

Spring 2022

Predicting Externally Visible Traits from a DNA Sample for Law Enforcement Applications

Niraj Pandkar
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_projects



Part of the [Artificial Intelligence and Robotics Commons](#)

Recommended Citation

Pandkar, Niraj, "Predicting Externally Visible Traits from a DNA Sample for Law Enforcement Applications" (2022). *Master's Projects*. 1084.
DOI: <https://doi.org/10.31979/etd.vg6d-z6qr>
https://scholarworks.sjsu.edu/etd_projects/1084

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

Predicting Externally Visible Traits from a DNA Sample for Law Enforcement
Applications

A Project

Presented to

The Faculty of the Department of Computer Science
San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Niraj Pandkar

May 2022

© 2022

Niraj Pandkar

ALL RIGHTS RESERVED

The Designated Project Committee Approves the Project Titled
Predicting Externally Visible Traits from a DNA Sample for Law Enforcement
Applications

by
Niraj Pandkar

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2022

Dr. Teng Moh Department of Computer Science

Dr. Mark Barash Department of Justice Studies

Dr. William Andreopoulos Department of Computer Science

ABSTRACT

Predicting Externally Visible Traits from a DNA Sample for Law Enforcement Applications

by Niraj Pandkar

A large majority of crimes such as homicides, sexual assaults and missing person cases are not solved within a reasonable timeframe and become cold cases. The ability to predict visual appearance and ancestry from a DNA sample will provide an unprecedented advancement in such criminal investigations. DNA based prediction of craniofacial features, phenotypes and ancestry can be used to reduce the pool of candidates onto which to perform further investigations. To achieve the above goal, it is first essential to substantiate, model and measure the intrinsic relationship between the genomic markers and phenotypic features. The first step is to standardize the 3D face scans using the CoMA data format followed by its projection into a low-dimensional latent embedding space. The second step is to reduce the dimensionality of the genetic space by performing Principal Component Analysis on the genomic markers to generate compact genomic properties. A simple multi-layer perceptron is trained to classify an ensemble of facial embeddings and genomic properties into genuine and imposter pairings. The classification model is able to match the DNA with the given 3D face with an average AUC score of 0.73. The introduction of hand-picked genomic markers serves to be an important contribution towards improving the final AUC score. Furthermore, results indicate that incorporating additional phenotypical properties such as sex and age leads to a better verification. Thus, this study proves to be an important milestone towards identifying suspects from their DNA sample in a criminal investigation.

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. Teng Moh for being a beacon of enlightenment. I have learned how curiosity can drive innovation and how persistence can fuel growth from you.

I would like to thank Dr. Mark Barash for his guidance and direction throughout this project. I am extremely grateful for the opportunity to have worked with you on this project. With your constant support and motivation, I was able to develop critical thinking and hone my personal skills.

I would also like to thank Dr. William Andreopoulos for listening to me patiently and providing valuable inputs. I also thank all the CS faculty members for providing support throughout my graduation.

Finally, I would like to thank my peers, friends and family who were by my side and helped me during my endeavour. And a special thanks to my mom and dad for always believing in me.

TABLE OF CONTENTS

CHAPTER

1	Introduction	1
2	Related Works	4
3	Data	6
3.1	Dataset	6
3.2	Need for Data Preprocessing	7
3.3	Data Preprocessing Techniques	8
3.3.1	Blender 3D Software	8
3.3.2	Mesh Registration	8
3.4	MeshMonk	8
3.4.1	Parameter Tuning	9
4	Methodology	11
5	Fusion Network	13
5.1	Generating Training Data	13
5.2	Concatenation and Scaling Features	14
5.3	Network Details	16
6	Facial Embedding	18
6.1	Convolutions in the 3D domain	18
6.2	3D Triangular Meshes and Spiral Sequence	19
6.3	SpiralNet++	20
6.3.1	Encoder-Decoder Architecture	21

6.3.2	Encoder Network	22
7	Genomic Embedding	24
7.1	Principal Component Analysis	25
7.2	Handcrafted Single Nucleotide Polymorphisms (SNPs)	26
8	Evaluation Metrics	28
8.1	Area Under Receiver Operating Characteristics (AUROC)	28
8.2	K-Fold Cross Validation	30
9	Experiments	31
9.1	Types of genomic and phenotypic properties	31
9.1.1	PCA	31
9.1.2	PCA + Sex + Age	32
9.1.3	Handcrafted SNPs + Sex + Age	32
9.2	Hyperparameter Tuning	32
10	Results and Discussion	35
11	Conclusion	39
12	Future Scope	41
	LIST OF REFERENCES	43
	APPENDIX	
	MeshMonk Registration	46

LIST OF TABLES

1	Verification Results using AUC scores	35
---	---	----

LIST OF FIGURES

1	3D Face Scan Data Sample	6
2	Meshmonk Pipeline for Template Registration [1]	7
3	Registered Template	9
4	MeshMonk Rigid Registration [1]	10
5	System Architecture Overview [2]	11
6	Data Generation for Fusion Network	14
7	Fusion Network	16
8	2D Convolutions [3]	19
9	Spiral Convolution [4]	20
10	AutoEncoder Architecture followed by CoMA [5]	21
11	Latent Space Representation of High Dimensional Face	22
12	Representation of Spiral Convolutions in Encoder network [2]	22
13	Snapshot of the Genetic Data	24
14	Principal Component Analysis	26
15	Example AUROC plot	29
16	K-fold Cross Validation [6]	30
17	Types of Genomic Embeddings	31
18	AUC Scores across Multiple Experiments	33
19	Test Losses across Multiple Experiments	34
20	AUC Scores for the Logistic Regression model	36
21	AUC Scores for the Multilayer Perceptron model	37

22	Generative Adversarial Networks	41
A.23	Non-rigid Registration (Before)	46
A.24	Non-rigid Registration (After)	47

CHAPTER 1

Introduction

Currently, the traditional forensic DNA profiling is performed by the government and private laboratories who focus on the identification capabilities of this method (e.g. matching a DNA profile from a crime scene item to a suspect via short tandem repeats genotyping). If there are no available suspects or no matches produced in the DNA database search, this approach is often resulting in a cold case. DNA molecule however, contains much more information such as about physical appearance (i.e. eye, skin and hair pigmentation, facial appearance etc.) and biogeographic ancestry of the donor (i.e. East European), which could offer that missing ‘intelligence link’, helping to solve a cold case. This approach is known as forensic molecular phenotyping and can be used to focus police investigation on a specific pool of suspects, exclude unrelated individuals and save substantial funds used to pursue a wrong investigative lead.

It is estimated that the US Police is able to close only 66% of murders and only 32% of sexual assault cases, while the clearance rates continue to decline [7, 8]. This essentially translates into over 250,000 unsolved murders accumulated since 1980s, while this number increases by approximately 6,000 cases every year. These disturbing statistics essentially means that the justice is not served for both 4 out of 10 murderers who have never been apprehended and likely continue to commit violent crimes, and for the victims and their families who do not receive a closure. In addition, there are over 600,000 individuals who go missing in the US every year, while the number of current missing person cases is estimated at 100,000, with tens of thousands of individuals remaining missing for more than one year and become cold cases [9].

However, directly generating 3D faces from DNA sequences remains an ambitious endeavour. Therefore, it is necessary to set the stage up for the eventual goal of generating 3D faces using the genetic prowess hidden in a DNA sequence. In this study,

the two major components - facial embedding relating to the 3D face and genomic embeddings relating to DNA properties are developed first. Further, a verification deep learning pipeline is built to match the genomic properties with the 3D face scans. The pipeline provides a matching score which is thresholded to validate whether the pairing of face and given DNA sequence is genuine or imposter.

This study was performed using limited training data samples. Even with a small subset of training data, a decent AUC score is reported credit to high quality spiral convolutions and smarter genomic embeddings. The usage of hand picked genomic markers serves as an important contribution towards stronger verification. The Fusion Network has been built so that it can serve as a discriminator network in a generative model in the future.

This study assumes that a database of 3D faces is available during inference for the purpose of verification with the given DNA sample. Even if the ambitious exercise of generating a 3D face of the perpetrator is successfully achieved, it won't be sufficient to apprehend the perpetrator. The system wouldn't be able to generate an accurate representation of the face because there are various external factors such as environmental changes on the face, plastic surgery and genetic and physical complexities of the human being that directly affect our ability to use this system as a standalone provider of justice. Such factors, including but not limited to damaged DNA, are beyond the scope of this study. It is meant to help the law enforcers to wiggle an investigation out of an impasse and progress the re-investigation of cold cases. The system will probably help in narrowing down the list of possible suspects

The report is organized into separate chapters for and following is a brief overview: In Chapter 2, the studies which were referred throughout the project have been briefly described. Chapter 3 talks about the existing data and the techniques used to pre-process them. Further, Chapter 4 talks about the first major component - facial

embedding - required to represent a 3D face. In the same sphere, approaches for a compact representation of the DNA sequence is described in Chapter 5. Both of these components are combined together to form a Fusion Network which is explained in detail in Chapter 6. Furthermore, Chapter 7 explains the need for AUROC as an evaluation metric and mentions the need for k-fold cross validation. Chapter 8 and 9 describe the experiments and presents the analysis of the results. Chapter 10 and 11 concludes the study and gives some concrete ideas about the possible future developments.

CHAPTER 2

Related Works

It is preferred that the input 3D scans be in a uniform low dimension to facilitate faster training of models. This computational efficiency is achieved by projecting the high-dimensional 3D faces into a low-dimensional space. Traditionally biological structures were compared using clinically visual assessment. This involved taking measurements by placing anatomical landmarks such as the tip of the nose on the target scan to establish correspondence. However, this exercise proved tedious and led to sparse landmark configurations. To combat this problem of sparse landmarks, J.D. White et al. [1] suggest utilizing surface registration algorithms to automatically establish dense configurations of quasi-landmarks. This registration algorithm involves two significant steps - rigid registration to impose the template on the target 3D scan and a non-rigid registration to warp the template onto the scan, eventually obtaining a dense correspondence.

Since uniform low-dimensional 3D scans are preferred to keep the model simple and computationally efficient, FLAME [10] defines a vertex-based model with a relatively low polygon count. In addition to facial shape features, the FLAME is trained to model pose and expression. It includes a learned shape space of identity variations, an articulated jaw and neck, and rotating eyeballs. They use co-registration and image texture to obtain high-quality alignment from a sequence of 3D scans with texture.

