INFORMATICS INSTITUTE OF TECHNOLOGY

INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER

Hair and Beard style Recommendation System based on the Face Shape

A dissertation by

Mr. Thuwarakan Jeyakumar

Supervised by

Mr. PPG Dinesh Asanka

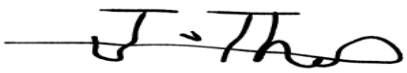
April 2023

DECLARATION

I declare that this dissertation, along with all its related sub-components, is the result of my original research. I confirm that none of the material presented has been previously submitted to any other university or organization as part of any degree or certification program. Additionally, I have appropriately cited any factual information that has been sourced from reliable outside sources.

Student Name: - J. Thuwarakan

Registration No: - 2019795 / w1790265

Signature:  **Date: 08/05/2023**

ABSTRACT

This project aims to develop a hair style recommendation system that utilizes Convolutional Neural Networks (CNNs) to classify different face shapes. The system is built using open-source libraries such as TensorFlow and Keras, and it leverages state-of-the-art machine learning techniques to provide personalized beauty recommendations.

The CNN model is trained on a dataset of images containing various face shapes, and once trained, it is used to predict the face shape of a user based on their image. The predicted face shape is then used to recommend hairstyles that would complement the user's face shape. The hairstyle recommendations are provided to the user in a user-friendly interface that includes interactive options such as selecting hair length, color, and style.

The system's performance is evaluated using metrics such as accuracy, precision, and recall. The results demonstrate the potential of using machine learning techniques for personalized beauty recommendations. The system's accuracy and precision are crucial for the reliability of the recommendations provided.

This study presents a novel approach to hair style recommendation that can be further investigated by incorporating additional features such as hair texture and color for more precise recommendations. This system has the potential to improve the customer experience at hair salons, helping clients choose hairstyles that complement their features and preferences. Additionally, it could be incorporated into beauty and wellness apps, providing users with an enhanced, personalized beauty experience.

Overall, the project showcases the utility of machine learning techniques in the beauty industry and offers new possibilities for personalized recommendations. It demonstrates the potential of combining image classification and recommendation systems to create a more personalized beauty experience. Future research could focus on expanding the system's capabilities to incorporate additional features and creating a more robust dataset for training the model.

ACKNOWLEDGEMENT

The entire process of finishing this research was quite difficult, but it also taught me a lot. It would not have been possible to complete my research without the generous assistance, direction, and drive of many individuals who stood by my side. I want to thank each of them from the bottom of my heart.

I want to thank my supervisor, Mr. PPG Dinesh Asanka, for his guidance and support throughout my final year project. From the beginning stages of generating ideas to submitting the final product, he has pushed me to do my best. His encouragement and motivation have been very helpful in motivating me to work hard and excel.

I extend my sincere appreciation to Mr. Guhanathan Poravi, the module coordinator for the final year research project, for his unwavering support and guidance throughout the project. Additionally, I would like to express my gratitude to all the lecturers at the Informatics Institute of Technology for sharing their wealth of knowledge and providing invaluable guidance throughout my degree program.

I would like to acknowledge and express my gratitude to the domain experts, technical experts, and evaluators who provided their invaluable expertise and knowledge throughout the research, contributing significantly to its success.

I am deeply grateful for the valuable feedback and consistent support I received from my friends, company colleagues, and seniors throughout the project. Their encouragement and constructive criticism played a vital role in the success of the project. Moreover, I would like to express my heartfelt appreciation to my family for being my pillars of support and motivation throughout my life. Their constant love and support have been the driving force behind my achievements.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Introduction

The introduction chapter is going to give a brief introduction to the readers about the hair and beard style recommendation system. This chapter has provided a fundamental understanding of the problem domain, the problem definition, the motivation that prompted the writer to embark on this project, existing works, research gaps, and how the researcher intends to address them. Also, this proposal briefly describes the challenges, scope, aim, and methodology with the help of this proposal.

1.2 Problem Domain

1.2.1 Fashion Industry

Fashion started when humans began wearing clothes, but it has seen explosive growth in recent years and will continue to grow. According to the research, the total value of this industry was $759.5 billion in 2021 and it is expected to grow to $1 trillion by 2025. According to analysis, humans prioritize fashion items from the stone age. The fashion industry is changing day by day with the help of technology. It has now become one of the biggest industries and it's given a lot of employment opportunities to the people. According to a study, the fashion industry employs more than seventy-five million workers worldwide (*GLOBAL GARMENT AND TEXTILE INDUSTRIES Workers, Rights and Working Conditions*, 2018) **.**The internet has helped in many ways to develop the fashion industry. So, **it is** the best time to invest our time and money into this industry. If you take the fashion industry, there are a lot of parts. For example, footwear, cosmetics, jewelry, dress style, hairstyle, perfumes, and a lot more. But in our case, we are going to see about the hair and beard style industry(Liu et al., 2019).

1.2.2 Hairstyle

The global hair style industry has become one of the biggest industries in current world. According to the research the market size of hairstyle industry become USD 20.61 Billion. This industry playing a huge role in the country economy(Shah and Hao, 2020). Nowadays most of the peoples spending lot of money to maintain their hairstyle and beard style. Most of the people having a confusion about which hair style is suitable for the but to resolve that we need to

consult with the hair stylists, but the problem is the cost for them is very expensive and need to spend lot of time(Wang et al., 2022)

1.3 Problem Definition

Finding a suitable hair or beard style has become very difficult for peoples, and the growth of the fashion industry in this decade has also played a huge role in both professional and normal life. So, if a person wants to know what the best hair or beard style is for him, he needs to approach a hair stylist or do some research on the internet and find out which is suitable for him. But those methods waste a lot of your valuable time, and if you decide to go with the hair stylist, it's more expensive. Also, if you need to find a hair stylist for men, it's hard to find the perfect one because most of them specialize in women's hairstyles, but when it comes to men's, you can't find many people.

1.3.1 Problem Statement

It is difficult to decide a suitable hair or beard style without the help of a hair stylist, but if a person wants to find a hair stylist, it's more expensive, and the availability of a men's specialized hair stylist is also very low.

1.4 Research Motivation

The main problem discussed in this proposal can be applied to most of the men who have confusion in selecting a hair or beard style. It will be hard for men to select a hair or beard style that is highly suitable for them, but with the help of image processing and machine learning, this problem can be simply solved by creating a recommendation system. This recommendation system will play a significant role in the fashion industry. Also, in terms of features, we can enhance this system and use it in various scenarios in related domains.

Nowadays, most people are concerned about their looks and styles, so the fashion industry is growing day by day. The recommendation system will help to develop the fashion industry and help to build a perfect business model.

1.5 Existing work

Table 1 - Existing work

Citation	Brief	Contribution	Limitation
(Science Council of	An image processing	The author created a	The accuracy of the MLP

Asia. Conference (16th : 2016 : Colombo, National Science Foundation (Sri Lanka) and National Academy of Sciences - Sri Lanka, no date)	model which can predict the face shape of the user using Machine learning	model that predicts the face shape. To do this, the author used a data set that contained 500 images of people's faces who belong to the age group 20–40. The author created a three-hidden layer with MLP and used a stochastic gradient descent (SGD) algorithm with a batch size of 16.	is 58 %. It can be increased if we add more valid data. Also, the author used the dataset that belongs to people who are from age 20 to age 40, so the accuracy of the prediction can change with the user's age
Shah and Hao, 2016	This research gives a brief description of how we can create a 3D model for face shape prediction from a single 2D image. In order	Author created a model which can take the 2D image as an input and process that input and create a 2D model structure according to that image In order to achieve this, the author used the dataset with the EOS-lightweight 3D Morphable Face Model (3DMM) fitting library and processed it.	The author uses a small and simple data set to train this model so the accuracy is low. Also, he didn't try the 3DCNN to classify 3D mesh.

