AUTOMATED DENTAL AESTHETICS WITH MACHINE LEARNING

by

Ashwinee Mehta

December, 2022

Director of Thesis: Nic Herndon, PhD

Major Department: Computer Science

While dentures contribute an important role in restoring the dental and facial structure, facial aesthetics are an equally important consideration during the restoration of a patient with missing upper, lower, or both upper and lower teeth for elevating the treatment outcomes. Each denture is tailor-made for every patient, meaning the dental technician takes the impressions and measurements of the patient to make a perfect functional fitting denture. The current denture design workflow does not systematically include the aesthetic factors, patient's pre-treatment facial shape and in-progress denture design visualizations, instead relying on discussing mockups with the patients during appointments. This results into waiting for the final denture fitting on the patient to evaluate the final denture aesthetics. In this research, we plan to develop and validate some facial aesthetic proportion techniques that are used in the current denture design workflow. Given a frontal image of a person with missing teeth or a collapsed face, the proposed method will automatically generate an image of the patient, with teeth, while restoring the patient's face shape. This will assist the dental technicians to choose the best aesthetically fitting denture model, its size and position based on the current state of the face. Towards this goal, the method will automatically identify several facial landmarks, classify the patient's facial shape for easy selection of the denture model, and automatically create a three-dimensional denture design by using the patient's frontal and side-view images, as well as im-

ages captured from inside of the mouth. The goal of the research is to streamline the denture design process by considering the facial and teeth aesthetics, to enable denture-in-progress visualizations that avoid the end-moment denture refinements, with a simple graphical user interface that is easy to use by dental technicians.

AUTOMATED DENTAL AESTHETICS WITH MACHINE LEARNING

A Thesis

Presented to The Faculty of the Department of Computer Science

East Carolina University

In Partial Fulfillment of the Requirements for the Degree

Master of Science in Computer Science

by

Ashwinee Mehta

December, 2022

AUTOMATED DENTAL AESTHETICS WITH MACHINE LEARNING

by

Ashwinee Mehta

APPROVED BY:

DIRECTOR OF THESIS: _____

Nic Herndon, PhD

COMMITTEE MEMBER: _____

Rui Wu, PhD

COMMITTEE MEMBER: _____

Venkat Gudivada, PhD

CHAIR OF THE DEPARTMENT

OF COMPUTER SCIENCE: _____

Venkat Gudivada, PhD

DEAN OF THE

GRADUATE SCHOOL: _____

Kathleen Cox, PhD

Table of Contents

LIST OF FIGURES

Chapter 1

Introduction

Dentures are artificial teeth and gums that are designed for replacing lost or removed natural teeth to support cheeks and lips and restore the smile. Loss of human teeth causes the oral bone structure to start to perish and human face to collapse. Face collapse only affects the lower third of the face. Dentures decrease the rate at which the oral bones disintegrate. Sometimes, patients may think that dentures will be uncomfortable with respect to its size, fit and bite factors. Also, when the patients with a facial collapse use dentures, their face structure changes. The current denture workflow does not include considering aesthetic factors for designing dentures and relies only on discussing mockups with the patients. The actual denture fit and its effect on the facial structure depends on the trying the designed denture.

Main contribution: This dissertation presents a machine learning based validation of facial aesthetic proportion techniques that are used in the existing denture design workflow. We first explain the assumption of the neoclassical vertical canon and how it is being used for defining different proportions of the face, facial surgeries and dental reconstruction procedures. We then propose a machine learning based approach to validate the neoclassical vertical canon before blindly applying these recommended neoclassical formulae. With the proposed method, it is evident from the results that the three facial thirds are not equal. Thus it is recommended not to use

the assumption that the face is divided into equal thirds while designing dentures for patients.

This dissertation also proposes a method that focuses towards the goal of finding a similar non-collapsed face to a given collapsed face for reconstructing the lower third of the collapsed face before designing the denture. The process of facial restoration needs a similar face which can be used as a reference for comparing the facial shape proportions to convert the collapsed face into non-collapsed facial shape. Inclusion and evaluation of facial aesthetics is important while planning for facial or dental reconstruction treatment. The proposed method is based on Python's Deep Learning Face Recognition which is built using Dlib for finding similar non-collapsed faces to a given collapsed face from a diverse target dataset of human population. By using the proposed method, one can find multiple similar images and use them for facial reconstruction for automatic denture designing.

Chapter 2

Automated Neoclassical Vertical Canon Validation in Human Faces

2.1 Problem Description

The face is one of the most important factors affecting the physical appearance of a person. Different facial proportions can be used for measuring the facial attractiveness, for recommending hairstyles, fashion jewelry, eyeglasses, etc. The measurement of facial attractiveness is also applicable in cosmetics, orthodontics and plastic surgery. Dental practitioners take into consideration the different facial proportions in order to create a denture of suitable shape, size, and position.

The neoclassical canons used for proportional evaluation of the face were developed in the 17^{th} and 18^{th} centuries. These canons were based on the assumption that certain fixed ratios existed between different parts of a human face. They are still used to define the proportions between various areas of the head and face. One of these eight defined neoclassical canons is the vertical canon, which states that the face is divided into three equal sections. The first section is from the top of the forehead (Trichion), to the bridge of the nose (Glabella), as shown in Figure 2.1, the second section is from the bridge of the nose to the base of the nose (Subnasale), and the third section is from base of the nose to the chin (Menton). Trichion, Glabella, Subnasale and Menton are the anthropometric landmarks that are identified before taking the measurements of the facial thirds. The vertical canon is widely used in

facial surgeries and dental reconstruction procedures.

[9] first investigated the applicability of the neoclassical facial canons in young North American Caucasian adults. Following this, the canons were also validated on several other population groups including Nigerians, African-Americans, Turkish, Vietnamese, Thai, and Chinese individuals. These studies were performed by adopting the standard anthropometric methods and the measurements were obtained using anthropometric tools like millimetric compass, sliding calipers, etc. Some studies have used images pre-annotated with the anthropometric landmarks.

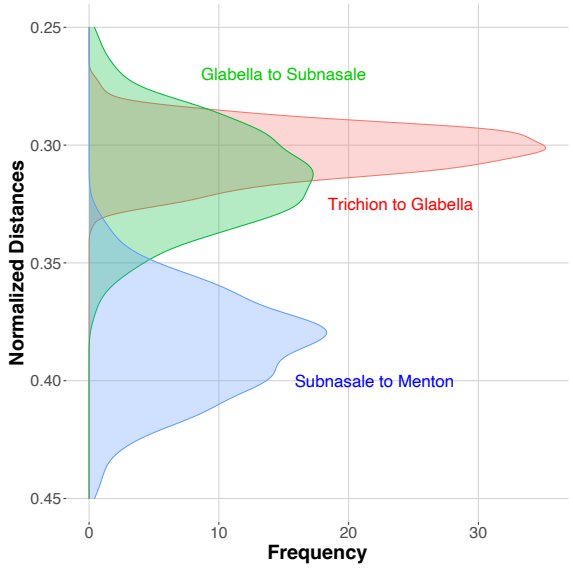
Missing teeth with age causes a person's face to collapse. While fixing the patient's teeth, it is also important to consider restoring the patient's facial shape. With a collapsed face, only the bottom one third of the face, i.e., from Subnasale to Menton is affected and needs to be restored. This facial restoration also needs some reference for comparing the facial shape proportions to convert the collapsed face into normal facial shape. Inclusion and evaluation of facial aesthetics is important while planning for facial or dental reconstruction treatment. Many clinical textbooks and journal articles recommend to use these neoclassical canons for evaluating the aesthetics. However, before blindly applying these recommended neoclassical formulae, it is important to validate them. With the advancements in technology, it is no longer required to use the traditional anthropometric tools to take measurements from the human face. We can use machine learning to train a model to automatically identify the different anthropometric landmarks of the human face and thus avoid the need for direct contact with patients. The objective of this study is to verify the vertical canon by using machine learning to eliminate the need to take the measurements manually using different anthropometric tools.

All the canon validation methods that have been proposed have used different physical instruments [15, 5, 1, 8] and software applications for taking measurements

(a) The neoclassical vertical canon states that the face is divided into three equal thirds.

(b) To test this hypothesis we used the Dlib-81 library to automatically place facial landmarks on an image.

(c) Out of the landmarks generated by the Dlib-81 library we used the ten circled landmarks to place the four lines.



(d) The density distributions of the three distances using annotated images from LFW, MUCT, and CUHK datasets seem to invalidate this hypothesis.

Figure 2.1: Facial landmarks are used to automatically place the lines for Trichion, Glabella, Subnasale, and Menton. These lines were used in one of the neoclassical canons, the vertical canon, to surmise that each face is divided into three equal sections. However, evidence shows that these distances are not equal.

of the face [7]. Some of the proposed techniques have used a ready-made database that had images with the anthropometric landmarks annotated [24, 20]. However, none of the techniques have used automated tools for getting the measurements and validating this canon. These techniques are discussed in Section 2.5.

Our proposed method used large volume of photos available online, annotated them with automated tools, and verified this hypothesis. We have performed the automatic validation of the vertical canon by annotating the images from three freely available image databases using machine learning. By using the proposed method, one can validate the vertical canon automatically without the need to use traditional anthropometric tools or instruments directly on the patient.

2.2 Experimental Design

The goal of our experiment is to automatically validate the vertical canon by annotating the images from three freely available image databases using machine learning. We tested the applicability of the vertical neoclassical canon on the facial images collected from three freely available datasets: Labeled Faces in the Wild (LFW) face database [11], the Milborrow / University of Cape Town (MUCT) face database [18], and the Chinese University of Hong Kong student database [28]. The LFW is a database of face photographs designed for studying the problem of unconstrained face recognition. The data set contains 13,233 facial images of 5749 individuals collected from the web. All the images in the LFW database have a resolution of 250 x 250 pixels. The MUCT face database consists of 3,755 facial images of 276 individuals. The individuals were sampled from students, parents attending graduation ceremonies, high school teachers attending a conference, and employees of the university at the University Of Cape Town campus in December 2008. This diverse population includes a wide range of

subjects, with approximately equal numbers of males and females, and a cross section of ages and races. All the images in the MUCT database have a resolution of 480 x 640 pixels. The CUHK database is for research on face sketch synthesis and face sketch recognition, and consists of 188 facial images of 188 individuals. All the images in the CUHK database have a resolution of 1024 x 768 pixels. All the facial images were not labeled with any anthropometric landmarks. To perform the validation, our experimental workflow had the following steps:

2.2.1 Annotate Images with Facial Anthropometric Landmarks

The first step was to annotate these images with facial anthropometric landmarks. We initially evaluated the Dlib's 68-point facial landmark detector [14], the most popular facial landmark detector. It can find 68 different facial landmark points including chin and jaw line, eyebrows, nose, eyes and lips as shown in Figure 2.2. In our preliminary work we determined that this library does not provide facial landmarks for the forehead. Therefore, we used an extended version of this library, 81 Facial Landmarks Shape Predictor[1], which provides 13 additional landmarks that delineate the forehead. Not all the landmarks generated are needed to get the measurements of the thirds of the face. Out of all the 81 landmarks, we used only the following landmarks, as shown in Figure 2.1c:

- 69 and 72, for the left- and right-forehead, respectively. These landmarks were used for the placement of the Trichion line.
- 77, 17, 26, and 78, for the left-temple, left-exterior eyebrow, right-exterior eyebrow, and right-temple, respectively. These landmarks were used for the placement of the Glabella line. This is the only line for which we used four landmarks, since none of the 81 landmarks are placed at the position for the bridge of the nose. Therefore, we used the mid-vertical distance between landmarks 77 and 17, along with the mid-vertical distance between landmarks 26 and 78, as the

[1]Available at `https://github.com/codeniko/shape_predictor_81_face_landmarks`.

anchors for the Glabella line.

- 31 and 35, for the left- and right-base of the nose, respectively. These landmarks were used for the Subnasale line.
- 7 and 9, for the left- and right-side of the chin, respectively. These landmarks were used for the Menton line.