Once the data is in a standard format, it can be fed to a neural network architecture for further processing. Since the face data is in a 3D format, defining a 3D convolution operator becomes necessary. S. Gong et al. [4] introduce a fast and efficient intrinsic mesh convolution operator that does not rely on the intricate design of kernel function. Based on this unique spiral convolution, the authors further evaluate their method

on three different types of tasks - dense shape correspondence, 3D facial expression classification, and 3D shape reconstruction.

S. S. Mahdil et al. [2] further take this concept of spiral convolution and create embeddings of the 3D scans influenced by phenotypical properties. They call it the Geometric Metric Learner, which forms the base of their fusion net. The Fusion Net is a binary classification feed-forward neural network that takes the embedding paired with genuine or imposter phenotypes to classify the pairing as genuine or an imposter. Thus, they achieve their end goal of matching face to DNA phenotypes.

Molecular photofitting also known as DNA phenotyping can be forensically useful to estimate the phenotypes from an individual's DNA sample. P. Claes et al. [11] used bootstrapped response-based imputation modeling (BRIM) to show the effect of 24 Single Nucleotide Polymorphisms (SNPs) on the facial morphology. They talk about the genetic architecture of the facial morphology and how predictive models can be built using the said connection. The physical accuracy was maintained inspite of the effects of the 24 SNPs on the base face and they concluded that additional SNPs can be integrated as they are discovered. This observation was taken into account and used during the experiments in the current study.

In the same sphere, a study done by D. Sero et al. [12] focussed on SNPs as a the main molecular feature of interest. They divided their study into two cohorts - GLOBAL to identify the genomic background of a person and EURO for testing recognition performance in a single relatively homogenous population. An SVM classifier was built to label faces into molecular feature categories. This was followed by a Naive Bayes fuser which fused the predicted molecular labels and DNA-inferred phenotypical properties. The classifier was trained to generate genuine and imposter probabilities. Thus, a baseline to match face and DNA was generated using this fusion technique.

CHAPTER 3

Data

3.1 Dataset

We have gathered over 1,500 3D face scans and genotyped about 300 of them. Specifically, this study makes use of a sample dataset of 277 individuals. The genomic data of subjects was stored in a VCF (Variant Call Format) comprising information such as the chromosome on which the variation is being called, the position of variation in the given sequence, list of key-value pairs describing the variations, etc. The calls were read from the VCF files and stored in a readable and convenient CSV (Comma Separated Values) file.

On the other hand, the corresponding 3D scans have been stored on a google drive link. These comprise the Wavefront object files (.obj), a WRP format file proprietary to Geomagic software, a jpeg image of the subject and, fingerprints of all their ten fingers. One of the 3D face scan sample can be see in Figure 1.



Figure 1: 3D Face Scan Data Sample

3.2 Need for Data Preprocessing

In order to train any deep learning network, it is necessary to have the data in a standardized format. In the following sections, we will introduce the SpiralNet++ [4] model in which all the input 3D scanned faces need to have 5,203 vertices. It is vital to have the same number of points representing the face and the same topology so that the pretrained network can take advantage of the underlying structure. This means that the number of edges and the manner in which the vertices are connected should also be the same across all the 3D faces.

The SpiralNet++ model takes as an input 3D scan with exactly 5,203 vertices with 19,995 edges in a specific topology. The original 3D faces have an average of more than 120,000 vertices and several more edges. Therefore, reducing the overall connectivity in terms of vertices and edges of all the existing 3D scans into the required dimensions is necessary.

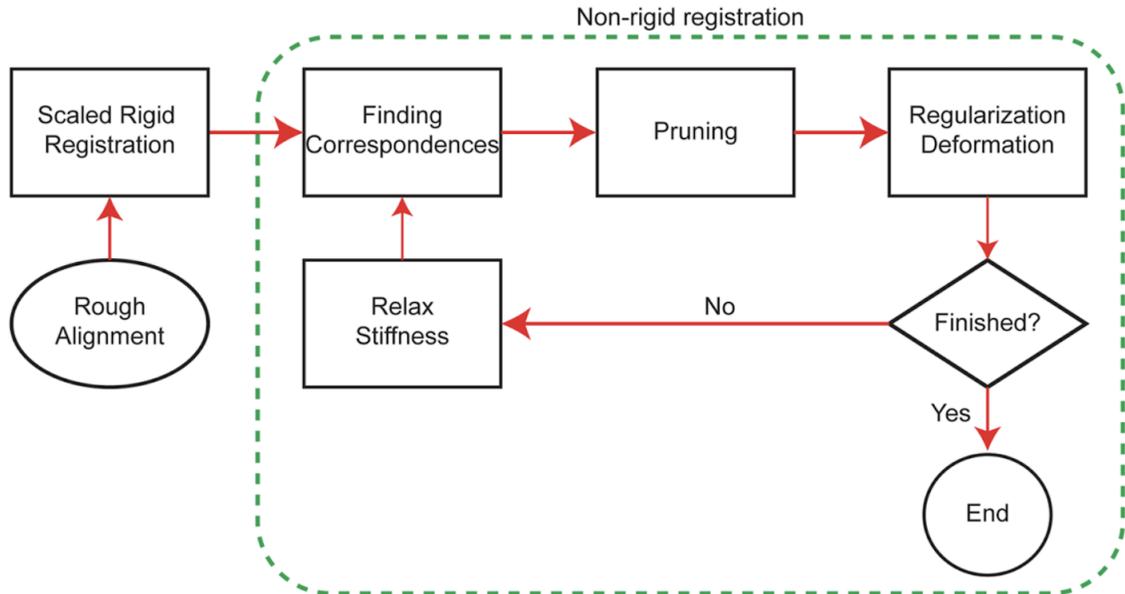


Figure 2: Meshmonk Pipeline for Template Registration [1]

3.3 Data Preprocessing Techniques

3.3.1 Blender 3D Software

The compression of the 3D scan - vertices and their corresponding edges and faces - can be done by using a software called Blender. The software gives two different options to perform this exercise - 1. DecimateModifier 2. LimitedDissolve

The Decimate Modifier allows us to reduce the mesh's vertex/face count with minimal shape changes. Limited Dissolve works by simplifying the mesh by dissolving vertices and edges separating flat regions. This can sometimes cause n-gon polygons, and hence a triangulation step is necessary immediately after to maintain the triangular mesh.

The drawback of using these two methods in Blender is that the number of vertices, edges, and faces cannot be specified and controlled. Therefore, compression usually happens based on deterministic parameters such as ratio, angle in planar, or number of un-subdivide iterations.

3.3.2 Mesh Registration

The mesh registration concept can be defined as aligning a standard template to the target 3D scan with minimal alignment error. This can be either full body registration or face mesh registration. In this case, we want to reduce the resolution of the available 3D scans into a fixed resolution space. Two different types of registration have been considered for this job. 1. MeshMonk 2. FLAME [10]

The methodology of each technique has been explained in brief and the differences have been explored in the following section.

3.4 MeshMonk

A scaled rigid registration based on the iterative closest point algorithm is performed to better align the template to the target surface. The transformation model is constrained to changing the position (translation), orientation (rotation),

and the isotropic scale of the template.

During the non-rigid registration, the shape of the template is altered to match the shape of the target surface. A visco-elastic model ensures that points that lie close to each other move coherently. The end result of registration can be seen in Figure 3 while the actual registration process on the mesh can be viewed in Appendix A.



Figure 3: Registered Template

The correspondences are updated by using pull-and-push forces (symmetrical correspondences) and the weighted k-neighbor approach. The holes and large triangles indicating badly captured or missing parts are indicated as correspondence outliers and not updated during transformation. In each iteration, correspondences are updated and outliers are identified, then an updated transformation model is used. The amount of smoothing is high (multiple Gaussian convolution runs) at the beginning iterations, when correspondences are still noisy and hard to define, and reduces gradually towards the later iterations when correspondences are more accurately defined.

3.4.1 Parameter Tuning

MeshMonk allows parameter tuning on 2 quality measures -

1. Shape fit

It is defined as the root mean squared distance of all template points to the

target surface after registration. It measures how well the shape of the template was adapted to the target shape. It can be measured over multiple images to deduct an overall quality of shape fit from the dataset [1].

2. Consistency of point indications across the same dataset

Given two models explaining the same amount of variance, the model requiring fewer parameters is favored, or given two models with the same number of parameters, the one explaining more variance in the data is favored. Principal Component Analysis is used to assess registration quality [1].

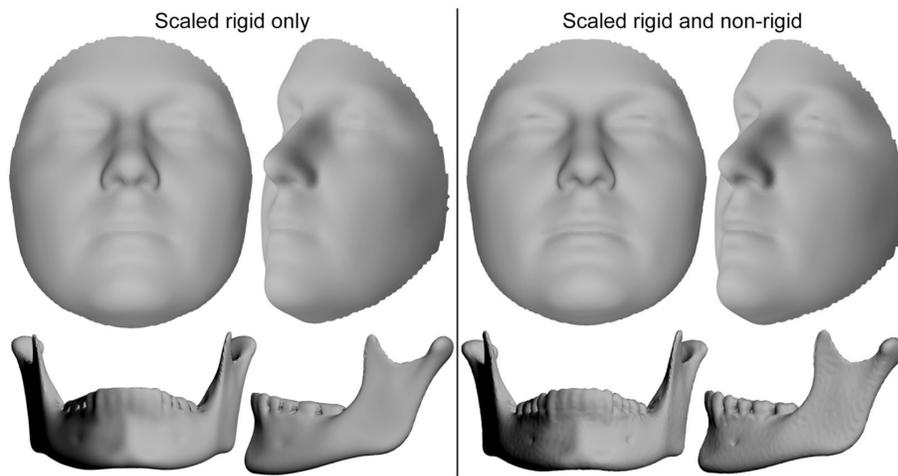


Figure 4: MeshMonk Rigid Registration [1]

CHAPTER 4

Methodology

In this study, a deep learning pipeline is constructed as a verification system. The system's purpose is to verify whether the given 3D face and a DNA sample belong to the same person. This network is called the Fusion Network [2] and it comprises of two major input components - facial embedding and genomic embedding. These embeddings relate to a compact representation of the high-dimensional 3D face meshes and genomic sequence respectively. Figure 5 gives a high-level architecture of the complete system. The input comprises of 3D face scans and genomic markers and the Fusion Network provides a binary output on whether they belong to the same person or not.