(Alzahrani, Al-Nuaimy and Al-Bander, 2021)	An a image processing system which can recommend Eyelashes for the user (author can use the same methods to do a hair recommendation system)	The author used combine models to recommend Eyelashes. To do that he created a face shape prediction model and eyes attribute identification model. Face shape identification model trained on the histogram of oriented gradients (HOG) features with Support Vector Machine (SVM) model as classifier.	Author mainly focused on female faces so if we train with both female and male faces it increases the accuracy
(Mehta et al., 2022)	An image processing model which can predict the face shape of the user using the Machine learning	The author used three different classifiers to create the face shape prediction model. Mainly principal Component Analysis (PCA) was applied on the top three performing classifiers. Gradient Boosted Trees Classifier is mainly used to train the model.	The author used only 13 landmarks in face to predict the face shape but if we combine different algorithms, author can check more face landmarks. The accuracy of the model also 70% but if we used the Gradient Boosted Trees Classifier in a correct way author can increase the accuracy up to 90%.

1.6 Research Gap

The author, after analyzing the many research papers related to the hair and beard style recommendation system, He has identified that there are some prediction systems available for predicting the face shape and some other attributes of the face, and based on that, some suggest hair style also, but it has not been used with the combination of hair and beard style recommendation. The author has also identified that most of the existing works are focused on females or both genders, but it has not been used for males, so our system is going to mainly focus on males. Most of the research is done for a particular age group, but our system is not going to focus on a particular age group.

Therefore, this research aims to fill the gap mentioned above by creating a novel prediction system using different approaches. The author also identifies with the help of existing works that most of the time the accuracy is medium or low, so in this research the author is going to increase the accuracy and try to reduce the time also.

1.7 Contribution to the body of knowledge

In the following section the author will elaborate on the technical and domain contributions that will be made by doing the proposed project.

1.7.1 Technical Contribution

The hair and beard style recommendation systems had better accuracy than the already existing systems. In order to do that, the author is going to try various algorithms and techniques. The implemented system is a combined approach to recommend a hair and beard recommendation system. The author is also going to use the big dataset to train the model. It is also assumed that the recommendation system will be further used in other prediction-related applications as well.

1.7.2 Domaine Contribution

The recommendation system will identify the face shape of the user and recommend the perfectly suitable hair or beard style to the user. It is mainly focused on male faces. By looking at the past predictions systems, most of them focus on female faces, and most of them use data of a particular age group, but our system is going to focus on males of all age groups. It's also going to recommend hair and beard styles to users. This will also create a new pathway for new investors originating in the domain and will eventually help the domain develop more.

1.8 Research Challenge

The hair and beard style recommendation system are a little bit challenging because it directly connects with the fashion industry, and nowadays people (users) have good knowledge of the fashion industry, so the accuracy of the solution is very important because if we can't achieve high accuracy, we can't satisfy the users. So, users can simply reject our system, but achieving a high accuracy model is a little bit hard, and when we suggest the hairstyle, we also need to consider various aspects like the person's face shape, his hair type, and a lot more. This is also a little bit challenging because we are going to use a single image of the user, so the quality of the image will play a huge role in this system. Also, if we consider the beard style, it's more challenging compared to the hairstyle. Currently, there is no existing system for predicting hair and beard styles, so we need to do a lot of research in this field. To achieve the aim, the author needs to test a lot of models and also needs to combine 2 or 3 models, which is very hard in a limited time period. Another big challenge is achieving the deadlines because the author needs to test and evaluate more than one model. He also needs to check all possible methods and needs to find the best method.

1.9 Research Question

RQ1: What are the technics and algorithms we can use to develop the face shape prediction model?

RQ2: What are the technics and algorithms we can use to develop the face identification and gender identification model?

RQ3: What is the recent advancement in recommendation models & architectures that can be taken into consideration when building a hybrid Recommendation Architecture?

RQ4: How can we create a recommendation system which can suggest hair or beard style for the users?

1.10 Research Aim

The aim of this research is to design, develop, and evaluate a recommendation system that can analyze the face of a user and predict the face shape of the user. Then, according to the face shape, it will recommend the hair or beard style for the users.

To further elaborate on the aim, this research project will produce a hairstyle prediction model for fashion designers and normal users. This system will allow users to add their face images, and the system will process that image and predict its shape, and according to that shape, it will suggest the hair and beard style that is suitable for the user.

1.11 Research Objectives

Based on the research aim and research questions, the following objectives were defined in the Project and Research levels. It can be found in **Appendix 01 – Research Objectives**

1.12 Scope

The scope is defined as follows based on the project objectives and a review of existing products with consideration to the time granted for this research project:

1.12.1 In Scope

The scopes of the project are: -

- Creating a system which can identify the users face.
- Creating a system which can identify the user's gender.
- Creating a system which can predict the users face shape.
- Creating a system which can recommend the hairstyle to the user based on the face shape.
- Creating a system which can recommend the beard style to the user based on the face shape.
- Graphical user interface for which can upload your image and display the result.

1.12.2 Out Scope

The following are the parts that will not be covered by the project:

- This system will not recommend the Mustache style or not consider that.
- This system will not recommend any face makeup ideas.
- System will not consider about the hair type and other thing when recommending the hair or beard style.

1.12.3 Prototype Diagram

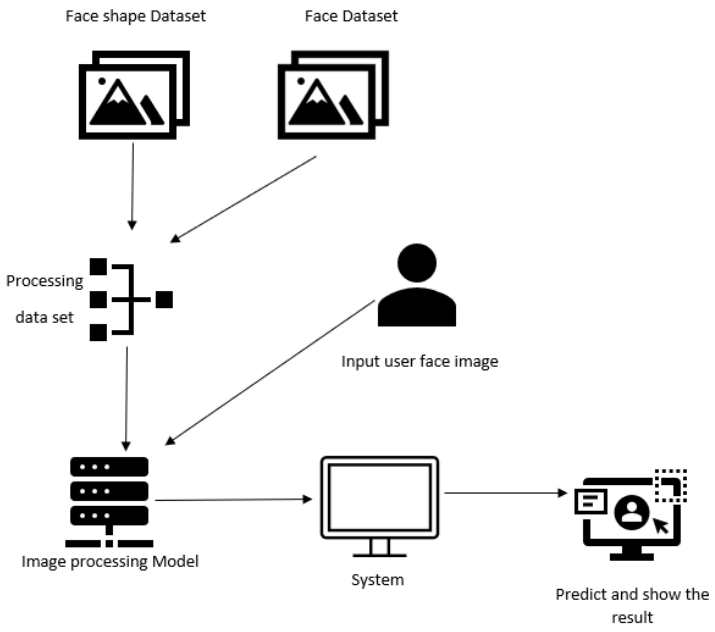


Figure 1 - Prototype Diagram

Above the prototype diagram is a description of how the prototype will work. Here, we are training the model with existing face and facial shape data. The system will get a picture of the user, use the trained model, and suggest a hairstyle or beard style to the user.

1.13 Resource Requirements

Based on the research objectives, aim, expected solution and deliverables the requirements are defined. The software, hardware and data resource requirements are as follows.

1.13.1 Hardware Resource Requirements

Table 2 - Hardware Resource Requirements

Hardware requirements	**Justifications**
Core i7 10th generation processor	able to perform highly intensive tasks
8 GB RAM or above	To manage heavy image data sates and development environment
Disk space of 50 GB space or above	To store models, large data sets, codes and documents which are related to the research

Graphics Processing Unit (GPU)	To training a model a powerful GPU is necessary for that google colab is a best choice.

