Tr: 88.11 M: 0.168	22	3	Ts: 85.44 Tr: 86.29 M: 0.146
55	3	Ts: 83.54 Tr: 88.91 M: 0.165	24	3	Ts: 86.71 Tr: 87.64 M: 0.133
56	3	Ts: 85.44 Tr: 89.62 M: 0.146	21	3	Ts: 87.34 Tr: 87.88 M: 0.127
57	5	Ts: 87.34 Tr: 88.83 M: 0.127	26	3	Ts: 88.29 Tr: 88.91 M: 0.117
58	3	Ts: 90.19 Tr: 92 M: 0.098	42	3	Ts: 91.14 Tr: 91.84 M: 0.089
59	3	Ts: 90.19 Tr: 92.16 M: 0.098	39	3	Ts: 91.46 Tr: 91.68 M: 0.085
60	3	Ts: 90.82 Tr: 92.87 M: 0.092	42	3	Ts: 91.77 Tr: 92.23 M: 0.082
61	3	Ts: 90.82 Tr: 93.34 M: 0.092	39	3	Ts: 91.46 Tr: 92.31 M: 0.085
62	5	Ts: 92.41 Tr: 92.87 M: 0.076	19	3	Ts: 92.41 Tr: 93.03 M: 0.076
63	3	Ts: 92.72 Tr: 93.74 M: 0.073	15	3	Ts: 93.35 Tr: 93.82 M: 0.066
64	5	Ts: 92.72 Tr: 93.90 M: 0.073	15	3	Ts: 93.67 Tr: 94.37 M: 0.063
65	5	Ts: 93.35 Tr: 94.06 M: 0.066	19	3	Ts: 93.67 Tr: 94.61 M: 0.063
66	5	Ts: 93.67 Tr: 94.06 M: 0.063	19	3	Ts: 94.30 Tr: 94.53 M: 0.057
67	5	Ts: 93.99	19	3	Ts: 94.62

		Tr: 94.37 M: 0.060			Tr: 94.93 M: 0.054
68	3	Ts: 94.30 Tr: 95.80 M: 0.057	16	3	Ts: 95.25 Tr: 95.48 M: 0.047
69	3	Ts: 95.25 Tr: 96.04 M: 0.047	10	3	Ts: 95.25 Tr: 95.80 M: 0.047
70	3	Ts: 96.20 Tr: 96.43 M: 0.038	3	3	Ts: 95.57 Tr: 95.88 M: 0.044
71	3	Ts: 95.89 Tr: 96.51 M: 0.041	9	3	Ts: 95.89 Tr: 96.35 M: 0.041
72	3	Ts: 96.52 Tr: 96.83 M: 0.035	9	3	Ts: 96.20 Tr: 96.67 M: 0.038
73	3	Ts: 96.84 Tr: 96.91 M: 0.032	9	3	Ts: 96.52 Tr: 96.91 M: 0.035
74	3	Ts: 97.15 Tr: 96.99 M: 0.028	9	3	Ts: 97.15 Tr: 97.31 M: 0.028
75	3	Ts: 97.47 Tr: 97.31 M: 0.025	9	3	Ts: 97.47 Tr: 97.62 M: 0.025
76	3	Ts: 98.10 Tr: 97.70 M: 0.019	9	3	Ts: 97.47 Tr: 97.94 M: 0.025
77	3	Ts: 98.10 Tr: 98.34 M: 0.019	15	3	Ts: 98.42 Tr: 98.57 M: 0.016
78	3	Ts: 98.42 Tr: 98.97 M: 0.016	15	3	Ts: 98.42 Tr: 99.13 M: 0.016
79	3	Ts: 98.73 Tr: 99.45 M: 0.013	15	3	Ts: 98.73 Tr: 99.45 M: 0.013
80	3	Ts: 99.68 Tr: 99.52 M: 0.003	10	3	Ts: 99.37 Tr: 99.60 M: 0.006

Table 1: Showing experimental results where Ts, Tr, and M represent test accuracy, train accuracy, and MAE, respectively.