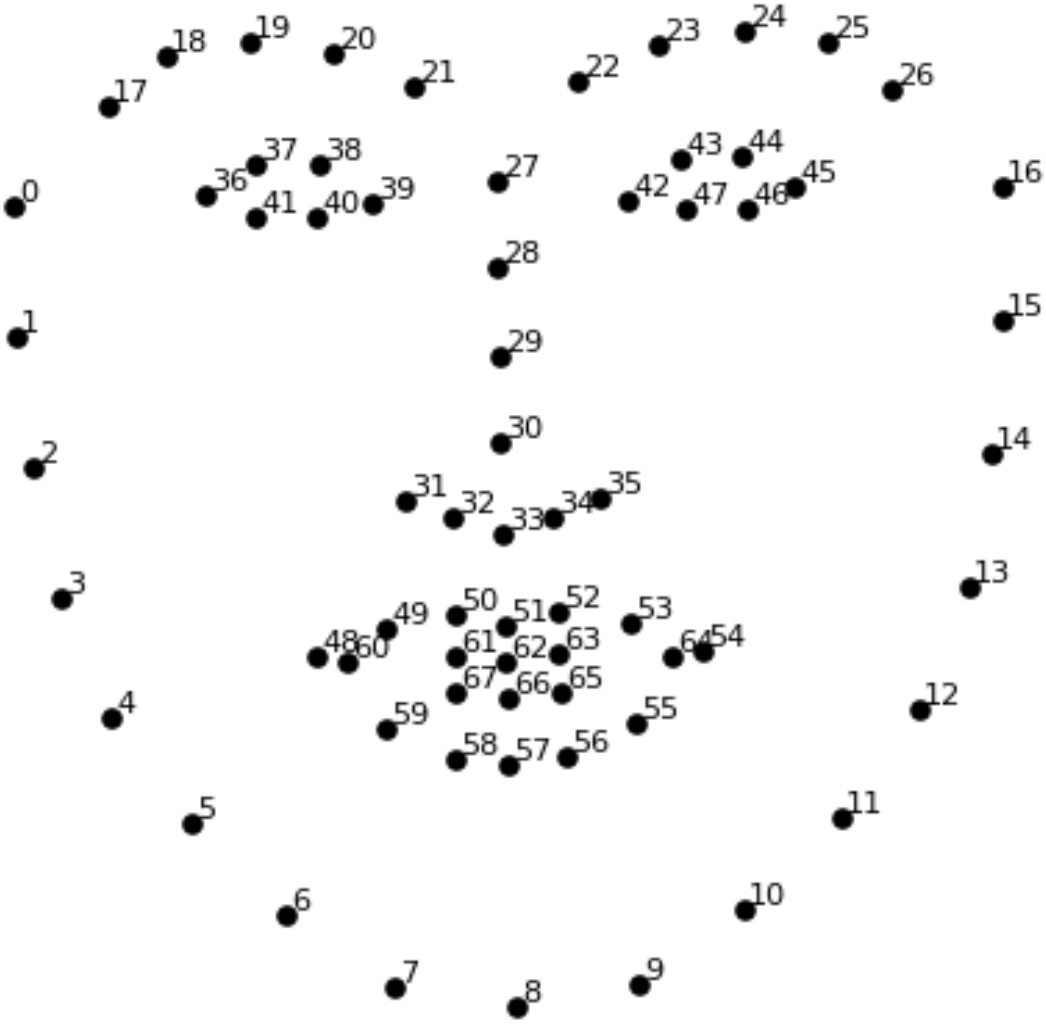


Figure 2.2: The set of 68-points detected by the pre-trained Dlib shape_predictor_68.

2.2.2 Assess the Annotated Images for Correct and Incorrect Landmarks

The second step was to visually inspect the placement of the four lines mentioned above, on all the images from the three datasets. During this manual inspection we noticed that the predictor placed the landmarks in the correct positions in images that had eyeglasses, beard, bald heads, hats, as well as for people of different age groups and races, and to different background colors and patterns. However, some of the lines were not correctly placed for images that did not have a front profile view. For some of these images, the 81 Facial Landmarks Shape Predictor misplaced some of the landmarks, which led to the incorrect placement of the Trichion, Glabella, Subnasale and Menton lines, as shown in Figure 2.3.

2.2.3 Select Images with Correctly Positioned Landmarks

Therefore, to get a correct and unbiased measurement of the thirds of the face, in the third step we planned to select about 500 images, that had correctly positioned landmarks. We identified 464 images from the LFW dataset, 86 images from the

Figure 2.3: Incorrect placement of landmarks on some images from the MUCT database due to shift in face positions. The left and middle images have the Glabella line misplaced, and the right image has the Glabella and Menton lines misplaced.

MUCT dataset, and 29 images from the CUHK dataset (for a total of 579 images), that had the correct automatic placement of the four lines.

2.2.4 Calculate Facial Distances of the Thirds of the Face

We used these images, in the fourth step to calculate the three facial distances: between Trichion and Glabella, between Glabella and Subnasale, and between Subnasale and Menton. The initial distance measurements were taken in number of vertical pixels between each two lines. Since we used images from all the three image databases, we had to take into consideration the fact that the images were provided with different resolutions. Therefore, we performed normalization of all the three distance values using Equation 2.1.

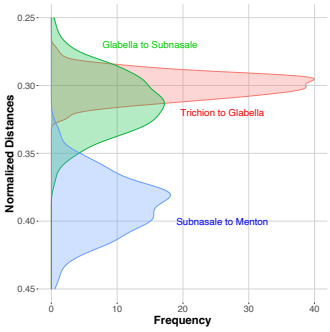
$$d'_i = \frac{d_i}{\sum_{j=1}^{3} d_j} \tag{2.1}$$

where d'_i is the normalized distance of d_i, and d_j is one of the three distances.

2.2.5 Plot Density Distributions of the Three Facial Distances

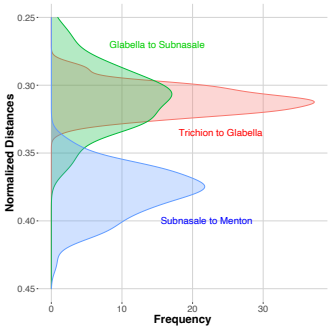
In the final step we used these normalized distances to generate density plots for the three distances. These are shown in Figure 2.1d for the 579 combined images, and in Figures 2.4b, 2.4d, and 2.4f, for the 464 images from the LFW database, the 86 images from the MUCT database, and the 29 images from the CUHK database, respectively.

(a) Sample image from the LFW dataset. The images in this dataset have a size of 250 x 250 pixels.

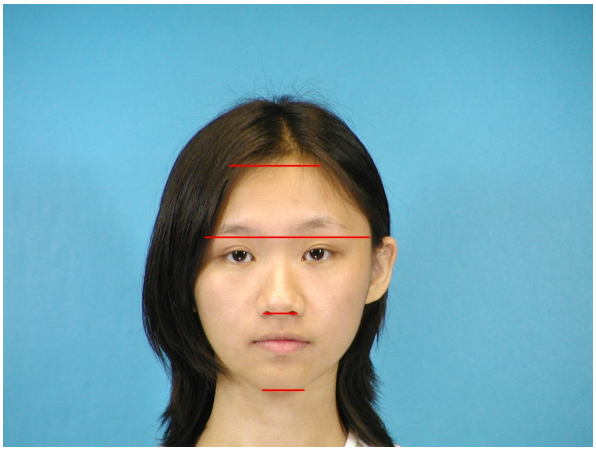


(b) The density distributions of the three distances for the 464 images used from the LFW dataset.

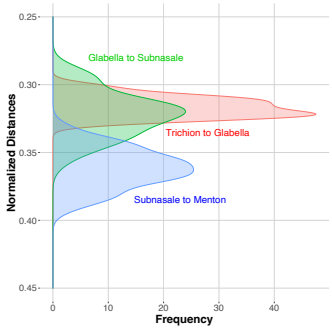
(c) Sample image from the MUCT dataset. The images in this dataset have a size of 480 x 640 pixels.



(d) The density distributions of the three distances for the 86 images used from the MUCT dataset.



(e) Sample image from the CUHK dataset. The images in this dataset have a size of 1024 x 768 pixels.



(f) The density distributions of the three distances for the 29 images used from the CUHK dataset.

Figure 2.4: Examples of images used in this analysis, shown to scale, from the following datasets: LFW, MUCT, and CUHK, along with their corresponding distance distributions. The distributions for each dataset indicate that the neoclassical vertical cannon is not valid.

2.3 Machine Learning Methods Used

2.3.1 Libraries

Dlib

Dlib [14] is a modern general purpose cross-platform open source software library written in C++ that contains machine learning and computer vision algorithms for solving real world problems like facial landmark detection, deep metric learning, object tracking, and more. While the library is originally written in C++, it has good, easy to use Python bindings. Davis King has been the primary author of dlib since development began in 2002. In that time dlib has grown to include a wide variety of tools. All code in dlib is designed to be as portable as possible and similarly to not require a user to configure or install anything. To help achieve this, all platform specific code is confined inside the API wrappers. Everything else is either layered on top of those wrappers or is written in pure ISO standard C++. Currently the library is known to work on OS X, MS Windows, Linux, Solaris, the BSDs, and HP-UX. Dlib is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments.

OpenCV

OpenCV (Open Source Computer Vision Library) [6] is an open source computer vision and machine learning software library written natively in C++. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify

the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. The library is used extensively in companies, research groups and by governmental bodies.

PIL

PIL is the Python Imaging Library [27] by Fredrik Lundh and Contributors. The Python Imaging Library adds image processing capabilities to the Python interpreter. This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities. The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool. The Python Imaging Library is ideal for image archival and batch processing applications. You can use the library to create thumbnails, convert between file formats, print images, etc. The library contains basic image processing functionality, including point operations, filtering with a set of built-in convolution kernels, and colour space conversions. The library also supports image resizing, rotation and arbitrary affine transforms.

2.3.2 Algorithms

Face Landmark Detection

The Face Landmark Detection algorithm offered by Dlib is an implementation of the ensemble of regression trees (ERT). This technique uses simple and fast features like pixel intensities differences to directly estimate the landmark positions. These estimated positions are subsequently refined with an iterative process done by a cascade of regerssors. The regressors produces a new estimate from the previous one, trying to reduce the alignment error of the estimated points at each iteration. The algorithm is very fast, as it takes about 1-3ms on desktop platform to detect a set of 68 landmarks on a given input face image.

The author of the Dlib library (Davis King) has trained two shape predictor models on the iBug 300-W dataset, that respectively localize 68 and 5 landmark points within a face image. A shape predictor can be generated from a set of images, annotations and training options. A single annotation consists of the face region, and the labelled points that we want to localize. The face region can be easily obtained by any face detection algorithm like OpenCV HaarCascade, Dlib HOG Detector, CNN detectors, etc.), instead the points have to be manually labelled or detected by already-available landmark detectors and models like ERT with SP68. Lastly, the training options are a set of parameters that defines the characteristics of the trained model. The training process of a model is governed by a set of parameters. These parameters affect the size, accuracy and speed of the generated model. These parameters can be properly fine-tuned in order to get the desired behavior of the generated model. The most important parameters are:

- **tree_depth:** Specifies the depth of the trees used in each cascade. This parameter represents the capacity of the model. An optimal value is 4, instead a value of 3 is a good trade off between accuracy and model-size.

- **nu:** It is the regularization parameter. It determines the ability of the model to generalize and learn patterns instead of fixed-data. Value close to 1 will emphasize the learning of fixed-data instead of patterns, thus raising the chances for over-fitting to occur. Instead, an optimal nu value like 0.1 will make the model to recognize patterns instead of fixed-situations, totally eliminating the over-fitting problem. The amount of training samples can be a problem here, in fact with lower nu values the model needs a lot (thousands) of training samples in order to perform well.

- **cascade_depth:** It is the number of cascades used to train the model. This parameter affect either the size and accuracy of a model. A good value is about 10-12, instead a value of 15 is a perfect balance of maximum accuracy and a reasonable model-size.

- **feature_pool_size:** Denotes the number of pixels used to generate the features for the random trees at each cascade. Larger amount of pixels will lead the algorithm to be more robust and accurate but to execute slower. A value of 400 achieves a great accuracy with a good run-time speed. Instead, if speed is not a problem, setting the parameter value to 800 or even 1000 will lead to superior precision. Interestingly, with a value between 100 and 150 is still possible to obtain a quite good accuracy but with an impressing run-time speed. This last value is particularly suitable for mobile and embedded devices applications.

- **num_test_splits:** It is the number of split features sampled at each node. This parameter is responsible for selecting the best features at each cascade during the training process. The parameter affects the training speed and the model accuracy. The default value of the parameter is 20. This parameter can be very useful, for example, when we want to train a model with a good accuracy and keep its size small. This can be done by increasing the amount of num_split_test to 100 or even 300, in order to increase the model accuracy and not its size.