1.13.2 Software Resource Requirements

Table 3 - Software Resource Requirements

Software requirement	Justification
Operating System (Windows 10)	The operating system required to handle the enormous complexity works. A 64-bit OS with the latest version will be able to carry out such a resource-intensive task. Windows also supports multiple software which can help in this research.
Python	The language that was used to develop the image processing model was chosen because it has a rich collection of rich libraries, easier and faster compared to other programing languages.
OpenCV / Scikit-Image / TensorFlow/ Keras	Image processing The Python libraries are very helpful in developing an image processing model because they include a lot of pre-defined functions, which can make our process easier
VS code	Integrated development environments (IDE) help a lot in writing codes. Out of available IDEs VS code is a best code editor to write python codes.
Mendeley	A reference management software is most important because we need to handle a lot of references in various reports, so it makes it easier.
GitHub	Version controlling and backup codes
MS Word	Tool to create reports and documents
Microsoft PowerPoint	Tool to create presentation
Draw.io	Tool to create diagrams
Google drive	Tool to backup important documents and files

1.13.3 Data Requirements

- Face image dataset – From Kaggle and Google Dataset Search
- Face shape dataset – From Kaggle

1.13.4 Skill Requirement

- Knowledge in image processing

  To create an image processing model, we need a vast knowledge of image processing. I especially need knowledge of theories, algorithms, and more background knowledge.

- Creative designing skills

  In order to create a best system, the developer need a good designing skill

- Academic writing

  Most of the time, the author needs to submit his findings in a report format. In order to do that, he needs a strong knowledge of academic writing.

- Knowledge in external image processing libraries.

1.14 Document Structure

This Chapter starts with an overview of the issue, supporting arguments, preliminary planning, and research goals. Definitions of the study's scope and a preliminary analysis of the body of work are also documented.

The literature review is described in chapter two. In terms of domain, technology, and products, the existing work and systems are examined and analyzed.

The project's approaches are described in Chapter three of the report. The approaches for research, management, and development are all covered, as well as activity calendars and job breakdowns.

The methods used to elicit requirements are described in chapter four. The many techniques for gathering requirements, identifying requirements, and comparing their benefits and drawbacks are reviewed.

Chapter five covers the Social, Legal, Ethical and Professional Issues related to the research project.

The design components of the project are covered in chapter six, along with a number of design choices, diagrams of the class, domain, contexts, and system architecture, as well as detailed explanations.

The project's implementation is covered in chapter seven. The chosen tools and technologies, development choices made, issues encountered, and developed algorithms are all documented.

The testing of the system is covered in the eighth chapter. It covers the various functional and nonfunctional testing carried out, its outcomes, and benchmarking with current systems.

The system evaluation is covered in chapter nine. The selection of several evaluators, the justifications, the evaluations provided, the authors' evaluations, and their debates are addressed.

The author's concluding views are covered in the tenth chapter. Review of the aims and objectives, adherence to social and security policies, and potential improvements are reviewed.

1.15 Chapter Summary

This chapter provided a basic understanding of this final year project. This chapter mainly defined the project of this research project. It clearly described the problem domain, problem background and research gap. The aim, scope, objective and challenge of the project is elaborated clearly in this chapter. Also, in this chapter provided the contribution of the researcher also explained clearly. The prototype feature diagram illustrates what are the features and components of the system. Finally, hardware, software, data, and skill requirements were identified as essential to complete this project.

CHAPTER 2: LITERATURE REVIEW

2.1 Chapter Overview

The fashion industry is increasingly relying on software solutions to improve the customer experience, and image classification using Convolutional Neural Networks (CNN) has become a popular tool for various tasks. Combining these technologies can lead to a useful product for society, such as a hair and beard recommendation system. In this chapter, the author will outline their research approach and explore existing works in the areas of image classification, hair recommendation, and beard recommendation. This review will cover the strengths and weaknesses of various approaches, including the algorithms they use, supported features, and implementation details. By evaluating these approaches, the author aims to identify best practices and opportunities for improving their own model.

2.2 Concept Map

As the initial step, a concept map was drawn to represent the scope of the literature review. The concept map is found in **Appendix 17 - Concept Map.**

2.3 Problem Domain

2.3.1 Hair & Beard style recommendation

The fashion industry has witnessed significant growth over the past few years, with a surge in the integration of software technologies to enhance the overall customer experience. This has led to the development of personalized styling applications that provide recommendations for dresses, shoes, perfumes, hair, and other fashion items.(Here Are the Best Haircuts for Your Face Shape, 2017) "Which hairstyles or beard style should we wear?" is a question that people have been asking themselves since the beginning of human civilization. Hair and beard styles have been influenced by various aspects such as culture, face shapes, personal preferences, and trends. The biggest problem of finding a suitable hair or beard style is that we often have to rely on the expertise of others, such as hair stylists and salons. This can be a time-consuming and expensive process, especially if we need to experiment with multiple styles to find the one that suits us best. Additionally, we may not always get the desired outcome, and it may take a long time for our hair or beard to grow back to the desired length or style(Pasupa, Sunhem and Loo, 2019a) . This is where a hair and beard style recommendation system come in handy. Such a system can help individuals choose a hairstyle or beard style that complements their face shape, skin tone, hair

texture, and other factors. By utilizing image classification techniques, a recommendation system can analyze a person's facial features and suggest suitable hairstyles or beard styles based on the individual's preferences and other relevant factors. This not only saves time and effort but also ensures that individuals make informed decisions when choosing a new hairstyle or beard style. (Emmanuel and Tio, 2019)

2.3.2 Face shape classification

When recommending a suitable hair or beard style, several factors need to be considered, including face shape, hair type, hair color, beard color, culture, and age. These factors play a crucial role in determining the most appropriate style for an individual. However, the most important factor to consider is typically the face shape, as it has a significant impact on which styles will look best. Once the face shape is determined, it can be used to narrow down the suitable styles. Other factors such as hair type, maintenance, personal style, and occasion can also be taken into account to make a final decision on the ideal hair or beard style to choose.

There are various technological advancements that have made it possible to accurately classify an individual's face shape, which is crucial in determining suitable hair or beard styles. One such advancement is the development of a face shape prediction model that utilizes machine learning algorithms to classify face shapes. This model can be trained using a vast amount of facial data and can provide accurate predictions for a wide range of face shapes. By integrating this technology into a recommendation system, experts and stylists can suggest suitable hair or beard styles for users based on their individual face shape. This not only enhances the accuracy of the recommendation system but also provides a more personalized experience for the user. Furthermore, this technology can also take into account other factors such as hair type, hair color, and culture to provide a comprehensive recommendation. With the use of such technology, individuals can be assured that the recommended hair or beard style will be the most flattering for their face shape, resulting in a better overall experience(Vittal, Gowda and Sankapal, 2022).

After considering the importance of face shape in recommending hairstyles or beard styles, the author concludes that to create an effective recommendation system, it is essential to determine an individual's face shape. The author suggests using new technologies to accurately predict face shapes, which can be achieved through machine learning algorithms. However, it is crucial to note that there are various face shapes in the real world. To simplify the process, the author

recommends focusing on the five main face shapes, namely oblong, oval, square, round, and heart-shaped faces. By doing so, the recommendation system can accurately suggest hairstyles or beard styles that are the most flattering for the individual's face shape. (Emmanuel and Tio, 2021a)

Table 4 - Face shape description

Face Shape	**Description**
Heart	The width of the forehead is greater than the width of the jawline and the chin may be pointy
Square	the widths of the forehead, cheekbones, and jaw are equal and there is a sharp jawline
Round	the widths of the forehead, cheekbones, and jaw are equal, and the jaw is slightly round as opposed to angular
Oval	the height of the face is very long compared to the width.
Oblong	the ratio of the height to width is 3/2.