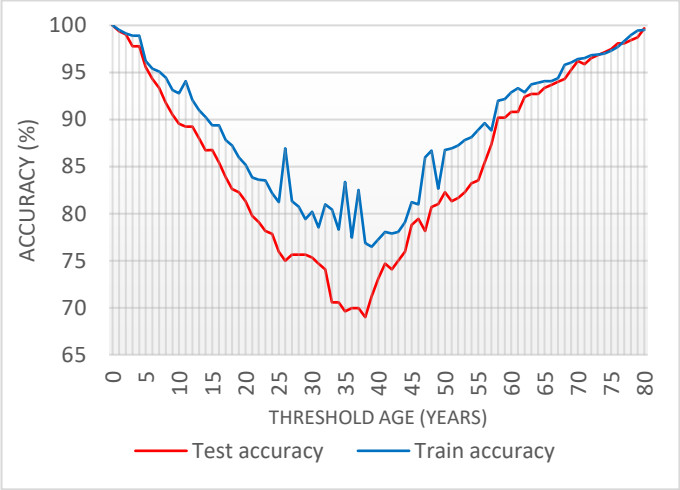


Fig. 16: Graph showing the proposed system's performance before PCA is applied

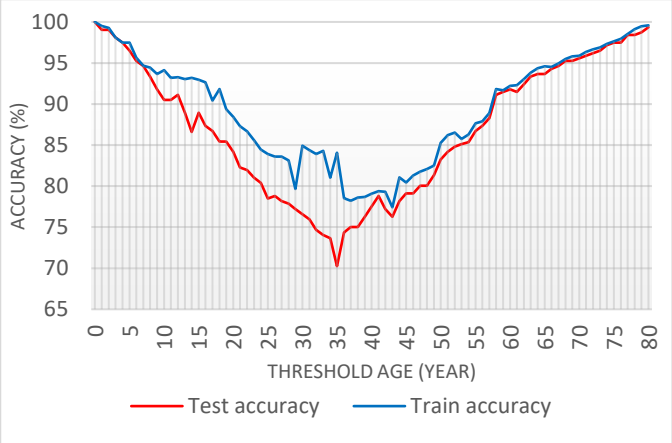


Fig. 17: Graph showing the proposed system's performance after PCA is applied

6. CONCLUSION AND RECOMMENDATION

In this paper, a novel feature set used to classify facial images for the age detection task was developed. The experiments carried out revealed that the ninety-three (93) facial feature set works best trying to classify facial images into adult and child at threshold age, 12 years with 91.43% accuracy, and 0.086 MAE.

For future works, I will recommend that the proposed feature set be explored more using a system structure, which is somewhat like the one used in [7]. That is, there should first be some sort of gender recognition, then gender-specific age grouping before age estimation because research has shown that the maturity rate of some important facial features differs in males and females [4, 15, 16].

7. ACKNOWLEDGEMENTS

I would like to acknowledge my supervisor, Dr. Neil Cooke, for his patience and guidance through every stage of my project work and for inspiring my interest in computer vision.

I would like to express my gratitude and appreciation to my parents, Prof & Mrs. David Iliya Malgwi, whose love, guidance, support, and encouragement have been invaluable throughout this study.

Finally, I would like to extend my gratitude towards the rest of my family, family friends, and friends for the total support you have shown me during my master's program.

8. REFERENCES

[1] Kwon, Y. and Lobo, N., 1999. Age Classification from Facial Images. *Computer Vision and Image Understanding*, 74(1), pp.1-21.

[2] Izadpanahi, S. and Toygar, O., 2012. Geometric feature based age classification using facial images. *IET Conference on Image Processing (IPR 2012)*, pp.1-5.

[3] Alom, M., Piao, M., Islam, M., Kim, N. and Park, J., 2012. Optimized Facial Features-based Age Classification. *World Academy of Science.*

Engineering and Technology Conference, 6, pp.319-324.

[4] Tharwat, A., Ghanem, A. and Hassanien, A., 2013. Three different classifiers for facial age estimation based on K-nearest neighbor. *2013 9th International Computer Engineering Conference (ICENCO)*, Giza, 2013, pp. 55-60.

[5] Abbas, A. and Kareem, A., 2018. Intelligent Age Estimation From Facial Images Using Machine Learning Techniques. *Iraqi Journal of Science*, 59(2A), pp.724-732.

[6] Agarwal, M. and Jain, S., 2018. Image Classification for Underage Detection in Restricted Public Zone. *2018 IEEE 8th International Advance Computing Conference (IACC)*, pp.355-359.

[7] Liu, K. and Liu, T., 2019. A Structure-Based Human Facial Age Estimation Framework Under a Constrained Condition. *IEEE Transactions on Image Processing*, 28(10), pp.5187-5200.

[8] Mullins, J., 2012. Age progression and regression. *Craniofacial Identification*, pp.68-75.

[9] Farazdaghi, E., Majidzadeh, F. and Nait-Ali, A., 2018. Facial Rejuvenation Modeling. *Biometrics under Biomedical Considerations*, pp.71-96.