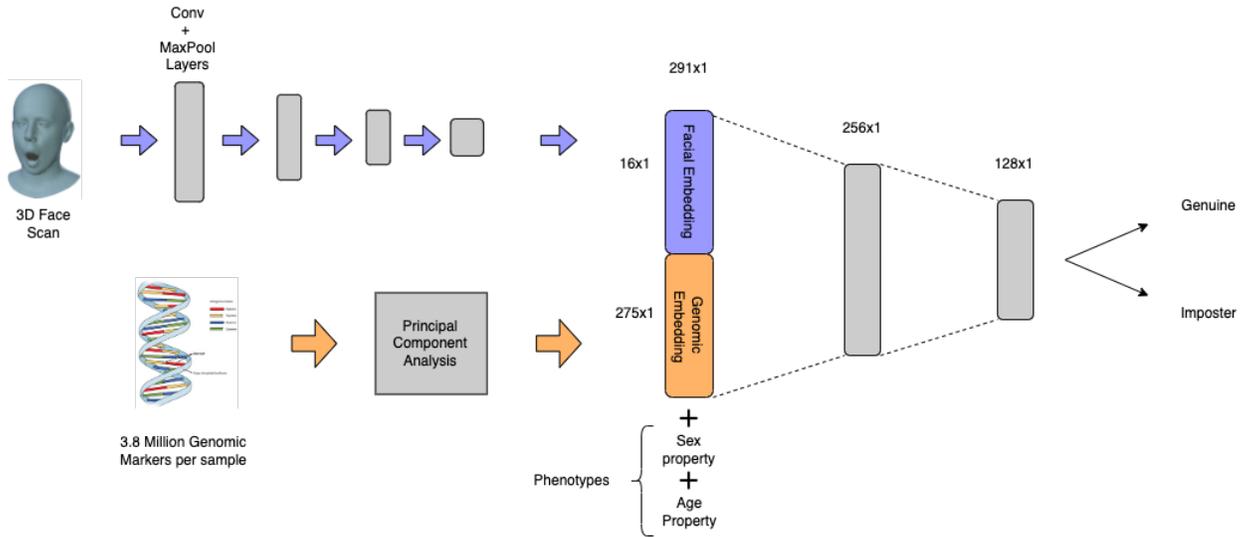


Figure 5: System Architecture Overview [2]

The Fusion Network is a feed-forward neural network which acts like a binary classifier. A conventional machine learning algorithm called Logistic Regression (LR) was also considered for the purpose of binary classification. The non-linear fashion of the neural network seemed to show an advantage over LR.

Since neural networks require all input data in a same dimension, the 3D faces are standardized and compressed into a low-dimensional space. This space is called the facial embedding. Similarly, since the DNA sequences are 3.8 million genomic-markers-long, only certain principal components are extracted using Principal Component Analysis (PCA) and thus, form the genomic embedding. Later on, it is observed that using hand-picked genomic markers instead of principal components improve the results substantially.

CHAPTER 5

Fusion Network

The Fusion Network [2] validates whether the combination of facial and the genomic embeddings are in correlation with each other. This means that verification of the inherent connection between the given face represented by its embeddings and the corresponding genomic and phenotypical properties is required. This is done by pairing up the two embeddings together and allowing the machine learning models to learn whether the given pair is genuine or imposter.

5.1 Generating Training Data

For this Fusion Network, we generated the synthetic data in the following way. During training, each embedding is shown to the network twice - first time with the correct pairing of the face embedding and corresponding genomic embedding and other phenotypical properties and second time with randomly sampled imposter properties. In each epoch, for each face, the imposter genomic embedding is randomly sampled from the remaining individual and paired up with the current facial embedding as shown in Figure 6.

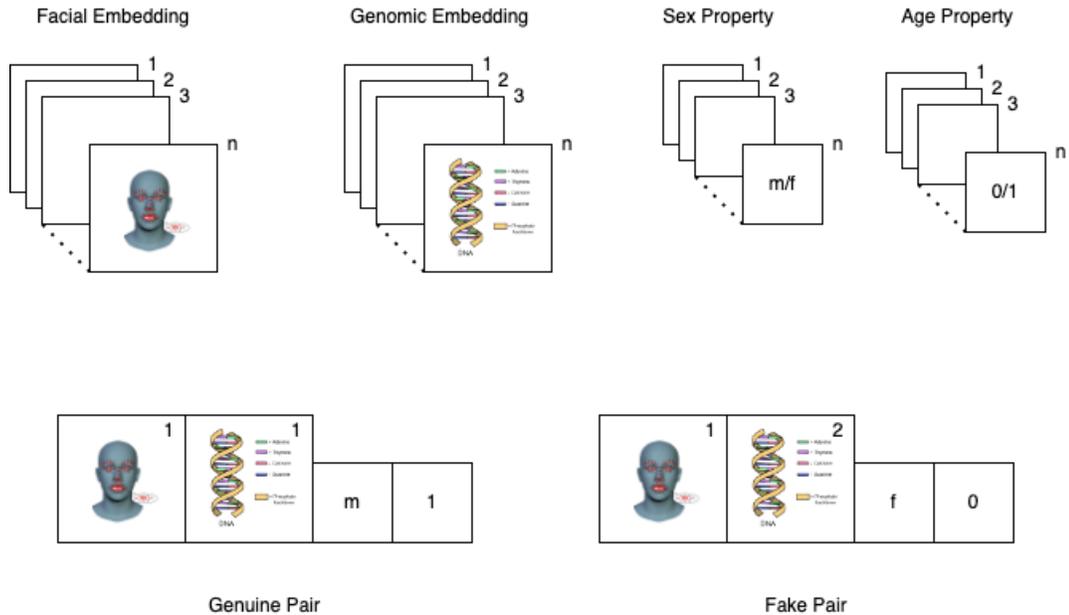


Figure 6: Data Generation for Fusion Network

The phenotypical properties such as age and sex are binarized. Sex comprises of two categories male and female. On the other hand, the numerical property of age is binarized by first finding out the median age of the existing sample which was 22 years of age. Everyone below 22 was given a label of 0 otherwise 1. Therefore for these binary traits, the fake pairing is generated by choosing a value from the other class.

5.2 Concatenation and Scaling Features

The input to the Fusion Network is comprised of different types of data - face, genome and phenotypes. These multi-modal features are concatenated together and fed to the ML models as a singular input. Since the data from different domains can have varying ranges, it is necessary to scale all the features. This helps the machine learning models to converge at a local or global minima faster.

A multi-modal input feature vector can consist of a few large values. This can

result in the model learning large weight values. A model with large weight values is often unstable and is sensitive to incoming input leading to higher generalization error. There are two ways we can scale the data -

1. Normalization

Normalization projects the existing data into a value in the range (0, 1). This requires prior knowledge of the maximum and minimum value in the dataset. But it is easy enough to calculate from the available data. The values are normalized according to the following formula -

$$y = (x - min)/(max - min)$$

Scikit-learn[13] provides a very handy function called `MinMaxScaler()` which rescales the given data in the range between 0 and 1 by default.

2. Standardization

Standardization rescaled the data such that the mean is 0 and the standard deviation is 1. It ensures that the transformed data conforms to the Gaussian Distribution (normal distribution). Standardization requires that we are able to estimate the mean and the standard deviation of the training data. The formula for standardization is as follows -

$$y = (x - mean)/standard_deviation$$

Scikit-learn library provides a handy function called `StandardScaler()` which allows for easy standardization of training data.

5.3 Network Details

A grid search was performed amongst various conventional machine learning algorithms like random forest, support vector machines (SVM) and logistic regression. Logistic Regression provided the highest scores amongst them all.

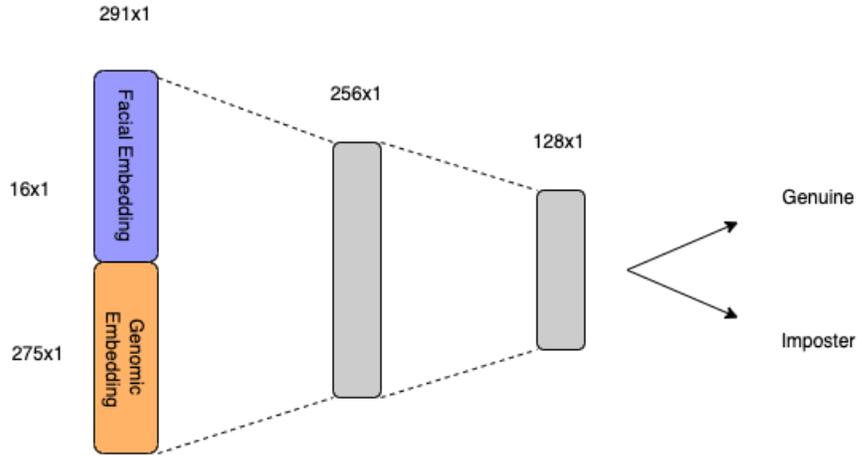


Figure 7: Fusion Network

The Fusion Network is a simple feed forward neural network acting as a binary classifier. The network is trained to classify the input pair of facial and genomic embedding into two classes - genuine or imposter. The network comprises of an input layer of dimension equal to the facial and genomic embedding. The number of hidden layers and neurons within are discussed in the Experiment section. Every hidden layer is followed by an activation function - ReLU. The popular Adam optimizer was chosen with a starting learning rate of 0.0001. The final layer was a sigmoid function which squashes the network output into a range of 0 and 1. Since the task was binary classification, a binary Cross Entropy Loss was chosen. Hyper parameter tuning was done to choose the best parameters.

According to the experiment results, a feed forward neural network is able to model the data better than a linear machine learning algorithm like logistic regression.

This may be due to the usage of non-linear activation functions such as ReLU. The neural network can also generalize well because of addition of dropout function in between layers. The dropout function drops a certain percentage of neurons in the leading layers, thus avoiding overfitting. Additionally L2 regularization of weights across the network allows the neural network to generalize well over the test set.

CHAPTER 6

Facial Embedding

The 3D face scans comprise of more than 120,000 vertices and approximately over 400,000 edges. Such a high dimensional representation is difficult to encode for the neural network. Therefore, it is essential to reduce the connectivity in the 3D mesh while preserving the overall topology. Neural networks are able to non-linearly reduce the feature space using 2D convolutions. Since, we are delving into 3D meshes, it becomes necessary to explore 3D convolutions.