- **oversampling_amount:** Specifies the number of randomly selected deformations applied to the training samples. Applying random deformations to the training images is a simple technique that effectively increase the size of the training dataset. Increasing the value of the parameter to 20 or even 40 is only required in the case of small datasets, also it will increase the training time considerably. In the latest releases of the Dlib library, there is a new training parameter: the oversampling jittering amount that apply some translation deformation to the given bounding boxes in order to make the model more robust against eventually misplaced face regions.

By properly fine-tuning the training options, it is possible to customize the training process in such a way that satisfy the constraints of the system we are developing.

Moreover, by selecting only the relevant landmarks is possible to create specific models that localize a particular subset of landmarks, thus eliminating unnecessary points as demonstrated in Figure 2.5.

Figure 2.5: Examples of alternative shape predictors. The model in the first quarter, detects the eyes and eyebrows. The model on its right, localizes the entire set of landmarks with a faster execution speed and a reduced size compared to SP68. The last two models detect respectively: the face contour, nose and mouth.

2.4 Results and Discussion

Figure 2.1d shows the distribution of the three distances across the selected images from the three databases. These density plots show that the forehead, the distance between Trichion and Glabella, varies between about a fourth and a third of the face, with the mean around 30%. The nose, between Glabella and Subnasale, has a wider distribution, with length values between about 25% and 38% of the vertical length of the face, and a mean closer to one third. The mouth, between Subnasale and Menton, seems to be the longest of the three distances, with lengths between about 32% and 45% of the vertical length of the face, and a mean of about 38%.

One of the confounding factors of these variations is the resolution of the images in our analysis. For images with lower resolution the misplacement of the landmarks has a bigger influence of the distances, since 1 or 2 extra pixels could increase or decrease a distance by about 4%. For images with higher resolution such a misplacement would have a lesser impact on the normalized distances. To evaluate this impact we plotted the density distributions for each dataset separately, as shown in Figure 2.4. While there doesn't seem to be significant differences in the lengths of forehead and nose, across the three datasets, the length of the mouth, the distance between Subnasale and Menton, seems to have more variance for images with lower resolution, than for images with higher resolution.

One thing that is common across these density plots is the fact that these distances are not equal, with about 1/3 for forehead, 1/3 for nose, and 1/3 for mouth, as stated in the neoclassical canon, but rather that they have a range of values, with longer length for mouth than for nose and forehead. Thus, they suggest that this canon is not valid, and therefore it should be used with caution in cosmetic, plastic, or dental surgeries, and reconstruction procedures.

2.5 Related Work

[5] have performed the validation of vertical and horizontal neoclassical facial canons in Turkish young adults. They used a millimetric compass to take the measurements manually. The measurements were taken manually twice by the same investigators by filling out a form for recording the values. Based on their measurements, it was observed that only one male face had an equally divided facial profile. It was observed that the neoclassical canons were not valid in the majority of the population and the canons vary among races and also countries.

[1] have performed the validation of the vertical canon, the orbital canon and the orbito-nasal canon in young adults originating from the Arabian Peninsula. They measured the neoclassical canon using a caliper and analyzed the measurements using Student's t-test, general linear modeling, and pairwise comparison of means. The results indicated that all the three canons had variations in measurements. It was found out that the lower and upper thirds were longer than the middle thirds, the intercanthal distance was wider than eye fissure length and the nasal width was wider than the intercanthal distance.

[8] has performed a study of young adults in South-South Nigerian Ethnic Groups, Izon and Urhobo, to determine if there is a variation in length among the upper, middle and lower thirds of the face. The measurements of the thirds were taken in millimeters by using a sliding caliper. They performed data analysis with SPSS 23 by using descriptive and inferential statistics. In conclusion, it was found out that the three thirds of the face varied in lengths. The mean lengths of the upper and lower thirds were significantly longer in the Izon than the Urhobo, while the mean height of the middle third was significantly longer in the Urhobo than the Izon. The mean height of the male lower third was significantly longer in the Izon than the Urhobo,

while Urhobo females had significant longer middle third than the Izon.

[24] have developed a model to predict the attractiveness of the face based on neoclassical canons, symmetry and golden ratios. They used the feature point database that consists of the locations of the feature points for the faces from the FERET database and the faces of famous people. Neoclassical canons were one of the many predictors of attractiveness. One of these neoclassical canons used was the vertical canon where forehead height = nose length = lower face height. From the experiment results, it was found out that the vertical canon had a significant relationship with attractiveness. It was also found out that the attractiveness scores decreased significantly as the proportions of the face deviated from the proportions defined by the canons.

[20] have explored the presence of neoclassical canons of facial beauty among young people in Croatia and checked for any possible psychosocial repercussions occurring in those who demonstrate deviations in relation to the canons. Nine neoclassical canons of facial beauty were analyzed on a sample of 249 people with face and profile photographs taken in natural head position. Calculations were performed in the statistical software MedCalc 14.8.1 and based on previously published data. One of the 9 canons analyzed is the three portion facial profile canon where trichion - nasion (tr - n) = nasion - subnasale (n - sn) = subnasale - gnathion (sn - gn). All analyses were performed in the software AudaxCeph. Significant deviations from neoclassical facial beauty canons were found in 55-65% of adolescents and young adults and gender and age showed no relation to deviations. Most of the deviations from canons that affected the quality of life were the ones related to proportions of facial thirds.

[15] have performed the validation of six neoclassical canons among healthy young adult Chinese, Vietnamese and Thais by taking nine projective linear measurements. The nine projective linear measurements were taken by the authors by using standard

anthropometric methods. These nine measurements corresponded to six neoclassical facial canons. It was found out that in neither Asian nor Caucasian subjects were the three sections of the facial profile equal.

[7] have performed a comparative study among sixteen Miss Universe, sixteen Miss Universe Thailand, neoclassical canons and facial golden ratios to find out the most beautiful facial proportion in the 21st century by using twenty-six facial proportion points. Acrobat Reader was used to measure the distances and angles and the data was recorded in Microsoft Excel to compare the facial proportions. From the results, it was found out that the three-section proportion was longer in Miss Universe Thailand than in Miss Universe group.

[3] have proposed a solution for integrating aesthetics analytics into the functional workflow of dental technicians. They have presented a teeth pose estimation technique that can generate denture previews and visualizations that helps the dental technicians for designing the denture by considering the aesthetics and choosing the most aesthetically fitting preset from a library of dentures, in identifying the suitable denture size, and in adjusting the denture position. In one of the use cases that are demonstrated in this paper, it is stated that the dental technician uses the facial and dental proportions to identify the correspondence between the denture and the face which means that it is important to have the facial proportions correct for the denture to fit well on a patient.

Chapter 3

Finding Similar Non-Collapsed Faces to Collapsed Faces using Deep Learning Face Recognition

Since the vertical canon was invalidated, as shown in Section 2.4, it cannot be used for predicting the tip of the chin of a person with collapsed face. Therefore, we propose finding a similar face, and then using the landmarks from that face to reconstruct the lower third landmarks of the person with collapsed face.

3.1 Problem Description

Images contain a lot of crucial information that can be used for a variety of applications. The importance of search applications that closely match facial features in image-based searches is increasing day by day. We all have wondered if there was someone out there who looked just like us. We are all aware of the urban legend that there are six people out there who look just like us. The problem lies in finding those similar people. Face matching and retrieval can be used in forensics applications in matching forensic sketches to face photograph databases. Face matching also has its applications in recommender systems for glasses, hairstyle, jewellery, etc.

A lot of work has been performed in the field of facial similarity where different methods and metrics have been proposed for finding similar faces to a given input face like scale-invariant feature transform (SIFT) descriptors, lookalike networks, lo-

cal binary pattern (LBP), Doppelganger lists, etc. There are situations in real life where it can be required to find a similar face to a face which is collapsed or has structural deformities. However, none of the existing methods have focused on faces with deformities or a collapse. All the existing methods have the input and output similar images of normal faces without collapse or deformities where the facial similarity is calculated and a similar face is returned from a target dataset.

Missing human teeth cause the body to reabsorb the bone that supported the teeth. Over the course of about 10-20 years the jawbone shrinks significantly which results in a condition known as facial collapse. Humans with facial collapse appear much older than they are. The facial collapse not only alters the human's facial appearance, but also affects their dental health. This facial collapse can be prevented with the placement of dental implants. The implant sends a piezoelectric signal to the bone which prevents the bone from reabsorption. Dentures are designed by using the denture impressions of the jaw and mouth after which the dentist will create models usually from wax or plastic, based off the impression. The patient then tries the model several times to check for fit, shape, and even color before the denture is made. The current denture design workflow does not have a systematic approach to include the aesthetic factors, patient's pre-treatment facial shape and in-progress denture design visualizations, instead relying on discussing mockups with the patients during appointments. This results into waiting for the final denture fitting on the patient to evaluate the final denture aesthetics.

With a collapsed face, only the bottom one third of the face is affected and needs to be restored before denture designing. This facial restoration needs a similar face which can be used as a reference for comparing the facial shape proportions to convert the collapsed face into non-collapsed facial shape. Inclusion and evaluation of facial aesthetics is important while planning for facial or dental reconstruction treatment.

Our proposed method focuses towards the goal of finding a similar non-collapsed face to a given collapsed face for reconstructing the lower third of the collapsed face before designing the denture. We have used Python's deep learning face recognition [10] which is built using Dlib [14] for finding similar non-collapsed faces to a given collapsed face from a diverse target dataset of human population. By using the proposed method, one can find multiple similar images and use them for facial reconstruction of the lower third of the face for automatic denture designing.

3.2 Experimental Design

We have worked on finding similar non-collapsed faces for a given collapsed face by using the Human Faces dataset [12]. The Human Faces Dataset contains 224,500 images of human faces. The image data in the dataset has been generated using StyleGAN2 [13] which focuses on improving the resolution and quality of images. The StyleGAN2 was trained by using the Flickr Faces HQ (FFHQ) and Large-scale Scene UNderstanding Challenge (LSUN) datasets. The Human Faces Dataset contains nine directories and each such directory contains multiple folders that contain the image files. Each image file contains an image of a single human face. The dataset is diverse in a way that it contains images of human faces of different age groups beginning from toddlers to old humans and different genders, i.e., male and female, yet there's no metadata (e.g., age, race, etc.). All the image files in the dataset have a resolution of 1024 x 1024. Our proposed workflow as shown in Figure 3.1 had the following steps:

3.2.1 Classify Images into Different Gender and Age Categories

In order to find a similar non-collapsed face, it was important to find the similar face around the same age-group and gender. The first step was to classify the image

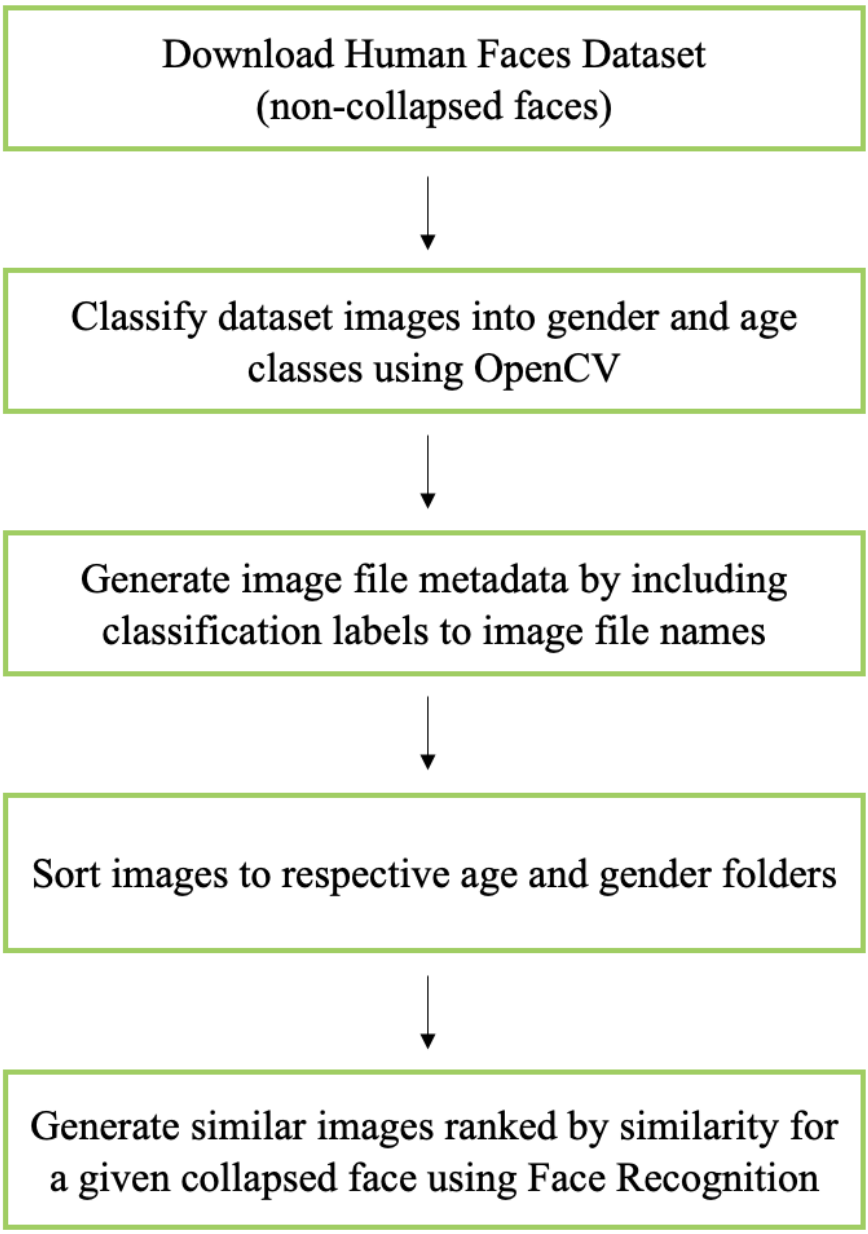


Figure 3.1: The Figure shows the flow of the proposed method where we first start with downloading the dataset to classify the images into different gender and age classes. Next, we generate the image file metadata and sort images into the different gender and age folders. The last step is to use Face Recognition to generate similar images for the given collapsed face.