Overall, identifying an individual's face shape is a crucial step in creating an effective recommendation system for hairstyles or beard styles. By utilizing new technologies and focusing on the main face shapes, the recommendation system can provide personalized suggestions that enhance an individual's appearance and confidence. (How To Determine the Shape of Your Face And 6 Different Types, 2021)

2.3.3 Machine Learning Approaches

Machine learning is a branch of artificial intelligence that enables computers to learn from data and experience without being explicitly programmed. It involves the development of algorithms and statistical models that allow computers to identify patterns and make decisions based on data inputs. The field of machine learning has grown rapidly in recent years, with advancements in data processing and computing power leading to breakthroughs in areas such as image recognition, speech recognition, natural language processing, and autonomous vehicles. Machine learning has a wide range of applications, from predicting consumer behavior to detecting fraud and diagnosing medical conditions. It is an exciting and rapidly evolving field with many opportunities for innovation and impact.(Bell, 2022) To elaborate, the vast development of machine learning has opened up new possibilities for creating a hair and beard style

recommendation system. By utilizing machine learning approaches, the author can create a face shape prediction system that accurately classifies an individual's face shape and recommends suitable hairstyles or beard styles accordingly(Emmanuel and Tio, 2021).

In this context, an image classification system is needed to classify face shapes. To achieve this, the author needs to analyze various machine learning techniques, including Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, Reinforcement Learning, and Deep Learning. Among these techniques, Deep Learning is well-suited for image classification tasks due to its ability to automatically learn hierarchical representations of the input data. Deep learning models can recognize features at multiple levels of abstraction, from low-level features like edges and textures to high-level features like shapes and objects. Therefore, a Deep Learning approach is appropriate for classifying users' faces into the 5 face shape classes. Furthermore, Deep Learning can handle large image datasets and is well-suited for image classification tasks due to its capacity to capture complex patterns in images and learn from large amounts of labeled data (Mahesh, 2018).

Another reason to select the deep learning approach is it have several algorithms such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Generative Adversarial Networks (GANs), Autoencoders, Deep Belief Networks (DBNs), Deep Boltzmann Machines (DBMs), Recursive Neural Networks (RecNNs), Long Short-Term Memory Networks (LSTMs), Deep Q-Networks (DQNs), Self-Organizing Maps (SOMs) and many more among these author can use Convolutional Neural Networks, Recurrent Neural Network, Generative Adversarial Networks and MLP (Multilayer Perceptron) for image classification but cossider various thing author plan to approach the project with CNN algorithms.

the selection of the deep learning approach is based on the availability of various algorithms suitable for image classification tasks. Some of the commonly used algorithms in deep learning include Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Generative Adversarial Networks (GANs), Autoencoders, Deep Belief Networks (DBNs), Deep Boltzmann Machines (DBMs), Recursive Neural Networks (RecNNs), Long Short-Term Memory Networks (LSTMs), Deep Q-Networks (DQNs), Self-Organizing Maps (SOMs), and many others(PROCEEDINGS PROCEEDINGS III RESULT). among these authors can use

Convolutional Neural Networks, Recurrent Neural Network, Generative Adversarial Networks and MLP (Multilayer Perceptron) for image classification.(Ao, Rieger and Amouzegar, 2010)

However, considering various factors, the author plans to approach the project with the Convolutional Neural Networks (CNN) algorithm. CNN is a widely used deep learning algorithm for image classification tasks because it can automatically extract features from images by using a series of convolutional layers. Moreover, CNN is capable of handling large datasets and can produce high accuracy in image classification tasks. Thus, CNN is a suitable choice for the author to classify users' faces into the 5 face shape classes(Abdullah et al., 2022).

2.3.4 Why was CNN chosen in this Research?

CNNs are specifically designed to process images and have shown impressive results in image classification tasks. The main reason for choosing CNNs in this research is that they are capable of handling large amounts of data and can identify complex patterns in images. Furthermore, CNNs can learn features that are invariant to translation, rotation, and scaling, which makes them suitable for face shape prediction and facial feature detection tasks(Sun et al., 2014). In addition, CNNs have several other advantages that make them good choice for image classification. For example, they are capable of automatically extracting features from raw pixel data, which reduces the need for manual feature engineering. Also, they can learn representations of features at multiple scales and can capture both global and local information in the images(Vittal, Gowda and Sankapal, 2022).

CNNs have several advantages over other deep learning algorithms such as Multilayer Perceptron (MLP), Generative Adversarial Networks (GAN) and Recurrent Neural Networks (RNN). First, MLPs are fully connected neural networks that require a large number of parameters to be trained, which makes them computationally expensive and let the model to overfitting. On the other hand, CNNs have a much smaller number of parameters, which makes them easier to train and less prone to overfitting. Like wish CNN is better than other algorithms like ANN, KNN and GAN(Pasupa, Sunhem and Loo, 2019).

2.3.5 Introduction to Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are a type of deep learning algorithm that is widely used for image recognition, classification, and segmentation tasks. CNNs are designed to process data with a grid-like topology, such as images, by applying a series of filters that extract relevant

features from the input data (Bird and Faria, 2018). CNNs consist of several layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers apply a series of filters to the input data, each of which extracts a specific feature, such as edges or corners. The pooling layers down sample the output of the convolutional layers, reducing the spatial dimensions of the feature maps and making the network more efficient. Finally, the fully connected layers apply a set of weights to the features extracted by the convolutional and pooling layers, producing the final output (Mayank Mishra, 2021).

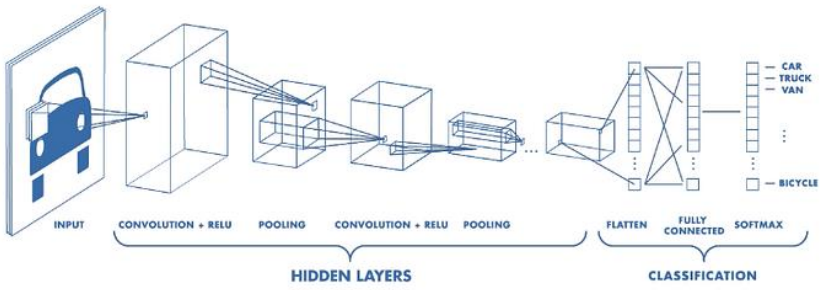


Figure 2 - CNN Layers

One of the key advantages of CNNs is their ability to learn hierarchical representations of the input data. The first layers of the network learn simple features, such as edges and corners, while the later layers learn more complex features of that image. This will help to learn about the faces in different stages and help to increase the accuracy of the model when it comes to predicting the face shapes.

2.4 Existing Work

This research containing face shape prediction model for suggesting the hair style for the user in here author predict the face shape using the model and based on that face shape author showing suitable hair style.

The selection of an appropriate architecture for training an image classification model is a critical step in creating an accurate and efficient system. This is because different architectures have varying strengths and weaknesses, and the optimal architecture for a particular task depends on factors such as the size of the dataset, the complexity of the images, and the computational resources available. To select a suitable architecture for their image classification task, can follow different approaches.

2.4.1 Basic CNN architecture

Training a model from scratch is one of the methods to do the classification model. (Baixian et al., 2018) in this research paper author used 640 images for each face shapes and using a binary feature image to a shape encoding map method to preprocess the dataset.

This researcher used that preprocessed dataset to create the CNN model which contains 2 two convolution layers and two pooling layers as well as a fully connected layer. Each convolution layer consists of 20 convolution kernels with a receptive field of 5×5, using ReLU as the activation function for the convolution layer. The two pooling layers adopt max-pooling operation this research ended up with a good accuracy around 76% but the main problem of this research is overfitting and the researcher suggesting s VGG and DenseNet architecture to increasing the accuracy.(Baixian et al., 2018).