6.1 Convolutions in the 3D domain

Convolutional Neural Networks (CNNs) have been a vital component in solving 2D computer vision tasks. Observing the success of this technique, it has been borrowed over into the 3D domain as well. 3D face reconstruction is a prime candidate for the application of convolutions. This technique helps in consolidating the most important parts of the 3D structure in addition to reducing the dimensionality of the problem. Reducing dimensionality supports faster training and lesser resource consumption while also retaining the topological relation (eg. nose, mouth, eyebrows, etc. on a face) of the 3D structure.

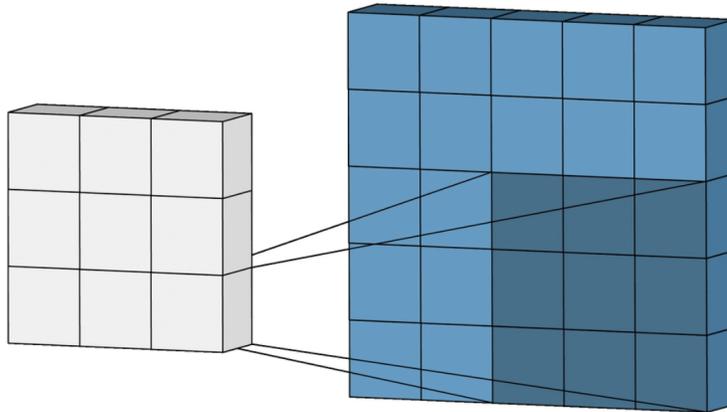


Figure 8: 2D Convolutions [3]

Convolution operators have been defined for the Euclidean domain as observed in Fig 1. A 2D kernel window slides across the 2D images to output a convolved output filter. As pointed out earlier, the authors of SpiralNet++ [4] have devised 3D convolutional operators which encapsulate the underlying non-Euclidean structure effectively.

They have defined a fast and efficient spiral convolution technique that can efficiently learn invariant shape features. This is paramount to our task of reconstructing the face with topological features such as eyes, ears, nose, etc. such that they are intact and accurate.

6.2 3D Triangular Meshes and Spiral Sequence

First, the 3D face objects (.obj, .ply, etc.) are converted into meshes. According to the open-source meshing tool OpenMesh, it gives the ability to read two types of meshes - triangular and polygonal. Triangular meshes are more efficient and hence are preferred over polygonal. SpiralNet++ also seems to use triangular meshes to

apply their spiral operator.

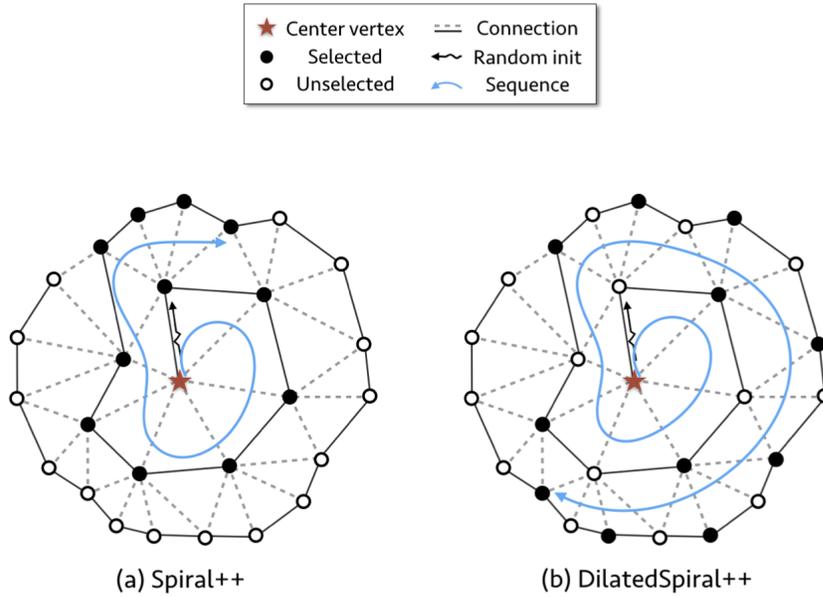


Figure 9: Spiral Convolution [4]

The spiral sequence is defined by selecting a vertex and enumerating the sequence by naturally following the spiral in either a clockwise or anti-clockwise fashion. The hyperparameters (or degrees of freedom as the authors call it) are

1. the orientation of the spiral (clockwise or anti-clockwise)
2. the starting vertex

The fixed-length serialization of the vertices allows the model to learn a high-level feature representation.

6.3 SpiralNet++

The immediate short-term goal was to map the 3D faces to their corresponding age and sex. The facial structures of men and women are distinct and characteristic. The aim will be to try to learn the structure of the face using the newly defined spiral

convolution and mapping it to the corresponding phenotypes such as pigmentation information (hair color, eye color) and other properties such as ancestry, sex, and age.

There is some skepticism on whether age can be predicted based on the facial structure alone. It would depend on how the look of the skin changes over a period of time and whether the model can generalize well for such a subtle difference in skin texture.

The authors of the SpiralNet++ paper have used the spiral convolution operator for three types of challenges - dense shape correspondence, 3D facial expression classification, and 3D shape reconstruction. Amongst these, the shape reconstruction problem defines an autoencoder architecture which will be potentially useful to map a 3D face to a phenotype.

6.3.1 Encoder-Decoder Architecture

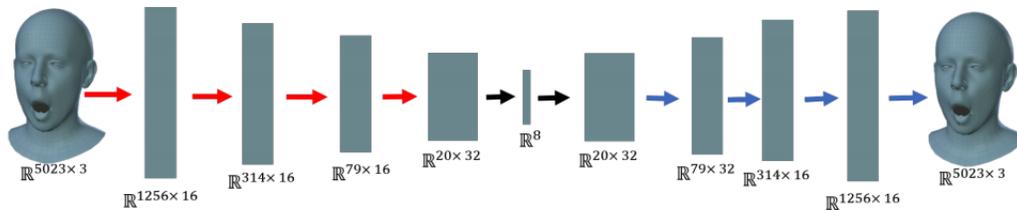


Figure 10: AutoEncoder Architecture followed by CoMA [5]

The main aim of this architecture is to project the face mesh into a latent space embedding, with minimal dimensions (\mathbb{R}^8 in the diagram), to represent the high dimensional facial structure. This latent space alone is then used to construct the whole face by upsampling operations indicating that the knowledge embedded in the low-dimensional latent space successfully encapsulates the detailed facial topological structure. The SpiralNet++ encoder-decoder architecture looks similar to Figure 10

6.3.2 Encoder Network

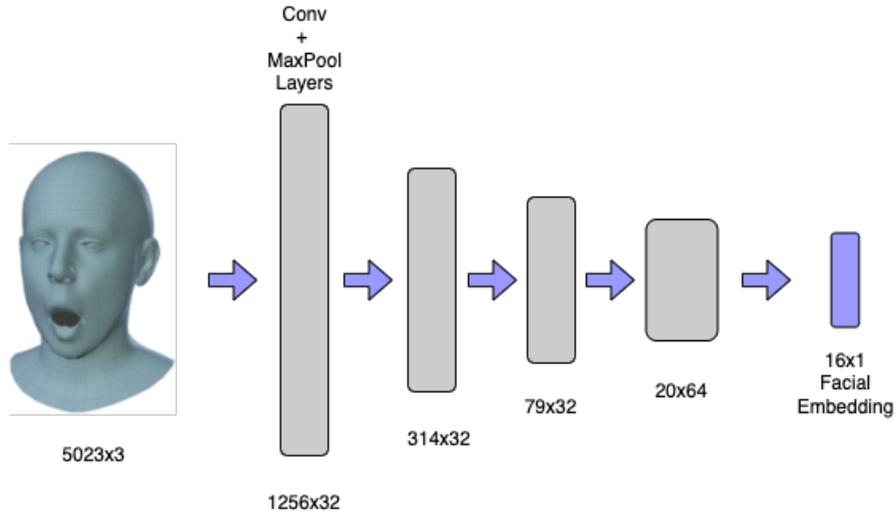


Figure 11: Latent Space Representation of High Dimensional Face

This encoder-decoder architecture is trained using the CoMA dataset [5]. Once the network is trained to encode high dimensional facial structures into latent space embedding, the encoder network, with its learned weights, is isolated. This forms one of the crucial parts of the Fusion Network called facial embedding.

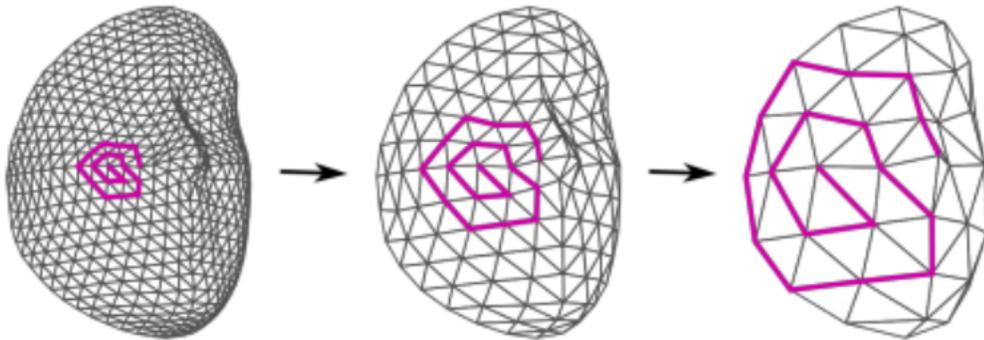


Figure 12: Representation of Spiral Convolutions in Encoder network [2]

The spiral convolution and maxpooling layers in the encoder network reduce the

size of the feature maps while retaining the overall topology of the face. This network shows a higher capacity to capture non-linear low-dimensional representations of 3D shape meshes. This method is proved to be faster than the other methods like Neural3DMM [14] and CoMA [5]. Therefore SpiralNet++ network was the perfect choice for generation of a facial embedding space.

This exercise will not only enable easy modifications in the latent embedding space but also help in faster training of neural network models. The decoder network of the autoencoder model can also be used in the future to generate a 3D face from a genetically-spiked facial embedding space. Moreover, the spiral convolutions can be adapted into a generative model to generate 3D faces. In this way, the face of a perpetrator can be sampled accurately from a low-dimensional embedding space.