according to gender into male and female categories. We used Python's OpenCV for age and gender detection of human faces in the image files. The images were classified into two genders, i.e., male and female, and further classified into different age groups under each gender, as shown in Figure 3.2. The different age classes included 0 to 2, 4 to 6, 8 to 12, 15 to 20, 25 to 32, 38 to 43, 48 to 53 and 60 to 100 years old.

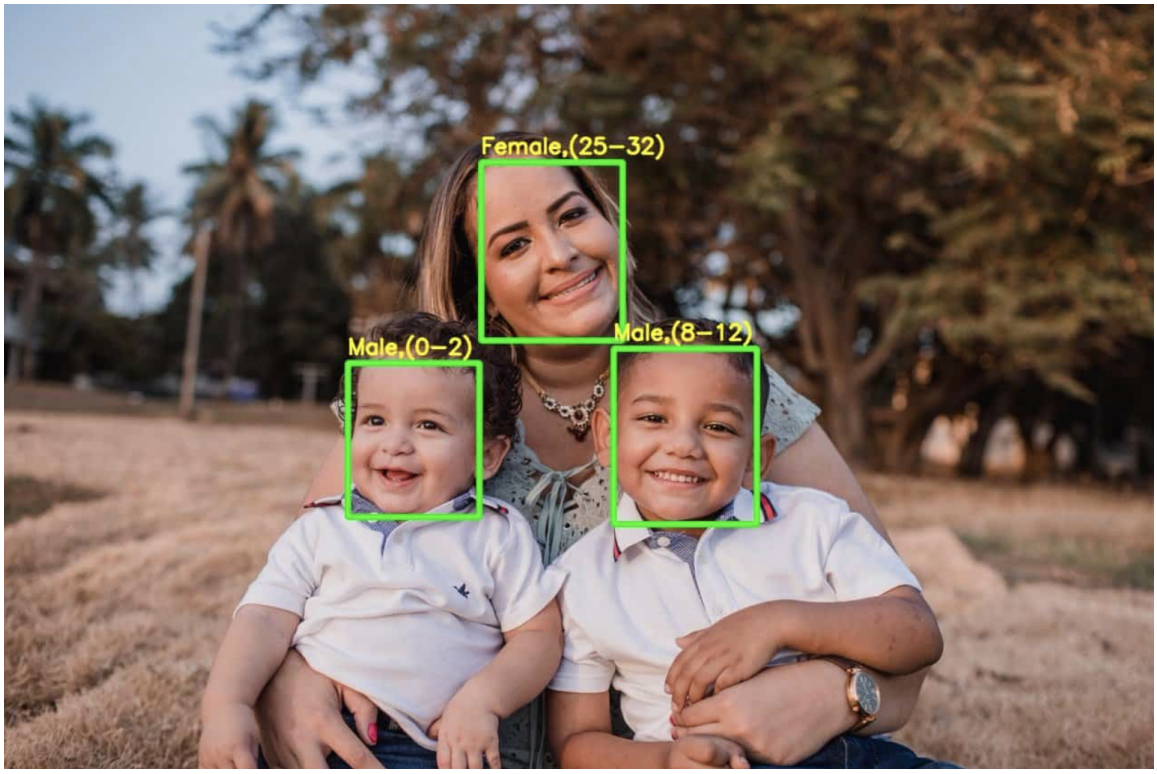


Figure 3.2: The figure shows how OpenCV's age and gender classification predicts the gender and age of three faces in the image. It correctly predicts the age and gender of a baby who is male and in the age group 0 to 2, of a female who is in the age group 25 to 32 and of a male kid who is in the age group 8 to 12.

The different steps involved in classification of gender and age include:

1. Read the image using the `cv2.imread()` method.

2. After the image is resized to the appropriate size, use the `get_faces()` function to get all the detected faces from the image.

3. Iterate on each detected face image and call our `get_age_predictions()` and `get_gender_predictions()` to get the predictions.

4. Print the age and gender along with the confidence levels.

3.2.2 Generate Image File Metadata

After detection of gender and age, the next step was to store the the output of classification as metadata for every image file. The file name of every image file was renamed to include the metadata of age and gender and their respective confidence levels.

3.2.3 Sort Image Files into Different Gender and Age Folders

After generating the image file metadata, the next step was to sort the classified data into different folders based on the gender and age for searching the similar non-collapsed face to a collapsed face in the folder that corresponds to the collapsed face's detected gender and age (e.g., `male_0to2`, `male_4to6`, etc.).

3.2.4 Generate Similar Images to given Query Image

After having all the images sorted in different folders as per their classification result, the next step was to look for a similar non-collapsed face to the given input collapsed face based on the gender and age of the input collapsed face by using Python's deep learning face recognition as shown in Figure 3.3. We found out that the process of generating similar images is slow. To speed up the process we created small samples of each folder mentioned in Step 3 that contained 500 images and generated similar images by searching only the sample folder.

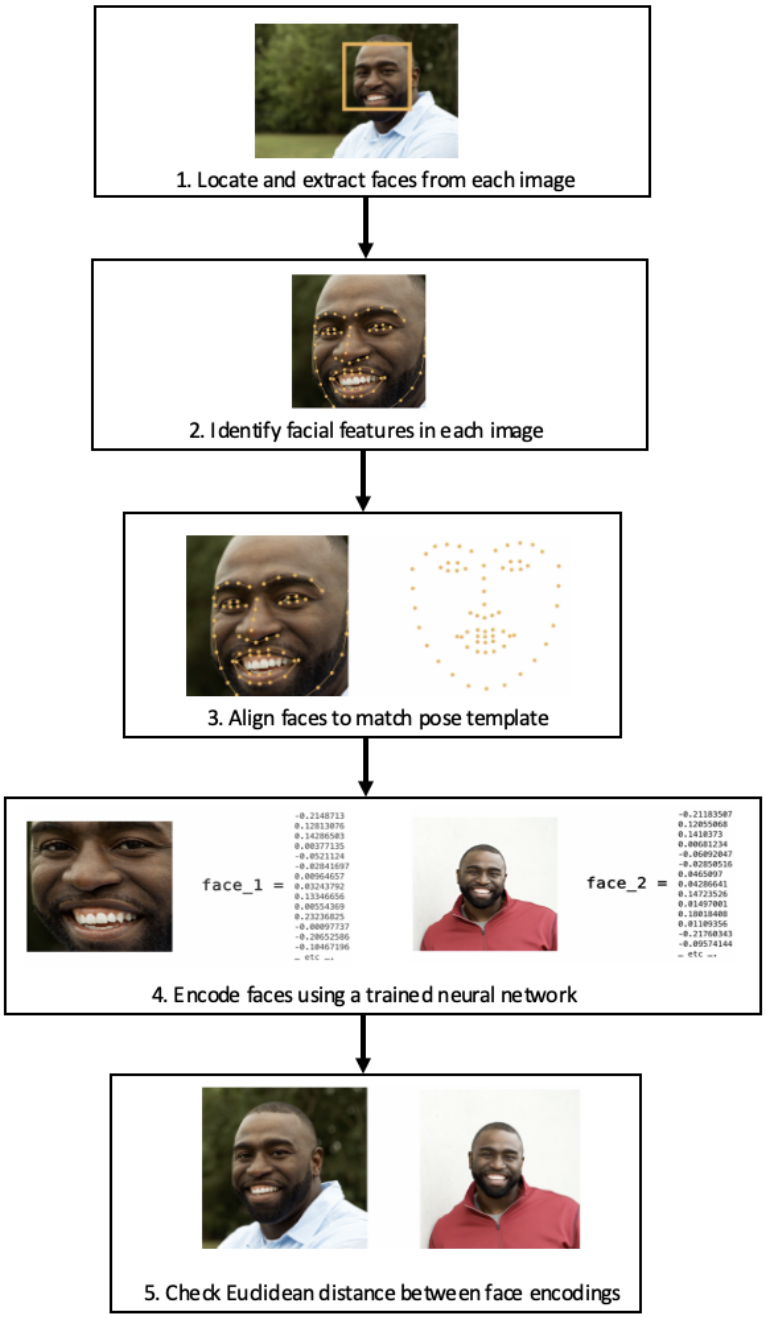


Figure 3.3: The figure explains the face recognition pipeline that starts with first locating and extracting the faces from each image. The facial features from each image are identified. Next, the faces in the image are aligned to match with the Dlib's 68-landmark pose template. Different facial measurements are calculated using deep metric learning followed by checking the Euclidean distance between the different face encodings for finding the most similar face match.

3.3 Materials and Machine Learning Methods Used

3.3.1 Libraries

OpenCV

OpenCV (Open Source Computer Vision Library) [6] is an open source computer vision and machine learning software library written natively in C++. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. The library is used extensively in companies, research groups and by governmental bodies.

Face Recognition

Face Recognition is the most popular free face recognition library, as it has 45k stars on GitHub. There are two options for using it; one can either use their Python API or their binary command line tool. There are installation instructions for all main platforms and even a docker image for fast set-up. The library uses many built-in

libraries such as Dlib and it uses machine learning to recognize the faces with an accuracy of 99.38% and it performs better than the human brain can do. The face recognition library has many methods to deal with faces in images and one of them known as face_locations will find the face's locations inside a particular image. This method will find those faces and return an array of coordinates of each face and we can print them out. With face recognition we can count the number of people within an image, recognize people's faces and identify who they are, extract faces within an image, find similar looking people, applying digital makeup to a face, etc,.

PIL

PIL is the Python Imaging Library [27] by Fredrik Lundh and Contributors. The Python Imaging Library adds image processing capabilities to the Python interpreter. This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities. The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool. The Python Imaging Library is ideal for image archival and batch processing applications. You can use the library to create thumbnails, convert between file formats, print images, etc. The library contains basic image processing functionality, including point operations, filtering with a set of built-in convolution kernels, and colour space conversions. The library also supports image resizing, rotation and arbitrary affine transforms.