Face shape prediction helps for various ways in this research author using the face shapes for hair and beard style recommendation but in this research (Young et al., 2019) the researcher using face shape for suggesting eyeglasses which author can use it for extend the system in here the researcher using 650 images of oval and square which resize to 28 × 28. To create the model, he used the Sequential API with two Convolutional Layer, two Pooling Layer and one fully connected layer. The accuracy of this model is around 75% also this researcher trained Multilayer perceptron (MLP) model also in this research to compare MLP and CNN technics and the accuracy of CNN model is better that MLP model.

2.4.2 VGG architecture for face shape prediction

VGG (Visual Geometry Group) is a convolutional neural network architecture that was introduced by researchers at the University of Oxford in 2014. VGG is one of the most popular and influential deep learning models, and it has achieved state-of-the-art performance on various computer vision tasks such as image classification, object detection, and segmentation. (Emmanuel and Tio).

(Mehta et al.) in this research the author used the VGG structure. VGG is one of the commonly used architectures in the CNN field. In here the author combined 2 different datasets and created a dataset which contains 1000 images for each face shapes. The original VGG network has several versions, with the most common ones being VGG16 and VGG19, which have 16 and 19 layers, respectively. These models were trained on the ImageNet dataset, which contains over 1

million images and 1,000 different classes. In here the author used both versions and VGG16 show accuracy of 72% and VGG19 show the accuracy of 76% so based on this research we can say using VGG19 is a better choice for face shape prediction.

2.4.3 Inception architecture

Inception is a CNN architecture designed by researchers at Google in 2014, that is widely used for image classification and object detection tasks. The core idea behind Inception is to use a combination of small filters in parallel, which allows the network to capture both low-level and high-level features of the input image, while keeping the computational cost relatively low (Emmanuel and Tio, 2021b).

Emmanuel's research suggests that the Inception V3 architecture is a suitable choice for creating a model for face shape prediction. Emmanuel trained different versions of the Inception architecture with the same dataset and found that the Inception V3 model performed the best. The model achieved an accuracy of 70%, which is a reasonable result for this type of task (Emmanuel and Tio, 2012). Emmanuel's research highlights the importance of selecting the right architecture when training an image classification model. The Inception V3 architecture is a good choice for face shape prediction because it is designed to handle complex and diverse images, which is important when dealing with human faces that can vary greatly in appearance. However, it is important to note that the accuracy of the model may depend on factors such as the quality and size of the training data, the complexity of the images, and the hyperparameters used in training. Therefore, it is important to carefully tune the model and optimize its architecture to achieve the best possible accuracy.

Method	Accuracy
Inception V3 [10]	84.4%
Region Similarity [1]	80%
3D face data and SVM [7]	73.68%
Active Appearance Model (AAM), segmentation, and SVM	72%
Hybrid approach VGG and SVM	70.3%
Proposed method	70%

Figure 3 - Comparing Existing System

One notable study that used the Inception architecture is the work by Mehta et al. (2021). In this study, the authors had a fine dataset for training an image classification model to predict the face shape of a person. The authors used face detection techniques to detect the faces in the images using OpenCV Haar Cascades and cropped the faces to create a dataset. The authors then trained four different models using different techniques, including Inception V2, Inception V3, SVM, and HOG. Among these techniques, the model trained using the Inception V3 architecture gave the best results. This model achieved an accuracy of around 85%, which was significantly better than the other architectures that were tested.(Mehta et al., 2021) The results of Mehta et al. (2021) demonstrate the effectiveness of the Inception V3 architecture for image classification tasks, particularly for face shape prediction. Therefore, it is important to carefully select and preprocess the training data, choose appropriate hyperparameters, and optimize the model architecture to achieve the best possible accuracy.

2.4.4 Ensemble approach for face shape prediction

Ensembling is a technique in machine learning where multiple models are combined to improve the performance of a predictive model. The basic idea behind ensembling is that by combining the predictions of several models, the ensemble can outperform any individual model. The ensemble can reduce the variance and bias of the individual models, making it more robust and accurate (Ao, Rieger and Amouzegar, 2010). So, the author can use Ensembling techniques to increase the model accuracy.

To create a high accuracy model for the image classification task, (Pasupa, Sunhem and Loo, 2019b) the author of this research used an ensemble technique that combined two different models: VGG16 and SVM. VGG16 is a popular deep convolutional neural network architecture that is often used for image classification tasks, while Support Vector Machine (SVM) is a machine learning algorithm that is commonly used for classification tasks. After creating the two models, the researcher used Multiple Kernel Learning (MKL) techniques to combine the two models and improve the accuracy of the final model. MKL is a method for combining multiple kernels (using similarity measures between data points) to improve the performance of a classification algorithm. In this case, the author used MKL to combine the outputs of the VGG16 and SVM models and create a final prediction.

Figure 4 - Performance of the System

The resulting hybrid model was able to achieve an accuracy of 70.33%, which is a significant improvement over the accuracy achieved by either of the individual models. This demonstrates the effectiveness of using a hybrid approach and multiple kernels learning to improve the accuracy of an image classification model.

(Abdullah et al., 2022) In this research, the researcher used an ensemble method with multiple machine learning techniques for face shape classification. The dataset consisted of 400 images for each of the four face shape classes. The researcher initially used the frontal face Haarcascade from the OpenCV library to detect and crop the faces in the images. The cropped face images were then resized to 256 x 256 pixels. After preprocessing the data, the researcher trained the model using the vgg16 architecture, the vgg16 model was used as a base model for feature extraction. Next, the researcher explored various machine learning techniques for classification, including Decision Tree (DT), Random Forest (RF), Gradient Boosting Machine (GBM), Extreme Gradient Boosting (XGB), and Multilayer Perceptron (MLP). Each of these techniques was trained on the extracted features from the VGG16 model. Finally, the researcher created an ensemble by combining the predictions from the individual models. The ensemble method is known to improve accuracy by leveraging the diverse strengths of different models. The final output from the ensemble model achieved an impressive accuracy of 86.5% (Baixian et al., 2018).

Ensembling techniques allow for the combination of multiple models to create a more robust and accurate prediction. By combining the predictions from different machine learning techniques, the ensemble model achieved a higher accuracy than any individual model alone. It is important to note that the success of the ensemble method can be influenced by factors such as the choice of base models, the diversity of the models, and the quality and representativeness of the dataset. Careful consideration of these factors can contribute to the overall effectiveness of the ensemble approach in face shape classification.

2.4.5 Summary and conclusion of Face shape prediction with CNN

Based on the existing works mentioned above, training a model from scratch can be a good approach for face shape prediction. However, it is important to note that achieving high accuracy often requires a large amount of data and significant training time. One challenge researcher may encounter is the availability of a sufficiently large dataset for training. To address this issue, researchers can explore alternative approaches such as utilizing pre-built architectures like VGG. Pre-trained architectures are models that have been trained on large-scale datasets, typically for general image classification tasks. These architectures have learned useful and generic features that can be beneficial for various related tasks, including face shape prediction. This approach can save time and computational resources compared to training a model from scratch. Additionally, using pre-trained architectures allows researchers to transfer the learned representations to the face shape prediction task, which can contribute to achieving higher accuracy with limited data.

There are several architectures that exist in the CNN but in here only few architectures are used in the face shape prediction domain. Most of the researchers used VGG, Inception and ResNet but there are several new architectures that exist which are not used in this domain for an example DenseNet, EfficientNet and RegNet which has been introduced in the past few years. So, if the author tried these above architectures, it would help him to increase the accuracy and it will be considered as a good contribution in this domain.

Data preprocessing plays a crucial role in training a face shape prediction model as it can significantly impact the accuracy of the model. The existing works author mentioned highlight the importance of data preprocessing and how it can lead to differences in accuracy even when using the same architecture, such as Inception V3. Data preprocessing involves several steps,

including data cleaning, normalization, resizing, and augmentation. These steps are performed to ensure that the input data is in a suitable format for the model to learn meaningful patterns and improve generalization. One of the researchers mentioned above recognized the significance of data preprocessing and applied techniques such as image augmentation to modify the dataset. Image augmentation involves applying various transformations, such as rotations, translations, scaling, and flipping, to artificially increase the size and diversity of the dataset. This helps to reduce overfitting and improve the model's ability to generalize to unseen data.