CHAPTER 7

Genomic Embedding

A gene consists of millions of DNA base pairs. Most of the genes are similar in all the people but a small number of them vary from person to person. Alleles are forms of the same gene with small differences in their sequence of DNA bases. These small differences contribute to each person's unique physical features.

Since the majority of the genes are similar, it gives rise to a sparse feature vector. The feature vector for each individual in the dataset becomes 3.8 million base-pairs long. The immediate disadvantage of working with such a sparse and long input vector is memory consumption. Some machine learning algorithms require the whole input training data to be loaded into memory for further processing. This makes the execution of these algorithms difficult or even impossible given limited memory and processing power.

ID	8_198839	8_830599	8_948028	8_1779298	8_2133593	8_3735829	8_3737474
REF	C	G	C	C	C	T	A
ALT	T	A	A	T	G	C	G
1_12-010	0 1	1 0	0 0	0 0	0 0	0 0	0 0
2_12-021	0 1	0 1	0 0	0 0	0 0	1 1	1 1
3_12-022	0 0	0 0	0 1	0 0	0 0	0 0	0 0
4_12-031	0 0	0 0	0 0	0 0	0 0	0 0	0 0
5_12-041	0 1	0 0	0 0	0 0	0 0	0 0	0 0
6_12-043	1 0	0 0	0 0	0 0	0 0	1 1	1 1

Figure 13: Snapshot of the Genetic Data

Figure 13 shows a snapshot of our available genomic data. The rows correspond to the individuals who had participated in the data collection exercise. The columns indicate the genomic markers and the values correspond to the variant calls. There is usually one reference allele (REF) and one alternate allele (ALT).

The binary code for the genotype values are in the format - "0|0, 0|1 or 1|0 and 1|1" which correspond to REF|REF, REF|ALT or ALT|REF and ALT|ALT. These genotypes could correspond to the homozygosity of the major allele, heterozygosity, and homozygosity of the minor allele respectively. Further, these 3 are given multiclass values of -1, 0 and 1. For example, if our reference allele is T and the alternative allele is C, then genotype TC would be encoded as 0, TT will be -1 and CC will be 1.

One of the unintended side effects of encoding the genetic data in the above manner is the relationship formed between the different genotypes due to the sequential integers. A more unbiased encoding can be achieved by using one hot encoding approach which But it has the downside of making the genetic data sparse. This experiment remains to be explored.

7.1 Principal Component Analysis

Data sparsity is one of the aspects of curse of dimensionality. While training the model, high dimensional data leads to overfitted models and thus the models aren't able to generalize well to the given data.

Therefore, dimensionality reduction techniques such as Principal Component Analysis helps transform the high-dimensional data to a lower dimensional set of uncorrelated components. Using the available genetic data, each principal component is a linear combination of the 3.8 million genomic markers and several such principal components are chosen. Specifically, 275 principal components explained nearly 98% of the variance in the data. This was used as a baseline genomic embedding for further experiments.

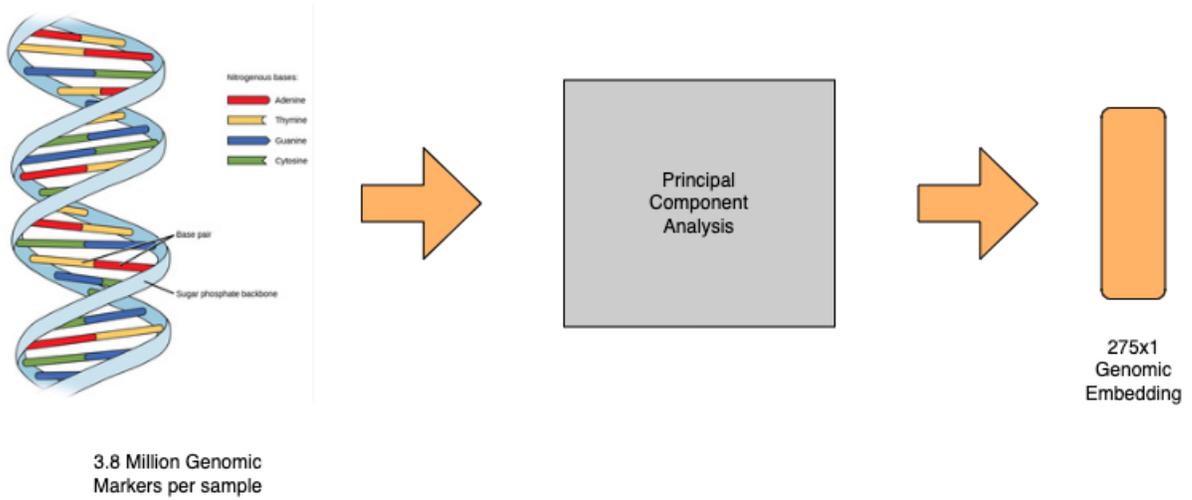


Figure 14: Principal Component Analysis

Even though the numbers indicate how the chosen components have explained almost all the variance in the data, it is essential to evaluate them qualitatively. One method is to directly map them to corresponding phenotypical features, preferably sex. It is known that 22 pairs of chromosome called autosomes are similar in both male and female. The remaining pair of chromosome relates to the sex of the person - X and Y. So the principal components chosen during the PCA exercise should be able to map accurately to sex.

7.2 Handcrafted Single Nucleotide Polymorphisms (SNPs)

There are several hundreds of genomic markers which directly correlate to the craniofacial externally visible features, pigmentation (hair colors, eye color) and ancestry. These hand-picked SNPs are cross referenced with the existing dataset of individual samples and essentially become the genomic embedding. These embeddings are further used in the Fusion Network in replacement of the PCA embeddings. Currently there are 41 SNPs which were used for this purpose.

Witnesses at a crime scene could divulge information relating to the person's age

and sex. These properties are considered as phenotypic features and work complementary to the genomic embedding. According to further experimentation, it is shown that concatenating phenotypic features increases the accuracy of the Fusion network. This is testament to the fact that the model is able to take advantage of the inherent correlation between the phenotypic features and the genomic embedding.

CHAPTER 8

Evaluation Metrics

Evaluation of the machine learning models is an important step in the deep learning workflow. The task of classification warrants evaluation metrics such as accuracy, AUC-ROC, precision, recall, sensitivity, specificity, etc. Specifically, Area Under the Receiver Operating Characteristics (AUROC) was used for the purpose of binary classification.

8.1 Area Under Receiver Operating Characteristics (AUROC)

ROC is the visualization of the tradeoff between False Positive Rate (FPR) and True Positive Rate (TPR). TPR measures the ratio of samples correctly classified as positive out of all the actually positive samples. On the other hand, FPR measures how often does the model predict a negative outcome when the samples are actually positive. TPR and FPR are calculated for every threshold and plotted on the chart. Therefore ROC is a probability curve while AUC represents the area under the ROC curve. Higher the AUC, the better the model is at predicting the two classes in a binary classification. AUC measures the separability or the degree of separation between the given classes. A good model has an AUC near 1 which means it has higher separability while a model which doesn't perform well has a score near 0.5. A score of exactly 0.5 indicates that the model isn't able to distinguish between the two classes at all. And a score closer to 0 indicates that the model has reversed its polarity for the prediction of classes.

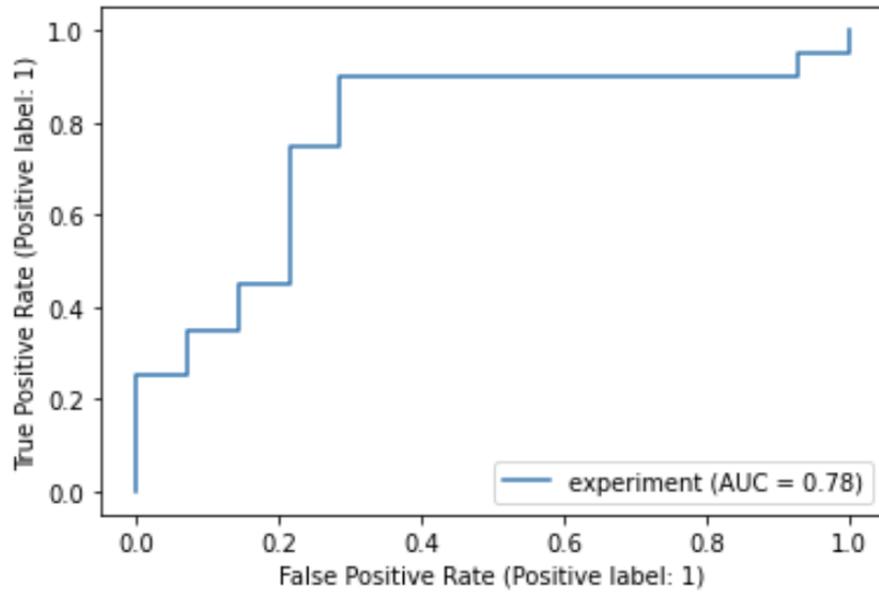


Figure 15: Example AUROC plot

The main difference between accuracy and AUC score is that accuracy is calculated on the predicted classes while ROC AUC is calculated on the predicted scores. For accuracy, a manual threshold needs to be set to distinguish between two classes. For AUC, a range of thresholds are experimented with to determine an average score. AUC is generally used when the problem warrants an equal weightage to the positive and negative samples.

8.2 K-Fold Cross Validation



Figure 16: K-fold Cross Validation [6]

Cross validation is a technique to mitigate overfitting. It is usually used to assess the effectiveness of a model by ironing out the noise. For example, there are 10 folds of the dataset out of which the first one is held out for validation while the other are used for training. In the next iteration, the second fold will be used as a hold-out set and the remaining 9 will be used to train. This allows the model to generalize well and estimate an error over k trials to get an overall effectiveness of the model.

CHAPTER 9

Experiments

The task of verifying the compatibility of the facial and the genomic embedding involved multiple experiments. Since the facial embeddings were generated using SpiralNet++, a renowned paper with tried and tested methods, they were paired with different versions of genomic embeddings. This included changing the approach of generating the actual genomic embedding in addition to complementing them with phenotypical properties like sex and age.

9.1 Types of genomic and phenotypic properties

The following experiments are designed to evaluate and measure the compatibility of the compact representations i.e. embeddings of face and DNA. For each type of experiment, there were two types of machine learning approaches employed - logistic regression and a multilayer perceptron.