3.3.2 Algorithms

Age and Gender Classification

OpenCV's gender and age classification is based on a convolutional neural network architecture with a total of 3 convolutional layers, 2 fully connected layers and a final output layer. "Conv1" is the first convolutional layer that has 96 nodes of kernel size 7. "Conv2" is the second convolutional layer that has 256 nodes of kernel size 5. "Conv3" is the third convolutional layer that has 384 nodes of kernel size 3. The two fully connected layers have 512 nodes each. Gender prediction is framed as a classification problem. The output layer in the gender prediction network is of type softmax with 2 nodes indicating the two classes "Male" and "Female". Age Prediction should be approached as a regression problem since we are expecting a real number as the output. However, it is difficult to accurately estimate the age of a person and even humans find it challenging. Hence, age prediction was framed as a classification problem where we try to estimate the age group the person is in. The age prediction network has 8 nodes in the final softmax layer indicating the age ranges 0 to 2, 4 to 6, 8 to 12, 15 to 20, 25 to 32, 38 to 43, 48 to 53 and 60 to 100 years old. The different model files used and loaded for performing the age and gender detection task are listed below:

- `gender_net.caffemodel`: It is the pre-trained model weights for gender detection.
- `deploy_gender.prototxt`: It is the model architecture for the gender detection model.
- `age_net.caffemodel`: It is the pre-trained model weights for age detection.
- `deploy_age.prototxt`: It is the model architecture for the age detection model.
- `res10_300x300_ssd_iter_140000_fp16. caffemodel`: It is the pre-trained model weights for face detection.

- `deploy.prototxt.txt`: It is the model architecture for the face detection model.

OpenCV based gender detection network performs well as compared to the age prediction network. It is very difficult to predict the age of people just from images. Figure 3.4 is from the paper [16] and it describes the confusion matrix for multi-class age classification results. The following observations can be made from the Figure 3.4.

- The age groups 0-2, 4-6, 8-13 and 25-32 are predicted with relatively high accuracy as we can see it clearly from the diagonal elements of the confusion matrix.
- The output is heavily biased towards the age group 25-32. So, even if the actual age is between 15-20 or 38-43, there is a high chance that the predicted age will be 25-32.

Some other observations about the age and gender detection network state that the accuracy of the models improved if padding was used around the detected face and that the predictions improved for some examples when the faces were aligned before detection. Also, it can be a good idea to try to use a regression model instead of classification for age prediction if enough data is available.

	0-2	4-6	8-13	15-20	25-32	38-43	48-53	60-
0-2	**0.699**	0.147	0.028	0.006	0.005	0.008	0.007	0.009
4-6	0.256	**0.573**	0.166	0.023	0.010	0.011	0.010	0.005
8-13	0.027	0.223	**0.552**	0.150	0.091	0.068	0.055	0.061
15-20	0.003	0.019	0.081	**0.239**	0.106	0.055	0.049	0.028
25-32	0.006	0.029	0.138	0.510	**0.613**	0.461	0.260	0.108
38-43	0.004	0.007	0.023	0.058	0.149	**0.293**	0.339	0.268
48-53	0.002	0.001	0.004	0.007	0.017	0.055	**0.146**	0.165
60-	0.001	0.001	0.008	0.007	0.009	0.050	0.134	**0.357**

Figure 3.4: The figure shows the confusion matrix from the paper [16] for the proposed age prediction model's multi-class age classification results.

3.4 Results and Discussion

Figure 3.5 shows the query image that contains a collapsed face of a male in the age group 60 to 100 and the top ten similar images to the query image ranked by their similarity scores. Figure 3.6 shows the query image of a collapsed face of a female in the age group 48 to 53 and the top ten similar images to the query image ranked by their similarity scores. The image with the lowest similarity score is the most similar image to the query image. In Figure 3.5, the most similar face has a similarity score of 0.6717 followed by the other less similar faces with an increasing similarity scores. In Figure 3.6, the most similar face has a similarity score of 0.6210 followed by other less similar faces with an increasing similarity scores.

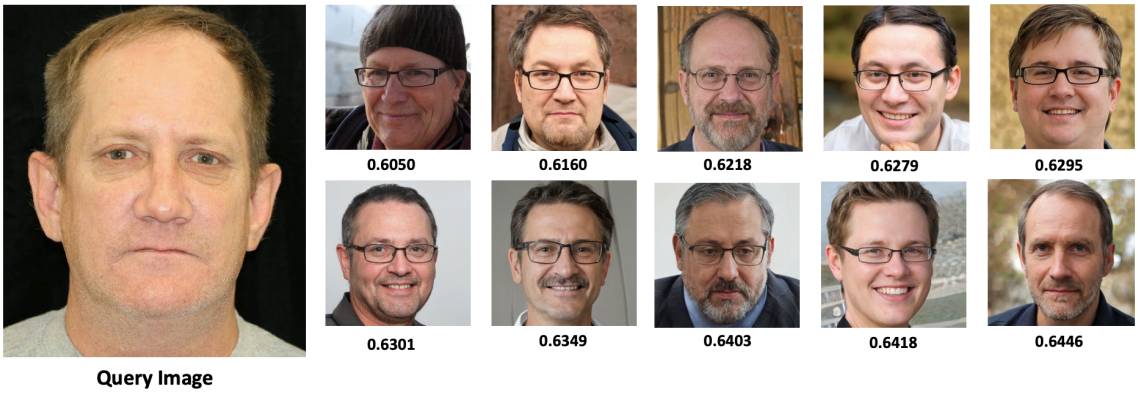


Figure 3.5: The query image is a collapsed face of a male in the age group 60 to 100. Top ten non-collapsed faces ranked by their similarity are found that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images has the similarity of 0.6518 and 0.6528.

We have performed experiments of finding similar faces on samples of every category as well as on entire classified dataset. In Figure 3.5 and Figure 3.6 the gender of all the similar images in the output matches the gender of the query image that contains a collapsed face. Also, the age of most of the similar images matches with the age of the query image in Figure 3.5 and Figure 3.6. Figures 3.7, 3.8, 3.9, 3.10, 3.11,

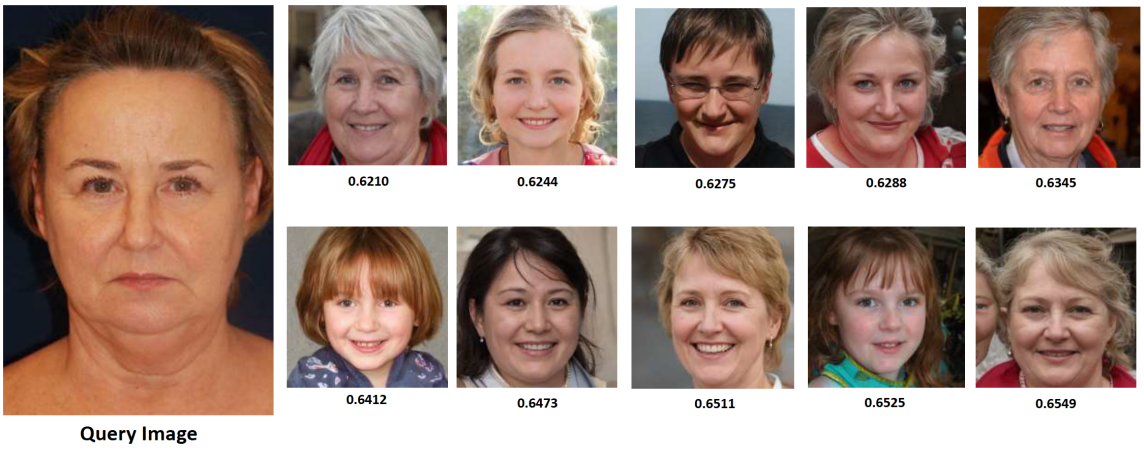


Figure 3.6: The query image is a collapsed face of a female in the age group 48 to 53. Top ten non-collapsed faces ranked by their similarity are found that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.65763 and 0.65766.

3.12, 3.13, 3.14, 3.15, 3.16, 3.17, 3.18, 3.19 and 3.20 show the query image and the resulting ten similar non-collapsed images from a sample of the dataset as the output based on the ranked similarity scores. Figures 3.23, 3.22 and 3.21, show the query image and the resulting ten similar non-collapsed images from the entire classified dataset as the output based on the ranked similarity scores. It is observed that more similar images were found when the entire dataset was considered vs. only a random sample of the dataset.

During the step of generating image file metadata i.e., gender and age labels, Python's OpenCV based age and gender classification also displays a confidence score for each label. This confidence score can be used for filtering out incorrectly labelled images from the age and gender category folders so that each folder contains only correctly labelled images. Having ten similar images gives the flexibility to choose from the set that might have some images that do not satisfy some criteria. Some of the query images in figures do not have a collapsed face but were included to show that the proposed method works for those age groups.

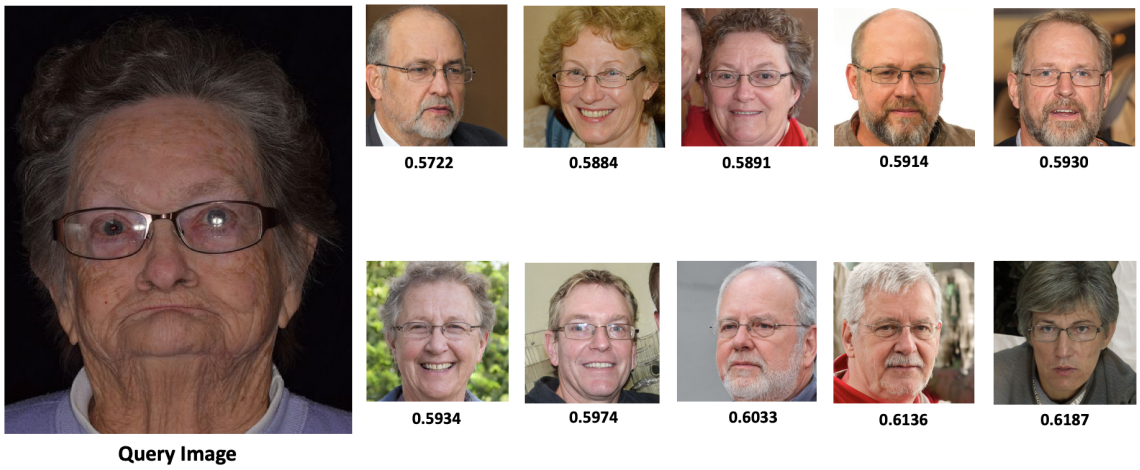


Figure 3.7: The query image is a collapsed face of a female in the age group 60 to 100. Top ten non-collapsed faces ranked by their similarity are found that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.6223 and 0.6243.

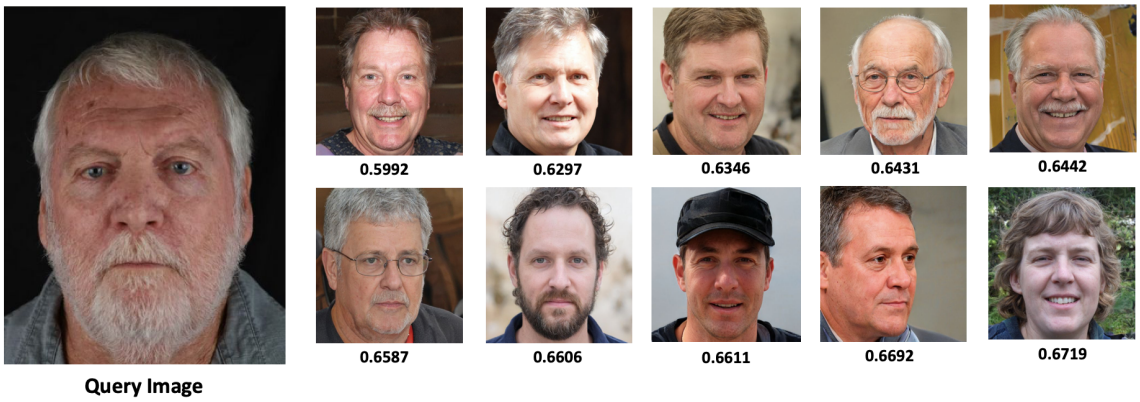


Figure 3.8: The query image is a collapsed face of a male in the age group 48 to 53. Top ten non-collapsed faces ranked by their similarity are found that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.6720 and 0.6733.