Ensembling also a good and not much used technique in this domain which led the research to achieve high accuracy. In here limited research are done in this domain using ensembling and also existing results also mainly used VGG architectures to create the ensemble methods so if author try to ensemble different architectures it may give more accuracy.

2.5 Technological Review

2.5.1 image pre-processing for face shape prediction

Image preprocessing is an essential step in the development of an image classification system, particularly in the context of face shape classification models. It involves a series of techniques and operations that are applied to raw images in order to enhance their quality, remove noise, standardize their format, and prepare them for further analysis. Effective image preprocessing is crucial for ensuring the accuracy and reliability of subsequent image classification models.

Emmanuel's research aimed to investigate the impact of image preprocessing on the accuracy of face shape classification models. In the study, the author utilized the OpenCV Haar Cascades algorithm for face detection and cropping, followed by resizing the images to a standardized size of 256 × 256 pixels. The dataset was then split into training and testing subsets, with 80% of the data allocated for training and the remaining portion reserved for testing (Emmanuel and Tio, no date).

The author has decided to use Multi-task Cascaded Convolutional Neural Networks (MTCCN) instead of OpenCV Haar Cascades for detecting the face in the images. This decision is based on the fact that MTCCN has been shown to have better accuracy in face detection, and it can also detect small faces, which is important for this project. Moreover, the author plans to crop the heads instead of the faces to avoid cropping important parts of the face, particularly in oblong and oval face shapes.

Furthermore, the author intends to resize the images to 244 x 244, which is a commonly used image size in CNN architectures. Resizing the images to this standard size can help ensure consistency across the dataset and facilitate the use of pre-trained models. Additionally, the author may apply data augmentation techniques to increase the diversity and size of the dataset, which can further improve the performance and accuracy of the subsequent face shape classification model (Sun et al., 2021).

Image augmentation is a technique used in computer vision to create new variations of existing images. It helps to address challenges like limited data and overfitting. By applying transformations like rotation, scaling, and noise addition, the author can fix overfitting issues. These variations make the model more robust and better able to handle different scenarios. Previous research suggests different ways for image augmentation but here the author going to use the trial-and-error method to select the best augmentation method (Ao, Rieger and Amouzegar, 2010).

2.5.2 CNN based face shape prediction techniques.

The literature review conducted in the previous chapters strongly supports the use of Convolutional Neural Network (CNN) algorithms for creating a face shape prediction model os a best choice in author's research. Several studies consistently demonstrate that CNNs outperform alternative approaches in terms of accuracy, precision, recall, and F1 score. CNNs are particularly effective in capturing intricate facial details and spatial relationships, surpassing traditional computer vision techniques, support vector machines (SVMs), and decision trees. The widespread consensus among researchers confirms that CNN algorithms are the most suitable choice for achieving accurate face shape prediction (Bell, 2022).

The previous research on face shape prediction has explored various architectures, including creating models from scratch (Young et al., 2019; Baixian et al., 2018), Inception V3 (Emmanuel and Tio, 2021a), VGG (Vittal, Gowda, and Sankapal, 2022), ResNet (Ratna Wati, Dinar Mutiara, and Puji Widodo, 2021), and others. Some of these models have achieved good accuracy, exceeding 75%. Additionally, ensemble models, which combine multiple models, have shown even higher accuracy, surpassing 80%. Based on these results, the author decided to employ an ensemble approach for face shape prediction (Jason, 2022).

The choice to use ensemble models was influenced by several factors. Firstly, there have been only a few previous studies that have explored ensemble methods for face shape prediction. Furthermore, these studies focused exclusively on using VGG architectures for ensembling models. However, there are several other CNN architectures available that have not been utilized in creating face shape prediction models (Pasupa, Sunhem and Loo, 2019b).

2.5.3 RegNet architecture for face shape prediction

RegNet, short for "Regularized Convolutional Neural Network," is a powerful architecture designed for deep learning tasks such as image classification. Introduced by Radosavovic et al. in 2020, The significance of RegNet lies in its ability to provide an effective and efficient solution for developing convolutional neural networks. With its architectural regularization, RegNet achieves superior performance while reducing computational costs, making it an attractive choice for resource-constrained environments. This architecture helps the author to build models that strike a balance between accuracy and efficiency, opening possibilities for more practical and real-world applications (Xu et al., 2022).

RegNet's characteristics make it particularly useful for face shape prediction tasks. Face shape prediction involves capturing and analyzing complex facial features to determine the shape of an individual's face. Given the intricate nature of facial structures, a powerful and efficient model is required to accurately predict face shapes. RegNet's architectural regularizes offer several advantages in the context of face shape prediction. First, they enable efficient scaling of the model, allowing it to handle larger datasets and capture more nuanced facial details. This scalability is crucial for achieving higher accuracy in face shape prediction, as it helps the model learn and understand diverse facial variations (Xu et al., 2022).

Additionally, RegNet's computational efficiency is beneficial when working with limited computational resources, such as edge devices or mobile platforms. It allows face shape prediction models based on RegNet to be deployed in real-time applications with low latency and reduced computational costs. There are several versions in RegNet like RegNetY160, RegNetX160 and many more for selection best version author going to use the trial-and-error method.

2.5.4 EfficientNet architecture for face shape prediction

EfficientNet is a highly influential convolutional neural network architecture introduced by Tan and Le in 2019. It revolutionized the field of deep learning by addressing the challenge of model scaling, which involves balancing model size, computational efficiency, and accuracy. EfficientNet achieves this balance by employing a novel compound scaling method that uniformly scales all dimensions of the network, including depth, width, and resolution. This approach leads to improved performance and efficiency, making EfficientNet a significant breakthrough in deep learning (Liu et al., 2020).

EfficientNet's architectural design makes it particularly useful for face shape prediction, a task that requires accurate analysis of facial features. Face shape prediction involves understanding and classifying the unique facial structures of individuals, which can vary significantly. EfficientNet excels in this area due to its optimized architecture. EfficientNet's compound scaling approach allows it to effectively capture intricate facial details, leading to more accurate face shape predictions. The scaling method ensures that the model can handle diverse facial variations and learn from large datasets. By efficiently scaling the model's depth, width, and resolution, EfficientNet achieves a balance between accuracy and computational efficiency, making it well-suited for face shape prediction tasks(Liu et al., 2020).

2.5.5 common model training technics in CNN

Early stopping is a technique used in Convolutional Neural Networks (CNNs) to prevent overfitting and optimize model performance. It involves monitoring the model's performance during training and stopping the training process before it reaches the point of overfitting. In here author going to use the EarlyStopping method from Keras library when the accuracy is not improved for 5 epochs the system will stop the training process.

When training a Convolutional Neural Network (CNN), it is essential to **Save the best model** during the training process. This can be achieved using the ModelCheckpoint technique, which automatically saves the model with the best performance based on a specified metric. The ModelCheckpoint callback in popular deep learning frameworks like Keras allows you to monitor a specific metric, such as validation accuracy or loss, and save the model weights whenever the performance on that metric improves. In here the author going to use the accuracy of the model.