[10] Ferková, Z., Urbanová, P., Černý, D., Žuži, M. and Matula, P., 2018. Age and gender-based human face reconstruction from single frontal image. *Multimedia Tools and Applications*, 79(5-6), pp.3217-3242.

[11] Razalli, H. and Alkawaz, M., 2019. Real-Time Face Tracking Application with Embedded Facial Age Range Estimation Algorithm. *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, Shah Alam, Malaysia, 2019, pp. 471-476.

[12] Grd, P., 2013. Introduction to Human Age Estimation Using Face Images. *Research Papers Faculty of Materials Science and Technology Slovak University of Technology*, 21(Special-Issue), pp.24-30.

[13] Shejul, A., Kinage, K. and Reddy, B., 2017. Comprehensive review on facial based human age estimation. *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, Chennai, 2017, pp. 3211-3216.

[14] Atallah, R., Kamsin, A., Ismail, M., Abdelrahman, S. and Zerdoumi, S., 2018. Face Recognition and Age Estimation Implications of Changes in Facial Features: A Critical Review Study. *IEEE Access*, 6, pp.28290-28304.

[15] Al-Shannaq, A. and Elrefaei, L., 2019. Comprehensive Analysis of the Literature for Age Estimation From Facial Images. *IEEE Access*, 7, pp.93229-93249.

[16] Liu, C. and Wilkinson, C., 2020. A guided manual method for juvenile age progression using digital images. *Forensic Science International*, 308, p.110170.

[17] Dalal, N. and Triggs, B., 2005. Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1(01), pp.886-893.

[18] Saravanan, C., 2010. Color Image to Grayscale Image Conversion. *2010 Second International Conference on Computer Engineering and Applications*, pp.196-199.

[19] Felzenszwalb, P., Girshick, R., McAllester, D. and Ramanan, D., 2010. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), pp.1627-1645.

[20] Kazemi, V. and Sullivan, J., 2014. One millisecond face alignment with an ensemble of regression trees. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp.1867-1874.

[21] Fu, Y., 2014. *FG-NET Data By Yanwei Fu*. [online] Yanweifu.github.io. Available at: <https://yanweifu.github.io/FG_NET_data/> [Accessed 22 May 2020].

[22] Cheng, J., 2020. *Jingchuncheng/All-Age-Faces-Dataset*. [online] GitHub. Available at: <https://github.com/JingchunCheng/All-Age-Faces-Dataset> [Accessed 29 July 2020].

[23] King, D., 2018. *Davisking/Dlib*. [online] GitHub. Available at: <https://github.com/davisking/dlib> [Accessed 18 June 2020].

[24] Rosebrock, A., 2019. *Jrosebr1/Imutils*. [online] GitHub. Available at: <https://github.com/jrosebr1/imutils> [Accessed 18 June 2020].

[25] King, D., 2015. *Davisking/Dlib-Models*. [online] GitHub. Available at: <https://github.com/davisking/dlib-models> [Accessed 2 July 2020].

[26] Tutorialspoint, 2020. *Grayscale to RGB Conversion - Tutorialspoint*. [online] Tutorialspoint.com. Available at: <https://www.tutorialspoint.com/dip/grayscale_to_rgb_conversion.htm#:~:text=So%20the%20new%20equation%20that,and%20Blue%20has%20contributed%2011%25.> [Accessed 13 July 2020].

[27] Zakka, K., 2020. A Complete Guide To K-Nearest-Neighbors With Applications In Python And R. [online] Kevin Zakka's Blog. Available at: <https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/> [Accessed 24 July 2020].

[28] Singh, A., 2020. Dimensionality Reduction and Visualization Using PCA (Principal Component Analysis). [online] Medium. Available at: <https://medium.com/@ashwin8april/dimensionality-reduction-and-visualization-using-pca-principal-component-analysis-8489b46c2ae0> [Accessed 27 July 2020].

[29] Arora, M., 2020. Feature Extraction-Principal Component Analysis. [online] Medium. Available at: <https://medium.com/@mansiarora_20448/feature-extraction-principal-component-analysis-a10705b330ce> [Accessed 28 July 2020].

[30] Erik Cheever, S., 2020. Eigenvalues And Eigenvectors. [online] Lpsa.swarthmore.edu. Available at: <https://lpsa.swarthmore.edu/MtrxVibe/EigMat/MatrixEigen.html> [Accessed 28 July 2020].

[31] Patel, K., 2020. [online] Kevinpatel.me. Available at: <https://kevinpatel.me/img/blog-details/face-landmarks/landmarks.webp> [Accessed 8 August 2020].