The dataset had some duplicate images which were returned in the similarity results. In this situation, only one such image was selected and the next similar image was considered. Some images in the dataset did not have clearly visible faces due to which the face recognition algorithm was not able to locate and extract faces in the first step. To resolve this error, we excluded such images after which we

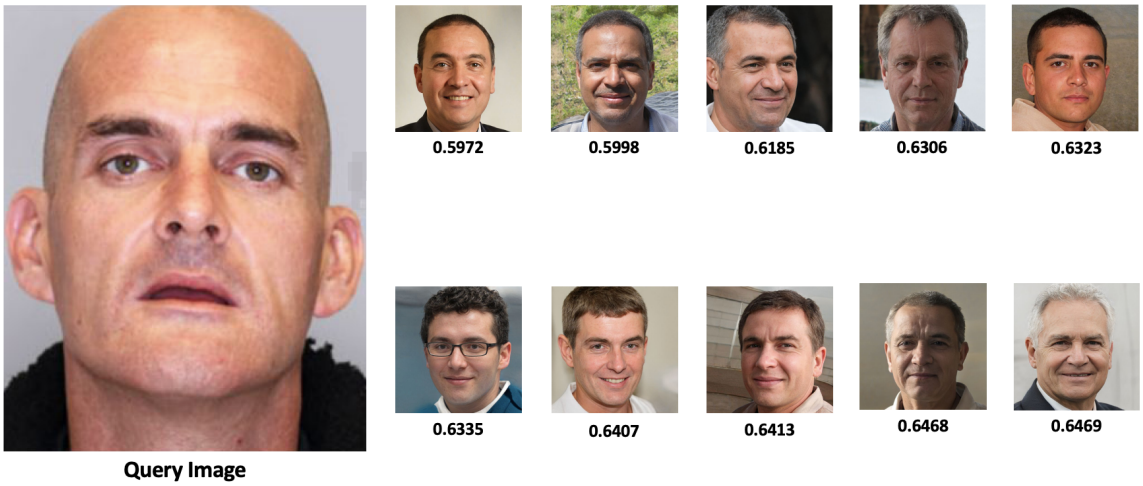


Figure 3.9: The query image is a collapsed face of a male in the age group 38 to 43. Top ten non-collapsed faces ranked by their similarity are found that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.6491 and 0.6525.

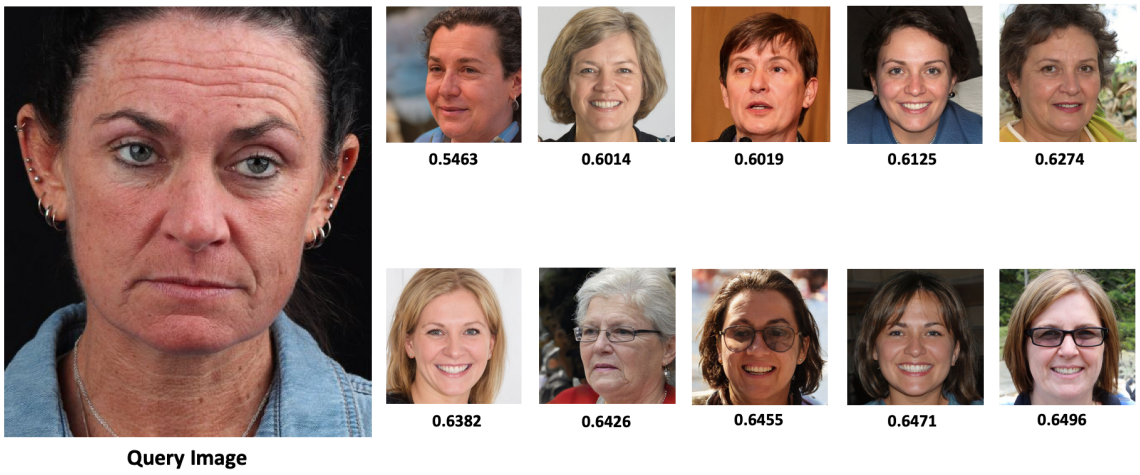


Figure 3.10: The query image is a collapsed face of a female in the age group 38 to 43. Top ten non-collapsed faces ranked by their similarity are found that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.6516 and 0.6560.

were able to find similar non-collapsed images to a given collapsed face. Since the dataset used in this paper did not have labelled data, we used machine learning to classify the images into different age and gender categories. During this process of

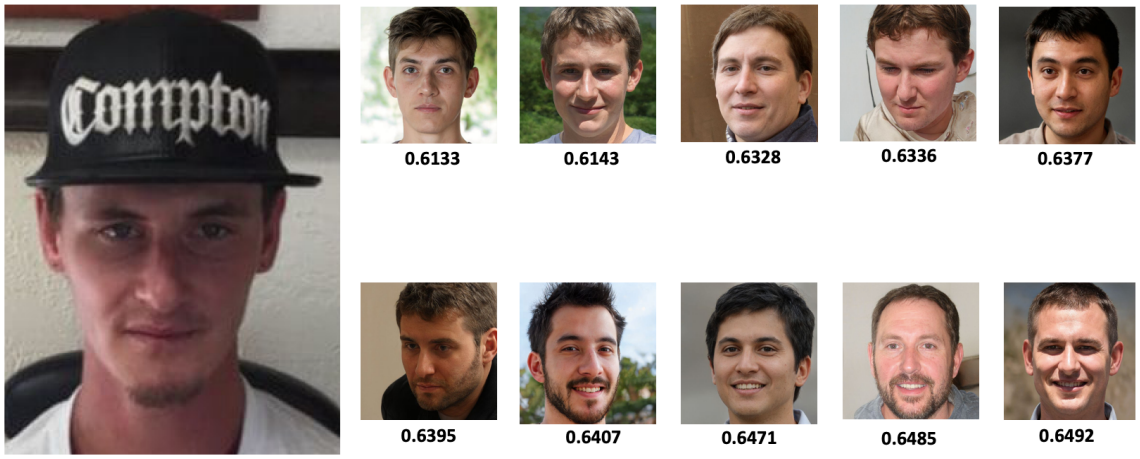


Figure 3.11: The query image is a collapsed face of a male in the age group 25 to 32. Top ten non-collapsed faces ranked by their similarity are found that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.6519 and 0.6530.

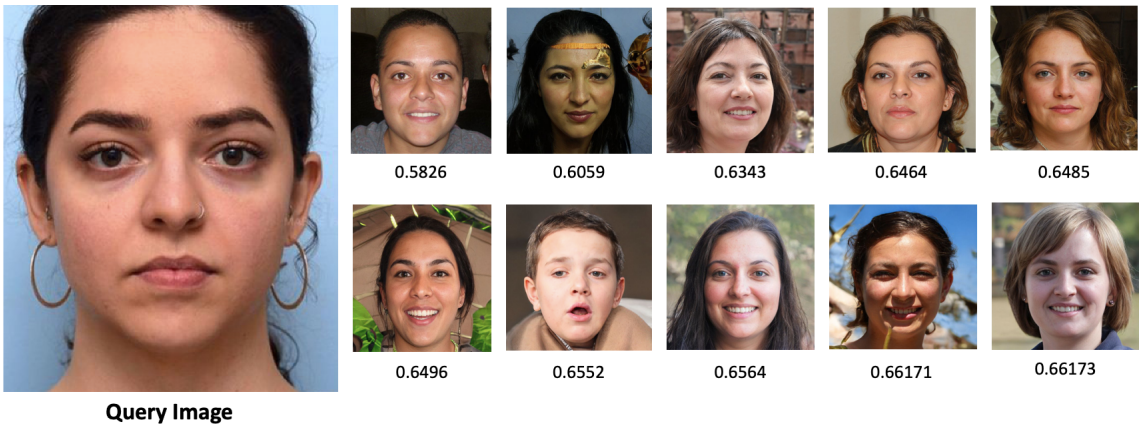


Figure 3.12: The query image is a collapsed face of a female in the age group 25 to 32. Top ten non-collapsed faces ranked by their similarity are found that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.6635 and 0.6670.

classification, there were images that were not classified into correct folders. Thus some folders contained images that belong to other age and gender category. This resulted into missing out on some images that were potential candidates for similar images. During the step of generating image file metadata i.e., gender and age labels,

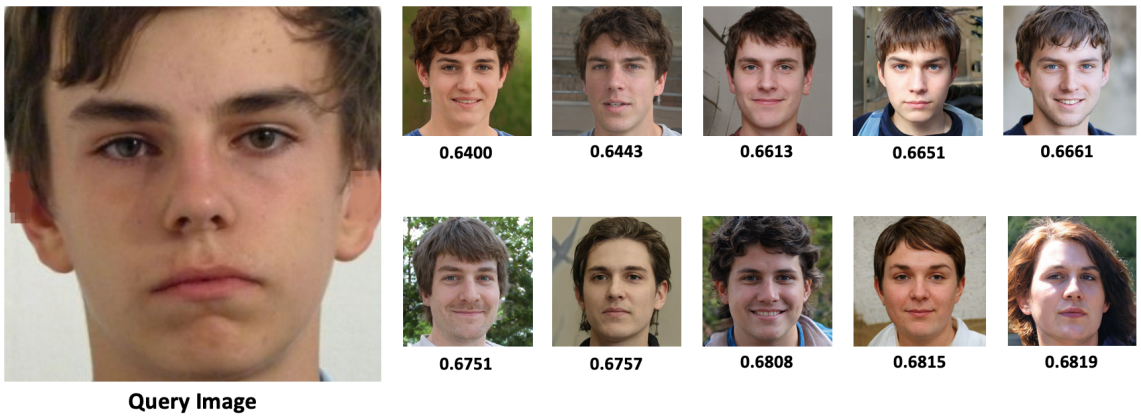


Figure 3.13: The query image is a collapsed face of a male in the age group 15 to 20. Top ten non-collapsed faces ranked by their similarity are found that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.6841 and 0.6862.

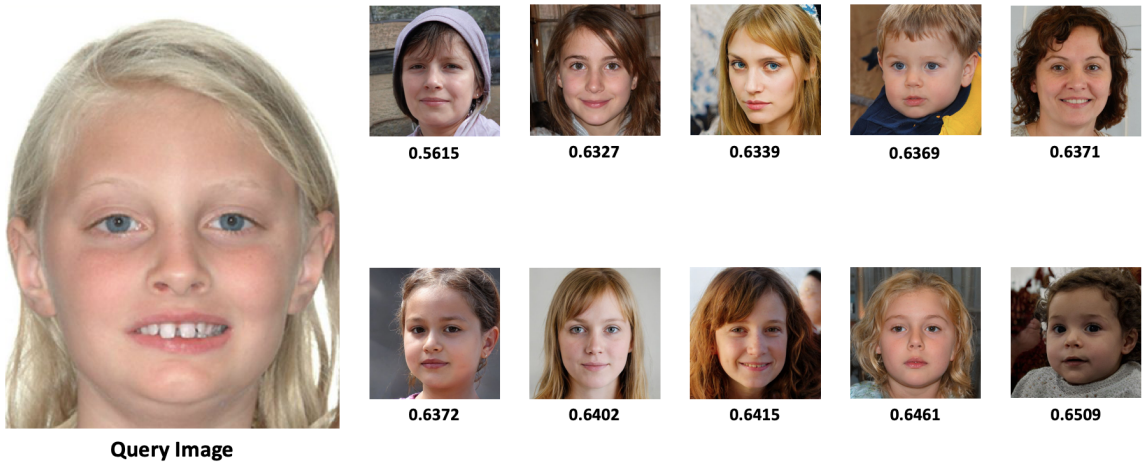


Figure 3.14: The query image is a collapsed face of a female in the age group 15 to 20. Top ten non-collapsed faces ranked by their similarity are found that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.6545 and 0.6548.

Python's OpenCV based age and gender detection also displays a confidence score for each label. This confidence score can be used for filtering out incorrectly labelled images from the age and gender category folders so that each folder contains only correctly labelled images.

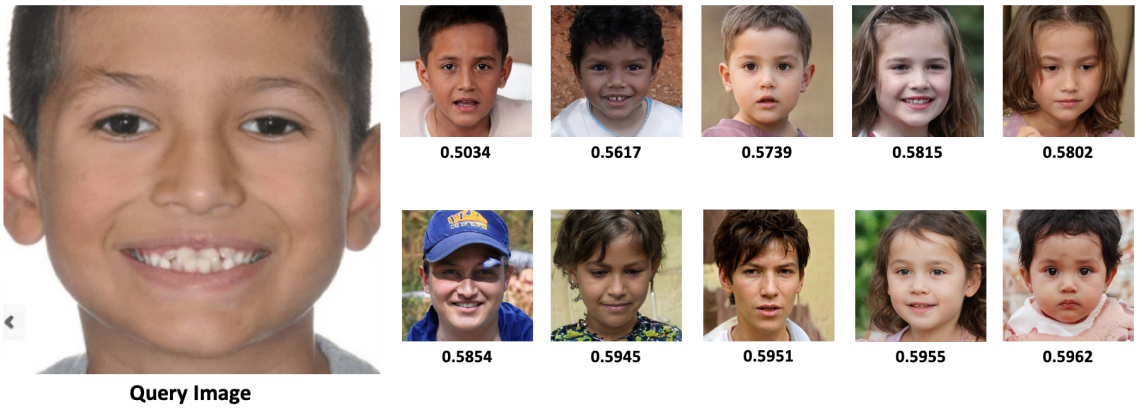


Figure 3.15: The query image is a collapsed face of a male in the age group 8 to 12. Top ten non-collapsed faces ranked by their similarity are found from a random sample of images that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.5967 and 0.5978.