2.5.6 Model Evaluation Techniques

In the image classification domain, various quantitative evaluation techniques are employed to measure the quality and performance of a model. These techniques provide objective measures that assess the accuracy and effectiveness of the image classification results. Here are some common quantitative evaluation techniques used in image classification:

- **Accuracy: -** It is the most basic and widely used metric for evaluating image classification models. Accuracy measures the percentage of correctly classified images out of the total number of images in the dataset (Liu et al., 2020).
- **Precision, Recall, and F1 Score: -** These metrics are commonly used in binary or multi-class classification tasks. Precision measures the ratio of true positive predictions to the total number of positive predictions, while recall calculates the ratio of true positive predictions to the total number of actual positive samples. F1 score is the harmonic mean of precision and recall, providing a balanced measure of both metrics (Jason, 2022).
- **Confusion Matrix: -** A confusion matrix provides a tabular representation of the predicted class labels against the actual class labels. It helps in assessing the performance of the model by showing the counts of true positives, true negatives, false positives, and false negatives. From the confusion matrix, various evaluation metrics such as precision, recall, and accuracy can be calculated (Baixian et al., 2018).

2.6 Chapter Summary

This chapter presents a literature review of relevant studies and research on hair and beard style recommendation system. The author conducted a comprehensive search of academic databases, journals, books, and other publications to identify and analyze a range of scholarly works. The author synthesized the key findings and themes from literature and highlighted the most significant and relevant works in the field. Based on this analysis, the author identified gaps and limitations in the existing literature and suggested potential avenues for future research.

CHAPTER 3: METHODOLOGIES

3.1 Chapter Summary

This chapter provides a brief explanation about the methodologies which used in the project. This section also presents a comprehensive summary of the research, project, and development methodologies that were chosen to advance the study. Additionally, the potential risks associated with the research were identified and strategies for mitigating these risks were proposed.

3.2 Research Methodology

The quality of a project is determined by three main factors: cost, time, and scope (Smith, 2012). Thus, if we need to provide good research, our research methodologies must also be carefully selected with the above factors in mind. The research methodologies were selected from the predefined Research Onion Model by Saunders et al. (2007). The table in here provides readers with a brief overview of the research methodologies **Appendix 10 - Research Methodology**

3.3 Development Methodology

3.3.1 Life cycle model

Among all the development methodologies, the agile methodology is chosen because it minimizes the risk when adding new functions and allows for iterative release. They also allow users to realize software benefits earlier, with frequent incremental improvements.

3.3.2 Requirement elicitation methodology

To find the requirements for our system, we can use various ways, such as surveys, interviews with domain experts, brainstorming sessions, literature review, experiment and observation. In our case, the author decided to do a survey with the group of users, and according to that survey, we can point out the requirements. Also, the author plans to use literature review and observation to collect requirements.

3.3.3 Design methodology

Among the possible candidates of object-oriented analysis and design and structured systems analysis and design methods Structured Systems Analysis and Design Method (SSADM) was chosen because it allows effective management of software complexity by virtue of modularity (Kothari, 2019). Also, it makes the code simple because we can create reusable components.

3.3.4 Evaluation methodology

Evaluation is the most important part of the research. It helps to deliver a quality product to the user. authors especially evaluate the accuracy and speed of the system. To evaluate the system among the various methods author planning to do Benchmarking methodology.

3.4 Project Management Methodology

Among many candidates' agile prince 2 was chosen for this research because it maintains product quality, recursive planning, and flexible delivery. Also, if you take the Agile Prince 2 methodology, it helps to achieve the requirements on time, so it will increase flexibility. It also helps in cooperating with stakeholders, increasing stakeholder confidence, and having the ability to change tools and technology if the requirements constantly keep changing (Snyder, 2019).

3.4.1 Deliverable

Deliverable can be found in **Appendix 12 - Deliverable**

3.4.2 Project Plan (Gantt Chart)

Gantt chart can be found in **Appendix 02 - Gantt Chart**

3.4.3 Risk management

When doing research, it's always connected with risks such as technical, theoretical, or any other unpredictable issues. A well-managed risk management plan is essential to finishing a project with the expected outcome the risk management can be found in **Appendix 11 - Risk management.**

3.5 Chapter Summary

This chapter examined the methodology for research, development, and project management with all of the requirements, the justification for choosing each requirement, and any risks that could be anticipated with a mitigation strategy.

CHAPTER 4: SOFTWARE REQUIREMENTS SPECIFICATION

4.1 Chapter Overview

This chapter focuses on identifying possible stakeholders of the project by taking a look at all possible points of interaction with the system with the use of a rich picture diagram, gathering their perceptions to analyze and come up with possible expected use cases, functional and non-functional requirements of the prototype.

4.2 Rich Pitcher Diagram

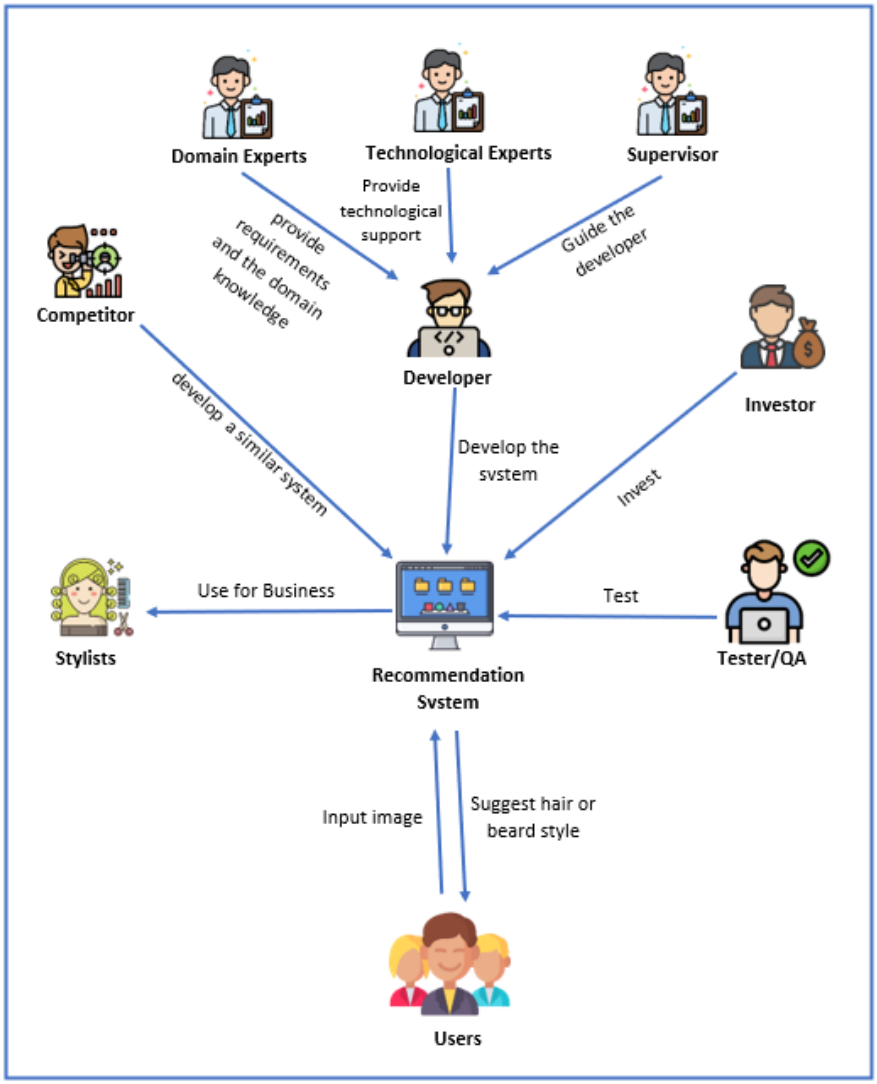


Figure 5 - Rich Pitcher Diagram

The rich picture diagram above depicts an overall view of the projects. It briefly showed how society interacts with the hair and beard style recommendation system. This will assist readers in understanding how the system will interact with society once it is implemented in the real world. This rich picture diagram also shows the real-world characters who can interact with the system and how they interact.