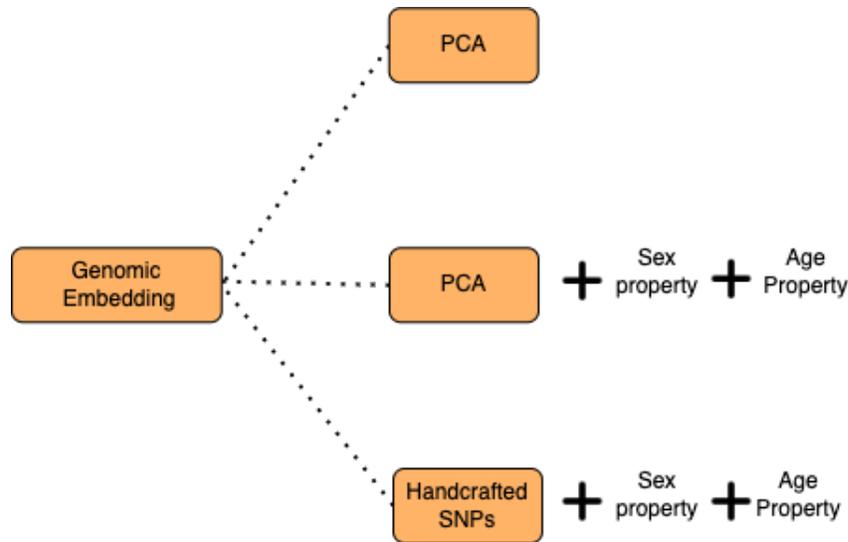


Figure 17: Types of Genomic Embeddings

9.1.1 PCA

The first experiment relates to pairing the facial embedding with PCA genomic embeddings. The input is a concatenation of 16-dimensional facial embedding with

275 dimensional genomic properties resulting in a 291-dimensional input. The model is trained with the purpose that it should recognize the given pair of embeddings as genuine or imposter. This experiment will be considered as a baseline experiment throughout this study.

9.1.2 PCA + Sex + Age

The second experiment complements the the PCA genomic components with phenotypical features such as sex and age. Both the properties are converted into binary features ($2 \times 1 + 2 \times 1 = 4 \times 1$) and concatenated to the earlier 291-D input making the input a total of 295-D. This experiment validates the contribution of external phenotypical characteristics to the genomic embedding.

9.1.3 Handcrafted SNPs + Sex + Age

The third experiment replaces the PCA embeddings with handpicked SNPs. Since these SNPs directly correlate to the craniofacial features, pigmentation and ancestry, this experiment measures the effect of these direct features onto the verification fusion model. There were 41 such SNPs selected from the individual's genomic data. This resulted in a total input size of 16 (face) + 41 (genome) + 4 (sex and age) totalling to a 61-dimensional input.

9.2 Hyperparameter Tuning

There are various parameters in a neural network which can be controlled to improve the generalization of the model. The following factors were tuned to help get a decent AUC score - wider and deeper networks, dropout, l2 regularization, learning rate regulation, activation functions, and different types of losses.

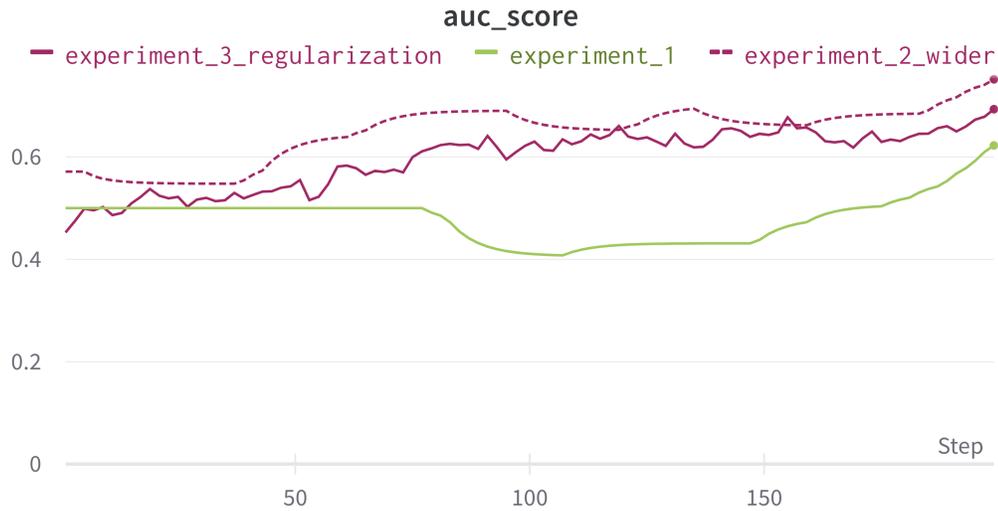


Figure 18: AUC Scores across Multiple Experiments

Figure 18 shows three experiments across various parameters. Experiment 1 was performed with 2 hidden layers with 16 and 8 neurons respectively. The plateau on the AUC score indicated that the network was stuck at a local minima and thus not able to learn quickly.

Keeping that into consideration, experiment 2 was performed by increasing the number of neurons in each layer to 32 and 16 thus making the network a bit wide. Also the learning rate was increase so that the gradient is able to push itself out of the local minima. With these changes, an increase in AUC score was observed with the test loss decreasing more rapidly as shown in Figure 19.

Furthermore, it was observed that the training loss was way lower than the testing loss. This indicated that the model isn't generalizing well to the data. This was remedied by introducing dropout of 0.2 i.e., allowing the layers to randomly drop 20% of the neurons while training. Additionally a L2 regularization was employed by introducing a weight decay of $1e-6$. This avoided the network from overfitting to the available data.

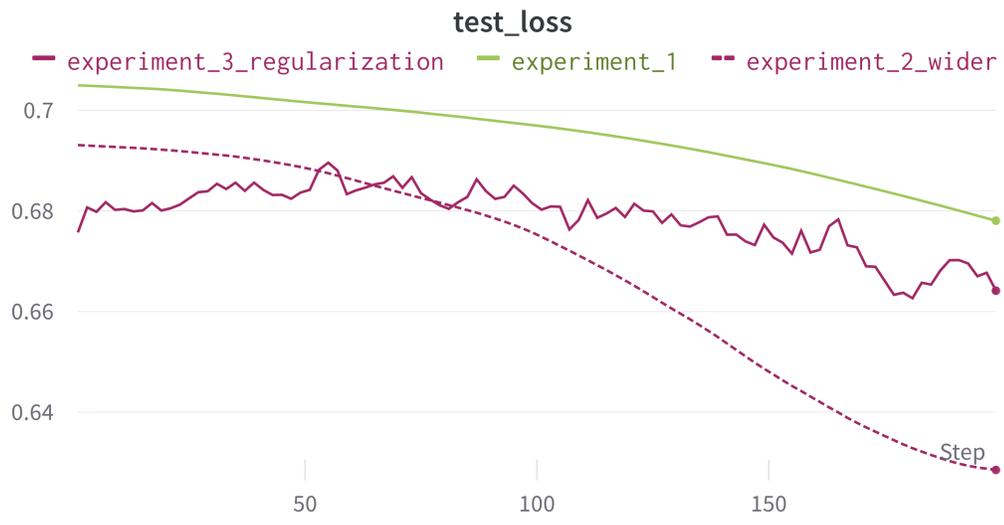


Figure 19: Test Losses across Multiple Experiments

CHAPTER 10

Results and Discussion

The goal of predicting the 3D face given a DNA sequence is a complex task with many moving parts. This study focuses on developing the major components required to achieve this end goal. Additionally, it validates the compatibility of these components - specifically the ones relating to the face and genome.

The validation exercise evaluates whether the combination of facial embedding and genomic properties is real or fake. The results are portrayed using receiver operating characteristics (ROC). The area under the receiver operating characteristics (AUROC) is a measure of how well the model is able to distinguish whether a pair is genuine or imposter. A high AUC score, closer to 1, indicates a good performance while an AUC score of 0.5 and below indicates poor performance.

Table 1: Verification Results using AUC scores

Embedding	ML Model	AUC Score(Avg. over 10 folds)
PCA	Logistic Regression	0.302 ± 0.06
	Multilayer Perceptron	0.33 ± 0.06
Sex + Age + PCA	Logistic Regression	0.585 ± 0.1
	Multilayer Perceptron	0.51 ± 0.05
Sex + Age + Handcrafted SNPs	Logistic Regression	0.618 ± 0.05
	Multilayer Perceptron	0.73 ± 0.05

According to the results, the first experiment showed very little promise in terms of proof that the facial embeddings and the genomic properties are compatible with each other. Even though the principal components generated for the genomic embeddings explained 98% of the variance in the data, the network couldn't make a correlation between the genomic properties and the facial embeddings. This resulted in a lower AUC score thus providing a suboptimal baseline result. Neither data

standardization nor any amount of hyperparameter tuning could help increase the AUC score indicating that the network isn't able to model their relationship well.

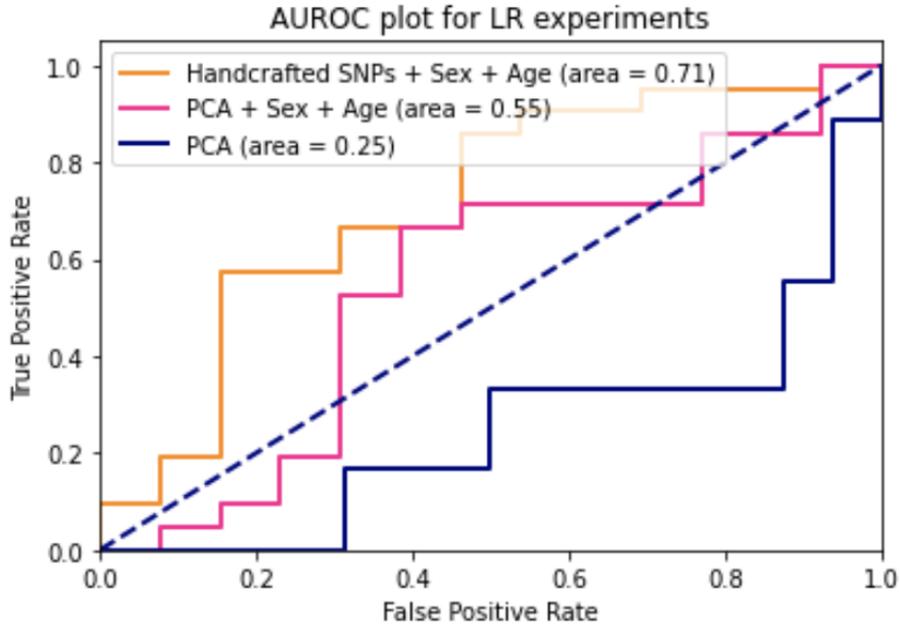


Figure 20: AUC Scores for the Logistic Regression model

Since genes directly encode information about a person's physical traits such as hair color, eye color, other facial features, and ancestry, it made sense to complement the embeddings with phenotypical properties such as sex and age. According to the results in Table 1, there was a significant increase in the AUC score observed after the addition of these two binary features. This is a testament to the fact that the phenotypical features strengthen the verification model by taking advantage of the inherent correlation between phenotypical properties and genomic embedding.