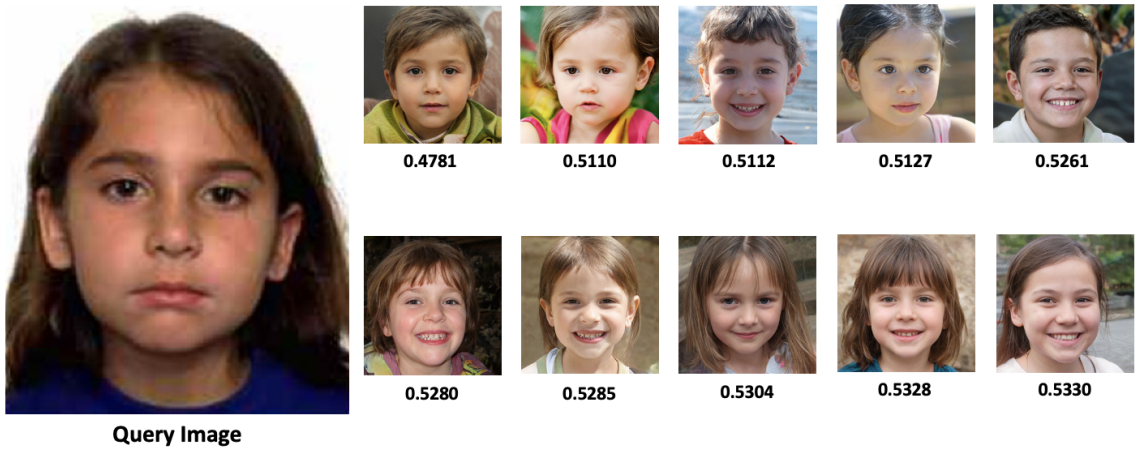


Figure 3.16: The query image is a collapsed face of a female in the age group 8 to 12. Top ten non-collapsed faces ranked by their similarity are found from a random sample of images that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.5334 and 0.5341.

3.5 Related Work

In [26], a method is developed to match similar faces in scattered datasets and to recognize images of a person in different states. A solution that works more effectively than traditional face recognition methods, which works with a high error rate

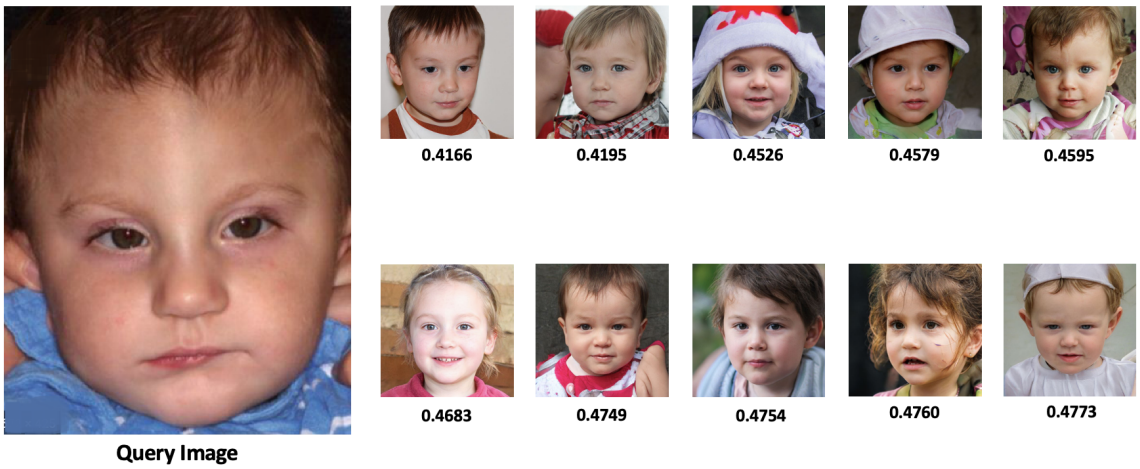


Figure 3.17: The query image is a collapsed face of a male in the age group 4 to 6. Top ten non-collapsed faces ranked by their similarity are found from a random sample of images that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.4775 and 0.4796.

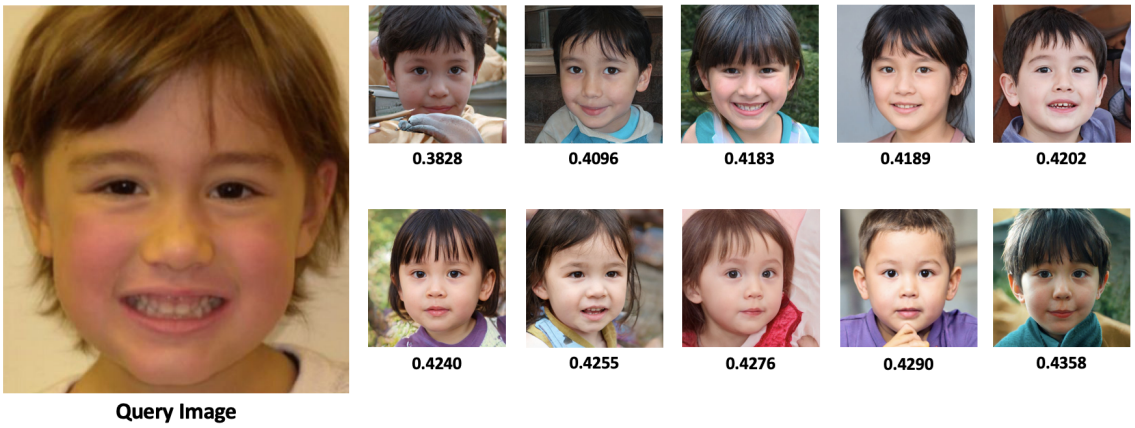


Figure 3.18: The query image is a collapsed face of a female in the age group 4 to 6. Top ten non-collapsed faces ranked by their similarity are found from a random sample of images that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.4364 and 0.4376.

according to different exposure values, is presented. By using SIFT descriptors that are used in object recognition problems, first, the face's points of interest are identified followed by matching points of interests between two faces. Then, the similarity ratios of two faces to each other are calculated by looking at the distances. The av-

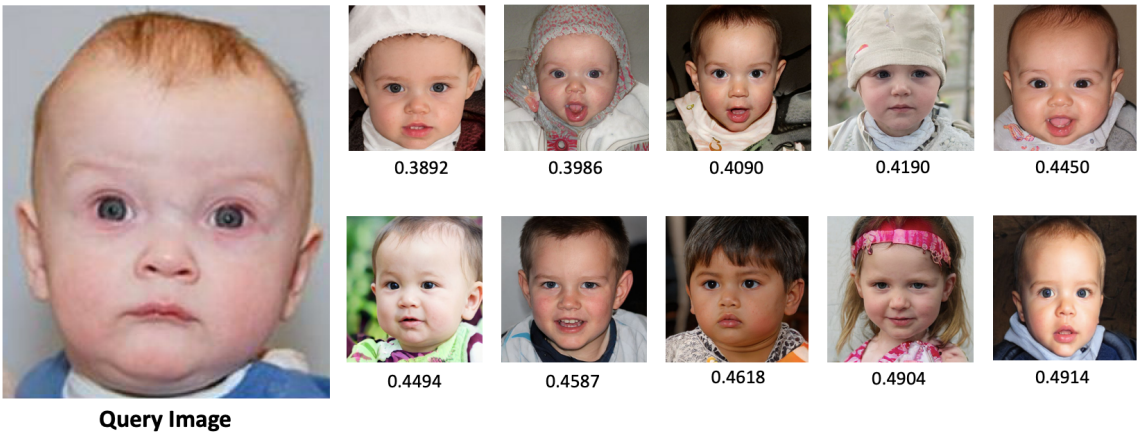


Figure 3.19: The query image is a collapsed face of a male in the age group 0 to 2. Top ten non-collapsed faces ranked by their similarity are found from a random sample of images that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.4938 and 0.4941.

Figure 3.20: The query image is a collapsed face of a female in the age group 0 to 2. Top ten non-collapsed faces ranked by their similarity are found from a random sample of images that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.5010 and 0.5027.

erage distances of the points of interest assumed to be correctly mapped determined this ratio.

[25] have introduced a new method for comparing pairs of face images that allows recognition or verification between images where the mutually visible portion of face is small. They have explored the idea of using a sorted Doppelganger list as a signature

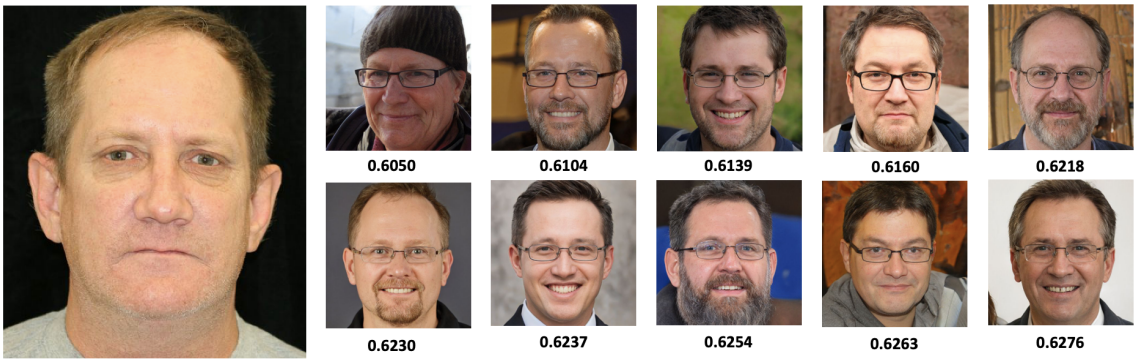


Figure 3.21: The query image is a collapsed face of a male in the age group 60 to 100. Top ten non-collapsed faces ranked by their similarity are found from the entire dataset that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.6279 and 0.6295.

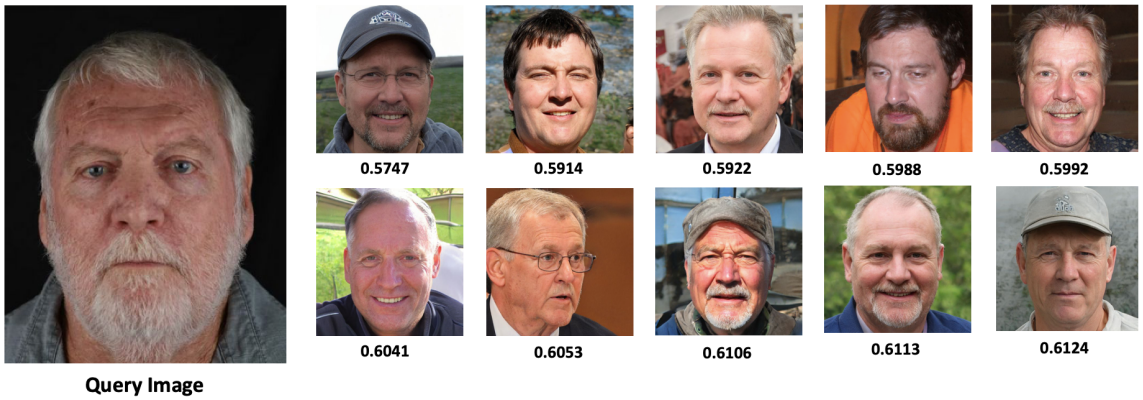


Figure 3.22: The query image is a collapsed face of a male in the age group 48 to 53. Top ten non-collapsed faces ranked by their similarity are found from the entire dataset that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.6131 and 0.6133.

and evaluate distance functions for comparing the signatures. Each probe in a query pair is compared to all members of the Library. The comparison results in a ranked list of look-alikes, the first being the most similar to the query. Then, a similarity measure between the two probes is computed by comparing the ranked lists.

In [23], the emphasis is on presenting evidence that finding facial look-alikes and recognizing faces are two distinct tasks. They expected that many features which

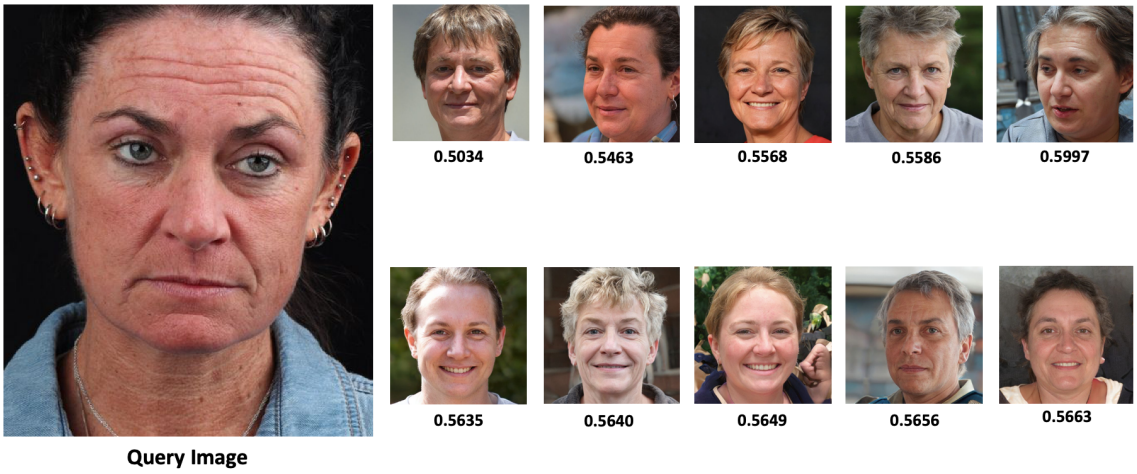


Figure 3.23: The query image is a collapsed face of a female in the age group 38 to 43. Top ten non-collapsed faces ranked by their similarity are found from the entire dataset that can be used for reconstructing the bottom third of the collapsed face. The next two ranked images have the similarity of 0.5700 and 0.5722.

are useful for face recognition will also be useful for face similarity. They have used the pre-trained VGG face CNN descriptor network for face-recognition and then fine-tuned the weights on a new dataset that is targeted at capturing preceived facial similarity to perform well at the face similarity task by proposing the lookalike network. The results show that the proposed method outperforms the face recognition baseline at the task of predicting which faces will appear more similar to human.

In [21], the focus is on deriving a measure of similarity between faces. They mention that illumination, pose variations, disguises, aging effects and expression variations are some of the key factors that affect the performance of face recognition systems. They have suggested a framework to compensate for pose variations and introduce the notion of 'half-faces' to circumvent the problem of non-uniform illumination and used the similarity measure to retrieve similar faces from a database containing multiple images of individuals. They concluded that the similarity measure helps in studying the significance facial features play in affecting the performance

of face recognition systems.

In [17], they have used the twin database to calculate a baseline measure of the worst-case scenario of facial similarity in face recognition using a deep CNN. They have carried out a performance analysis of two Face Recognition tools presented with highly similar faces and using an experimental twin threshold, potential look-alikes were extracted from the datasets for further analysis. The similarity measure presented in this paper can be used to compare facial similarity to a comparison score from a face recognition system in order to better understand the impact that facial similarity has on Face Recognition and to identify potential look-alikes from large datasets.

In [2], they have showed that adopting an adversarial deep learning-based approach allows for the model to maintain the accuracy at face matching while also reducing demographic disparities compared to a baseline (non-adversarial deep learning) approach at face matching and paves way for more accurate and fair face matching algorithms. They have proposed a deep-learning adversarial approach for reducing bias in face-matching algorithms. The proposed GAN-based framework consisted of two parts: one which tries to maximize the face matching quality and the other which tries to minimize the ability of the network to infer the demographic properties of the individual whose facial image is under consideration.

In [4], they have presented a novel approach for extracting characteristic parts of a face. Instead of finding a priori specified features such as nose, eyes, mouth or others, the proposed approach is aimed at extracting from a face the most distinguishing or dissimilar parts with respect to another given face, i.e. at "finding differences" between faces by feeding a binary classifier by a set of image patches, randomly sampled from the two face images, and scoring the features by their mutual distances.

In [22], they have investigated the role of differences between people in the com-

parison process. They have proposed a dual process model in which dissimilarity can function in two different ways with opposite effects on social judgement. They have discussed two different arguments: 1. Dissimilarity between people may decrease the likelihood of placing them in the same category. 2. Activated dissimilarity may determine the detection of feature overlap between people during comparison and therefore influence the holistic similarity assessment.

In [19], they have compared seven state-of-the-art face recognition algorithms with humans on a facematching task where humans and algorithms determined whether pairs of face images, taken under different illumination conditions, were pictures of the same person or of different people. It was found out that three algorithms surpassed human performance matching "difficult" face pairs and six algorithms surpassed humans on "easy" face pairs. They mention that although illumination variation continues to challenge face recognition algorithms, current algorithms compete favorably with humans.

Chapter 4

Conclusion and Future Work

4.1 Conclusion

The neoclassical canons are used to define the different proportions between various areas of the head and the face. These facial canons have been recommended in various textbooks about orthodontics, prosthodontics, plastic and dental reconstructive surgeries for planning the treatment procedure. We tested the hypothesis of the face being vertically divided equally into thirds using machine learning. Our results indicate that the vertical dimensions of the face are not always divided equally into thirds. Thus, this vertical canon should be used with caution in cosmetic, plastic or dental surgeries or any reconstruction procedures.

The bottom third of the face is adversely affected when a facial collapse due to missing teeth and jaw bone loss occurs. It is important to properly restore this third of the face before designing dentures. The lower third of a collapsed face can be restored by using a similar non-collapsed face as a reference face. We have proposed a method to find a similar non-collapsed face to a given collapsed face using Python's deep learning face recognition. Our results provide the similar non-collapsed images to the given collapsed image which can be used for reconstructing the bottom third of a collapsed face before designing dentures. The more diverse and larger the target dataset, the more chances of finding a closer and similar image to the given collapsed

image.

4.2 Future Work

The analysis on vertical canon validation would require further evaluation, as many groups were not well represented in these datasets. For example, there are very few children, very few people over the age of 80, and a relatively small proportion of women. In addition, many ethnicities have very minor representation or none at all. In addition to creating a new dataset that has a wider representation, we also recommend collecting metadata about the images, which should include the details about each individual, such as age, race, etc., as well as whether they have all teeth or if they have dentures (which is difficult to determine from these images). Furthermore, the images should include a side profile view for each person, in addition to the frontal view. Another alternative to a new image database, that is worth exploring, is to collect and annotate three-dimensional scans. These have the potential to enable better localization of the four lines, as with two-dimensional front views it is difficult to determine the position of the Trichion and Glabella.

When looking for a similar non-collapsed face which can be used as a reference to the query image that contains a collapsed face, it would be ideal to have a reference face that is a closest match to the query image as this would help in more accurate reconstruction procedure of the query image that contains a collapsed face. Since this proposed method is about finding similar images from the target dataset, the bigger and diverse the dataset is, the more is the probability of finding a closer and a similar match to the query image. However, it should be made sure that the target dataset does not contain images of people with a collapsed face. In addition to age and gender, there can be other factors like race, ethnicity, skin color, height, weight,

etc. that can affect the similarity score and help to find the closest match. It is worth exploring the similar images in a more wide and diverse dataset to check for the similarity scores and similar images. We would like to explore more on this if we could actually use these outputs for facial reconstruction. Some of the similar images that were returned by face recognition had the images of the opposite gender i.e., a similar image of a female was returned for an image containing a male collapsed face. We would like to explore more on this if we could actually use these outputs for facial reconstruction. Also, when classifying images into different gender and age categories, OpenCV gives a confidence score for each performed detection. We would like to consider this score to only select images that are correctly classified into their age and gender category.

BIBLIOGRAPHY

[1] Al-Sebaei, M. O. The validity of three neo-classical facial canons in young adults originating from the arabian peninsula. *Head & face medicine 11*, 1 (2015), 1–7.

[2] Alasadi, J., Al Hilli, A., and Singh, V. K. Toward fairness in face matching algorithms. In *Proceedings of the 1st International Workshop on Fairness, Accountability, and Transparency in MultiMedia* (New York, NY, USA, 2019), FAT/MM '19, Association for Computing Machinery, p. 19–25.

[3] Amirkhanov, A., Bernhard, M., Karimov, A., Stiller, S., Geier, A., Gröller, M. E., and Mistelbauer, G. Visual analytics in dental aesthetics. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 635–646.

[4] Bicego, M., Grosso, E., and Tistarelli, M. On finding differences between faces. In *International Conference on Audio-and Video-Based Biometric Person Authentication* (2005), Springer, pp. 329–338.

[5] Bozkir, M., Karakas, P., and Oguz, Ö. Vertical and horizontal neoclassical facial canons in turkish young adults. *Surgical and Radiologic Anatomy 26*, 3 (2004), 212–219.

[6] Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[7] Burusapat, C., and Lekdaeng, P. What is the most beautiful facial proportion in the 21st century? comparative study among miss universe, miss universe thailand, neoclassical canons, and facial golden ratios. *Plastic and Reconstructive Surgery Global Open 7*, 2 (2019).

[8] Eboh, D. E. O. Horizontal facial thirds of young adults in two south-south nigerian ethnic groups. *Anatomy & Biological Anthropology 32*, 4 (2019), 115–119.

[9] Farkas, L. G., Hreczko, T. A., Kolar, J. C., and Munro, I. R. Vertical and horizontal proportions of the face in young adult north american caucasians:

revision of neoclassical canons. *Plastic and Reconstructive Surgery 75*, 3 (1985), 328–338.

[10] Geitgey, A. Face recognition library, 2018.

[11] Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[12] kaggle. Human faces dataset, 2020.

[13] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 8110–8119.

[14] King, D. E. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research 10* (2009), 1755–1758.

[15] Le, T. T., Farkas, L. G., Ngim, R. C., Levin, L. S., and Forrest, C. R. Proportionality in asian and north american caucasian faces using neoclassical facial canons as criteria. *Aesthetic plastic surgery 26*, 1 (2002), 64–69.

[16] Levi, G., and Hassner, T. Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (2015), pp. 34–42.

[17] McCauley, J., Soleymani, S., Williams, B., Dando, J., Nasrabadi, N., and Dawson, J. Identical twins as a facial similarity benchmark for human facial recognition. In *2021 International Conference of the Biometrics Special Interest Group (BIOSIG)* (2021), pp. 1–5.

[18] Milborrow, S., Morkel, J., and Nicolls, F. The MUCT Landmarked Face Database. *Pattern Recognition Association of South Africa* (2010). `http://www.milbo.org/muct`.

[19] O'Toole, A. J., Jonathon Phillips, P., Jiang, F., Ayyad, J., Penard, N., and Abdi, H. Face recognition algorithms surpass humans matching faces over changes in illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 9 (2007), 1642–1646.

[20] Pavlic, A., Zrinski, M. T., Katic, V., and Spalj, S. Neoclassical canons of facial beauty: do we see the deviations? *Journal of Cranio-Maxillofacial Surgery 45*, 5 (2017), 741–747.

[21] Ramanathan, N., Chellappa, R., and Roy Chowdhury, A. Facial similarity across age, disguise, illumination and pose. In *2004 International Conference on Image Processing, 2004. ICIP '04.* (2004), vol. 3, pp. 1999–2002 Vol. 3.

[22] Ruys, K. I., Spears, R., Gordijn, E. H., and de Vries, N. K. Two faces of (dis) similarity in affective judgments of persons: Contrast or assimilation effects revealed by morphs. *Journal of Personality and Social Psychology 90*, 3 (2006), 399.

[23] Sadovnik, A., Gharbi, W., Vu, T., and Gallagher, A. Finding your lookalike: Measuring face similarity rather than face identity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (June 2018).

[24] Schmid, K., Marx, D., and Samal, A. Computation of a face attractiveness index based on neoclassical canons, symmetry, and golden ratios. *Pattern Recognition 41*, 8 (2008), 2710–2717.

[25] Schroff, F., Treibitz, T., Kriegman, D., and Belongie, S. Pose, illumination and expression invariant pairwise face-similarity measure via doppelgänger list comparison. In *2011 International Conference on Computer Vision* (2011), pp. 2494–2501.

[26] Torun, R. B., Yurdakul, M., and Duygulu, P. Finding similar faces. In *2009 IEEE 17th Signal Processing and Communications Applications Conference* (2009), pp. 960–963.

[27] Umesh, P. Image processing in python. *CSI Communications 23* (2012).

[28] Wang, X., and Tang, X. Face photo-sketch synthesis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31*, 11 (2009), 1955–1967.