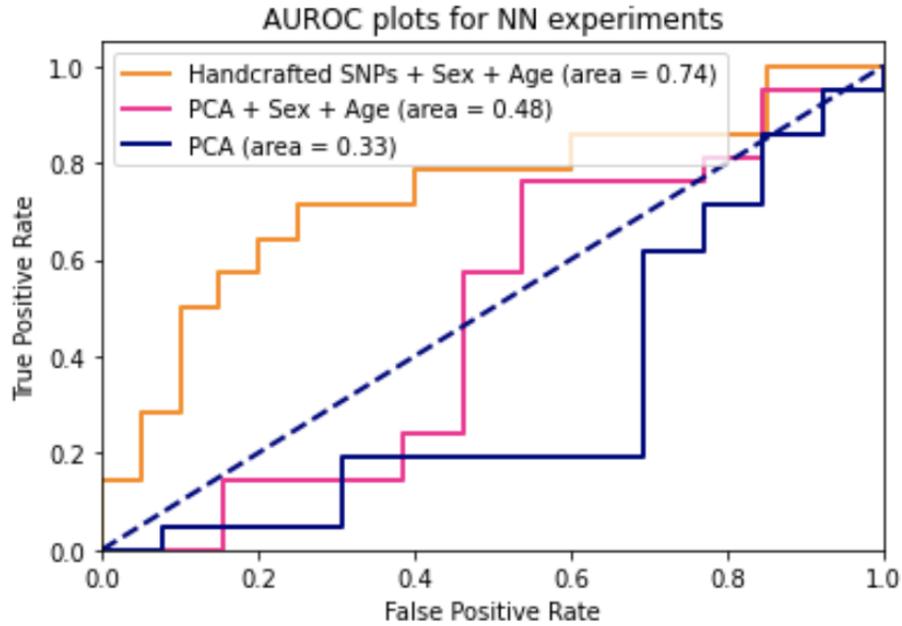


Figure 21: AUC Scores for the Multilayer Perceptron model

The third experiment replaces the genomic embedding generated using PCA by hand-picked Single Nucleotide Polymorphisms (SNPs). These SNPs are manually picked based on their contribution to the facial features, pigmentation and ancestry. The direct correlation between these genomic markers and the craniofacial properties leads us to even more accurate results. The increase in the AUC score for the simple MLP network is a testament to this fact. The AUC score jumped from around 0.55 to 0.73 which is a significant improvement.

The results in Table 1 are reported based on 10-fold cross validation. This means that these AUC scores are an average over 10 different data splits. This exercise helps in mitigating error over a wide range of test sets and removing noise overall. In all the experiments across the two methods - LR and MLP, "PCA + Sex + Age" was the only one where LR outperformed the MLP. While the training loss decreased, the testing loss increased indicating that the model wasn't generalizing well and performing poorly

on the test data.

All the experiments are performed using two ML approaches - logistic regression and multilayer perceptron. The non-linear neural network is able to model the data better than a linear ML algorithm like logistic regression. Even though the PCA embeddings explained most of the variance in the data, it is not a satisfactory representation of the genomic properties. This gave way to handpicked genomic markers which were a direct representation of the facial features including pigmentation and ancestry. It immediately reflected positively on the AUC scores and provided better contribution to the verification model.

Thus, the results indicate that the inherent relationship between the select SNPs and craniofacial features can be taken advantage of. This relationship will help in modeling the 3D face by leveraging this intelligent link from the molecular witness. Therefore, this exercise provides a good first step in predicting the intricate craniofacial characteristics from a DNA molecule.

CHAPTER 11

Conclusion

Thus, this study works towards the end goal of predicting the externally visible characteristics and has achieved an important milestone of matching the genomic and facial properties. Majorly, two data types have been introduced and worked upon - 3D face scans and genetic data. As with any deep learning project, time was invested in cleaning, preprocessing, standardizing and sanitizing the available data. This included transforming the genomic data stored in VCF files to consumable pandas-compatible CSV files. Simultaneously, the scanned 3D faces were non-rigidly registered using a standard CoMA [5] face template. This standardization allowed the usage of the 3D facial data for pretrained models like SpiralNet++. Thus, the data has been prepared and formatted to be used in future experiments.

Furthermore, geometric deep learning was used to project the high dimensional 3D faces into a latent embedding space. This projection was achieved by using spiral convolutions in an auto-encoder network proposed by SpiralNet++ [4]. Thus a SpiralNet++ model was trained for this purpose using the CoMA dataset. Therefore, a pretrained SpiralNet++ model has been made available for convenience.

In another sphere, it was observed that principal component analysis on the genomic data wasn't useful for the Fusion Network. The network couldn't model the inherent connection between the facial embedding and the genomic properties although the principal components explained almost the complete variance in the data. Furthermore, it was observed that a few hand picked SNPs (41 SNPs specifically) worked quite well for this purpose. This concludes that a better compact representation of the genetic data is required to optimally use the inherent phenotypical properties.

Additionally, during experimentation, it was observed that a well-tuned non-linear fully-connected multilayer perceptron could outperform a linear machine learning

function like logistic regression. The Fusion Network marks an important step towards a generative model because it performs a very similar job to a discriminator network. In this manner, a holistic view of the deep learning pipeline for matching the DNA and face was explained in this project.

Thus, the major components developed in this project will pave way for future studies in generation of 3D faces. This prediction of externally visible characteristics will help resurface cold cases and strengthen existing investigative leads. This project has been pursued in the hopes that it will complement the current traditional forensic approaches of identifying criminals. This work will not only help apprehend criminals but also serve justice to the victims who have not received closure.

CHAPTER 12

Future Scope

The eventual goal of generating a 3D face along with predicting its facial features from a given genomic sample is a complex task. The study in this report acts as a milestone in terms of producing components and techniques required to achieve the end goal. Automatic synthesis of real images is a task which is well suited for Generative Adversarial Networks (GANs). There has been a lot of research in synthesizing synthetic 2D images from textual descriptions using GANs. Analogically, in the current problem, a DNA sequence can act as a textual description of the 3D face.

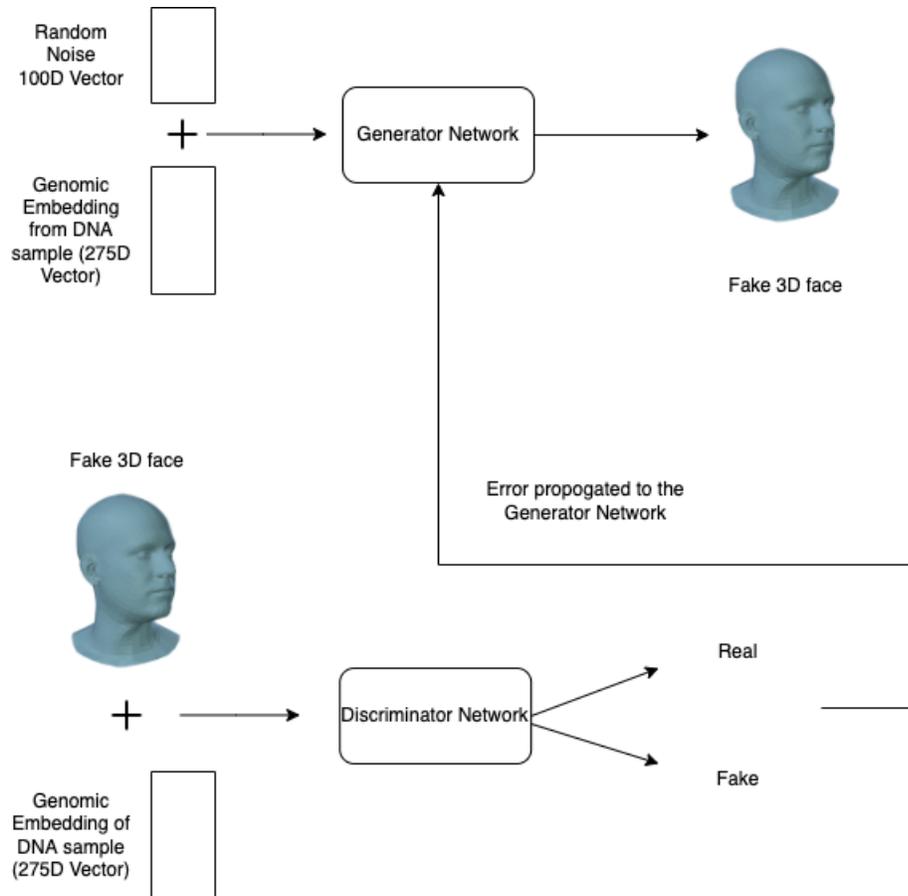


Figure 22: Generative Adversarial Networks

The formulation of facial embedding, genomic embedding and the fusion network will act as a foundation for GANs modelling. The generative network can use the genomic embedding as an input to generate a fake 3D face using spiral convolutional networks. Simultaneously, the discriminator will work very similar to the fusion network. The discriminator will act as a binary classifier to classify a pair of the generated fake 3D face and genuine genomic embedding into imposter and genuine classes. The crux of this network is the propagated loss from the discriminator to the generator, so that generator learns to produce more accurate 3D faces aligned with the given DNA sequence. It will be interesting to explore whether pretrained models like CLIP [15] and DALL-E [16] by OpenAI can be fine-tuned to work for our purpose.

In order to improve the facial embeddings, a semantic approach can be employed using metric learners [17]. This will allow the latent space to be influenced by the phenotypical properties thus providing meaningful embeddings. Similarly, a more effective projection space [18] needs to be adopted for the genomic data to leverage all the variations in genes effectively.

The Fusion Network can also be modified to generate regression scores instead of class probabilities to obtain the top-k closest 3D face lookalikes. This will work given a large 3D face database of available suspects. Additionally, experiments can be performed to validate this hypothesis by testing this theory on siblings dataset [19].

Thus, multiple solid approaches and concrete plans have been presented to continue this study. As research in the geometric deep learning field progresses, the eventual goal of generating 3D faces from just a DNA sample seems to be more in sight. This will help delegate swift justice in law enforcement and make this world a better place to live in.

LIST OF REFERENCES

- [1] J. D. White, A. Ortega-Castrillón, H. Matthews, A. A. Zaidi, O. Ekrami, J. Snyders, Y. Fan, T. Penington, S. Van Dongen, M. D. Shriver, and P. Claes, “Meshmonk: Open-source large-scale intensive 3d phenotyping,” *Scientific Reports*, vol. 9, no. 1, p. 6085, 2019, iD: White2019. [Online]. Available: <https://doi.org/10.1038/s41598-019-42533-y>
- [2] S. Mahdi, N. Nauwelaers, P. Joris, G. Bouritsas, S. Gong, S. Bokhnyak, S. Walsh, M. D. Shriver, M. Bronstein, and P. Claes, “3d facial matching by spiral convolutional metric learning and a biometric fusion-net of demographic properties,” in *2020 25th International Conference on Pattern Recognition (ICPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jan 2021, pp. 1757--1764. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/ICPR48806.2021.9412166>
- [3] “Intuitively understanding convolutions for deep learning.” [Online]. Available: <https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1>
- [4] S. Gong, L. Chen, M. Bronstein, and S. Zafeiriou, “Spiralnet++: A fast and highly efficient mesh convolution operator,” 10 2019, pp. 4141--4148.
- [5] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, “Generating 3d faces using convolutional mesh autoencoders,” 2018. [Online]. Available: <https://arxiv.org/abs/1807.10267>
- [6] “Cross-validation: K fold vs monte carlo.” [Online]. Available: <https://towardsdatascience.com/cross-validation-k-fold-vs-monte-carlo-e54df2fc179b>
- [7] “Clearance rates.” [Online]. Available: <http://www.murderdata.org/p/blog-page.html>
- [8] H. Sutton, “Sexual assault case closure rates lowest in 50 years,” *Campus Security Report*, vol. 16(3), pp. 9--9, June, 2019.
- [9] “Cold case homicide stats. project: Cold case,” 2019, accessed on 02/01/2021. [Online]. Available: <https://projectcoldcase.org/cold-case-homicide-stats/>
- [10] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, “Learning a model of facial shape and expression from 4d scans,” *ACM Trans. Graph.*, vol. 36, no. 6, nov 2017. [Online]. Available: <https://doi.org/10.1145/3130800.3130813>

- [11] P. Claes, H. Hill, and M. D. Shriver, “Toward dna-based facial composites: preliminary results and validation,” *Forensic science international. Genetics*, vol. 13, pp. 208–216, Nov 2014, IR: 20211203; CI: Copyright © 2014; JID: 101317016; 9007-49-2 (DNA); OTO: NOTNLM; 2014/06/06 00:00 [received]; 2014/08/10 00:00 [revised]; 2014/08/12 00:00 [accepted]; 2014/09/08 06:00 [entrez]; 2014/09/10 06:00 [pubmed]; 2015/06/18 06:00 [medline]; ppublish.
- [12] D. Sero, A. Zaidi, J. Li, J. D. White, T. B. G. Zarzar, M. L. Marazita, S. M. Weinberg, P. Suetens, D. Vandermeulen, J. K. Wagner, M. D. Shriver, and P. Claes, “Facial recognition from dna using face-to-dna classifiers,” *Nature Communications*, vol. 10, no. 1, p. 2557, 2019, iD: Sero2019. [Online]. Available: <https://doi.org/10.1038/s41467-019-10617-y>
- [13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [14] G. Bouritsas, S. Bokhnyak, S. Ploumpis, M. Bronstein, and S. Zafeiriou, “Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation,” 2019. [Online]. Available: <https://arxiv.org/abs/1905.02876>
- [15] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” 2021. [Online]. Available: <https://arxiv.org/abs/2103.00020>
- [16] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, “Zero-shot text-to-image generation,” 2021. [Online]. Available: <https://arxiv.org/abs/2102.12092>
- [17] S. S. Mahdi, H. Matthews, N. Nauwelaers, M. Vanneste, S. Gong, G. Bouritsas, G. S. Baynam, P. Hammond, R. Spritz, O. D. Klein, B. Hallgrímsson, H. Peeters, M. Bronstein, and P. Claes, “Multi-scale part-based syndrome classification of 3d facial images,” *IEEE Access*, vol. 10, pp. 23 450–23 462, 2022.
- [18] J. Li, T. G. Zarzar, J. D. White, K. Indencleef, H. Hoskens, H. Matthews, N. Nauwelaers, A. Zaidi, R. J. Eller, N. Herrick, T. Günther, E. M. Svensson, M. Jakobsson, S. Walsh, K. Van Steen, M. D. Shriver, and P. Claes, “Robust genome-wide ancestry inference for heterogeneous datasets: illustrated using the 1,000 genome project with 3d facial images,” *Scientific Reports*, vol. 10, no. 1, p. 11850, 2020, iD: Li2020. [Online]. Available: <https://doi.org/10.1038/s41598-020-68259-w>

- [19] H. Hoskens, D. Liu, S. Naqvi, M. K. Lee, R. J. Eller, K. Indencleef, J. D. White, J. Li, M. H. D. Larmuseau, G. Hens, J. Wysocka, S. Walsh, S. Richmond, M. D. Shriver, J. R. Shaffer, H. Peeters, S. M. Weinberg, and P. Claes, “3d facial phenotyping by biometric sibling matching used in contemporary genomic methodologies,” *PLOS Genetics*, vol. 17, no. 5, pp. 1--28, 05 2021. [Online]. Available: <https://doi.org/10.1371/journal.pgen.1009528>

APPENDIX

MeshMonk Registration

Figure A.23 shows how the standard template (blue mesh) is rigidly placed on the target template (off-white mesh). This placement is aided by the manual landmarks placed on both the target and standard template.

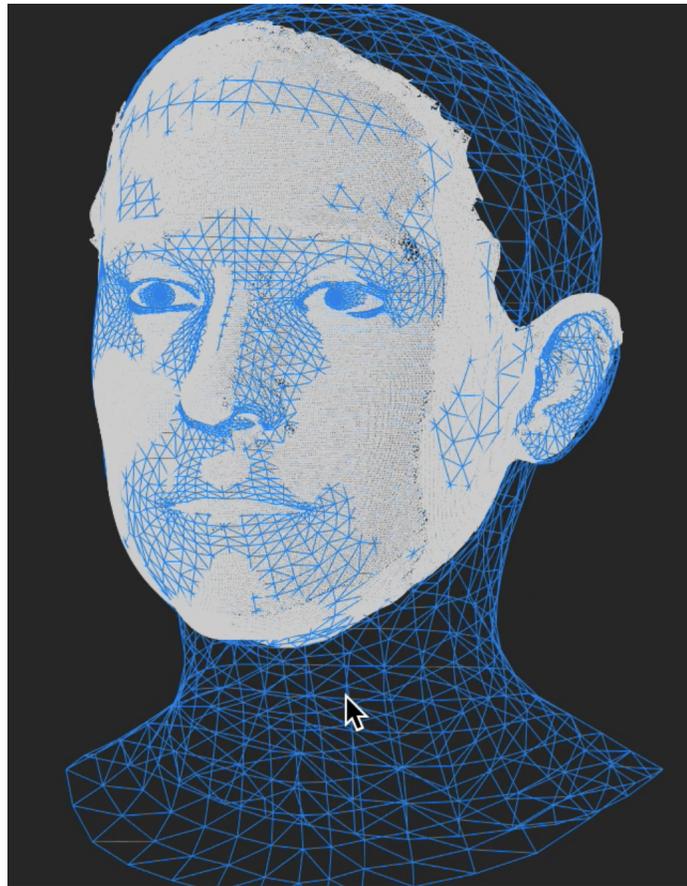


Figure A.23: Non-rigid Registration (Before)

After the non-rigid registration, it can be observed in Figure A.24 that the standard template has been altered to match the shape of the target surface.

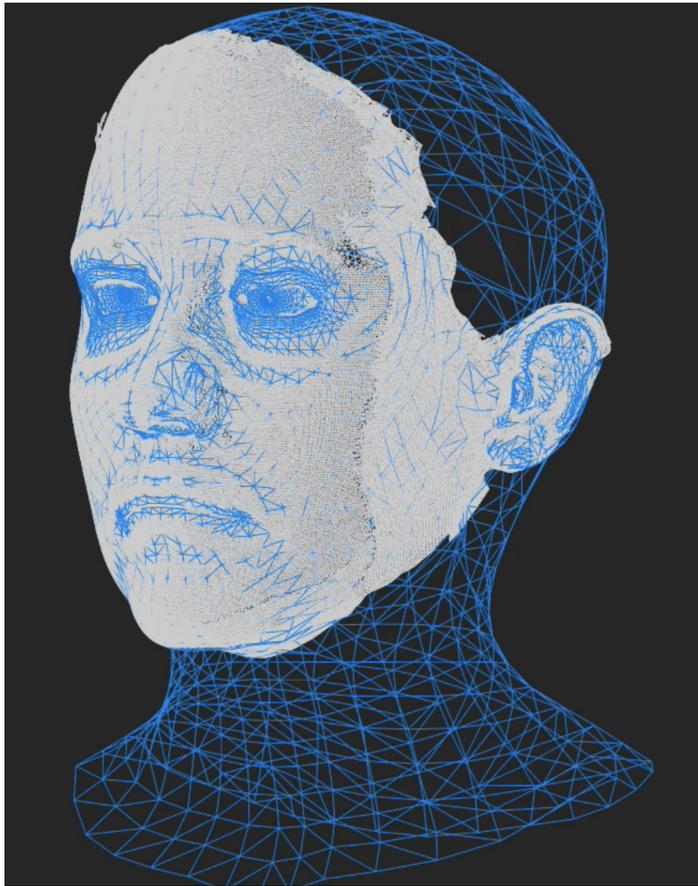


Figure A.24: Non-rigid Registration (After)