4.3 Stakeholder Analysis

The Stakeholder Onion Model illustrates recognized stakeholders who are associated with the system, along with an explanation of each stakeholder's involvement in the system, in Stakeholder Viewpoints

4.3.1 Stakeholder Onion Model

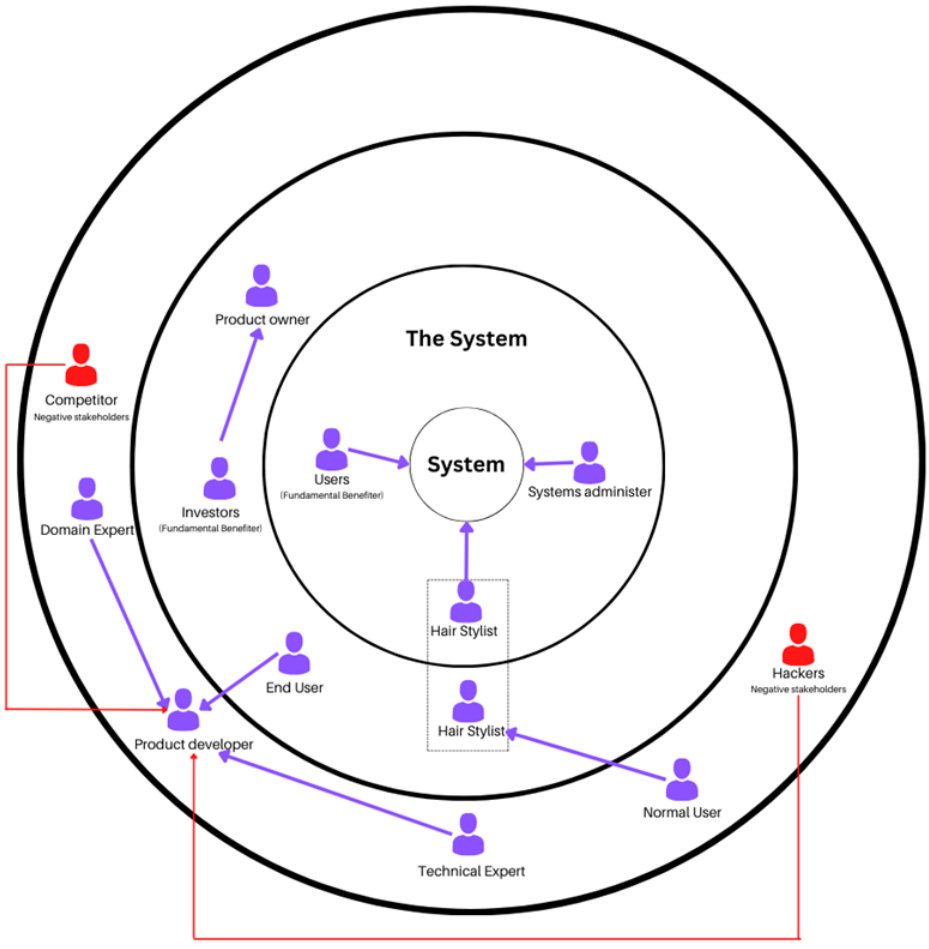


Figure 6 - Stakeholder Onion Model

4.3.2 Stake holder Viewpoints

Below diagram give a brief idea about the stakeholders and explain their role and benefits with the table format

Table 5 - Stake holder Viewpoints

Stakeholder	Role	Benefits/ Role Description
Users	Functional Beneficiary	Using the system to find suitable hairstyle for them
Hair stylist	Functional Beneficiary	using the system to suggest hair and beard styles for their customers
Product Owner	Functional Beneficiary	Owner of the system and who have all the rights of the system
Investors	Financial Beneficiary	Persons who invest in the research, development, evaluation, and marketing of the product receive the profit from the product.
End user	Functional Beneficiary	Using the system to find suitable hairstyle for them
Competitor	Negative Stakeholder	Create the system which have the same features of the system
Domain Expert	Expert	Provide opinion and suggestions about the product domain
Product developer	Financial Beneficiary	Developed the system
Technical expert	Expert	Provide opinion and suggestions about the technical side of the product
Hackers	Negative Stakeholder	intends to disrupt and interrupt the system and its data.

4.4 Selection of Requirement Elicitation Methodologies

In order to gather requirements for the development of the research project, there were multiple requirement elicitation methodologies that were followed. literature review, interviews, survey & prototyping were the methodologies chosen for this purpose. The reasons to choose the specified requirement elicitation methodologies have been discussed below.

4.4.1 Literature review

The literature review is the most important part of the final-year project development. It will help the author get a basic understanding of the project and also help him develop it perfectly with the help of previous research. So, the author has done some brief research on the hair and beard style recommendation system. It aided the author in identifying the research gap and providing information about existing technologies and the methods he needed to use to complete the project successfully.

4.4.2 Survey

One of the important things is gathering requirements from the end user. The proposed system's end users are ordinary people who want to make their own beards or hairstyles, so the number of end users is large, so it's better to use the questioner to cover them all. Compared to an interview, the survey helps cover a larger number of end users. It will also allow the author to clarify if the proposed solution would be helpful to the intended users.

4.4.3 Brainstorming

Brainstorming is another effective method for identifying requirements; it can be done alone or with others at various stages of the project, and it aids in the identification of new requirements. The brainstorming sessions are held with supervisors, domain experts, and technical experts, and the author receives numerous suggestions that help improve the project.

4.4.4 Prototyping

Since the project was chosen to follow the Agile Software Development Lifecycle, prototyping would allow the author to recursively try out various alternative implementations to identify any areas for improvement while testing and evaluating the prototype.

4.5 Discussion of Findings

2.5.1 Survey

Discussion of the survey can be found in **Appendix 09 - Survey**

4.5.2 Prototyping

The author embarked on a prototyping journey and encountered various requirements and challenges along the way. One of the biggest obstacles was the lack of proper datasets for training and testing the models. To address this issue, the author took a proactive approach by

merging three datasets to create a new, comprehensive dataset. Despite these efforts, the developer still faced validation problems, such as incorrect data uploads from users or instances where the model was unable to distinguish between male and female faces. These validation issues highlighted the need for careful and thorough testing to ensure the model's accuracy and reliability. Throughout the prototyping phase, the author discovered numerous requirements, each posing its own unique set of challenges. Nevertheless, the author remained persistent in overcoming these obstacles, ultimately achieving a successful prototype.

4.5.3 Literature Review

Table 6 - Summary of Literature Review

Finding	**Citation**
Based on the review, the author has determined that a hair and beard style recommender system would offer numerous benefits to a large and diverse group of users. Furthermore, the review highlighted the shortcomings of previous face identification models, specifically with regards to their accuracy levels. This information has informed the author's approach to the design and development of the system, ensuring that it not only addresses the identified needs but also addresses the limitations of previous models.	(Science Council of Asia. Conference (16th: 2016 : Colombo, National Science Foundation (Sri Lanka) and National Academy of Sciences - Sri Lanka, no date)
The accuracy of the previous models is low because they used simple datasets and old CNN architecture.	(Shah and Hao, 2016)
Ensemble approach can increase the accuracy and performance of the system	(Alzahrani, Al-Nuaimy and Al-Bander, 2021)
Using the combination of face shape prediction and gender identification model can give more personalized recommendation	(Mehta et al., 2022)

4.5.4 Brain Storming

The author conducted brainstorming during several phases of the project's development. The author conducted both group and self-brainstorming sessions to find out the requirements for this project, which helped the author determine the design of the project. Discussions with the supervisor help to validate the project idea and the requirements of the project. During these

brainstorming sessions, the author discovered that combining the beard and hair style recommendation systems allowed him to stratify the majority of the users.

4.6 Summary of Finding

The requirements table displays a summary of the requirements gathered during this project's literature review, survey, prototyping, and brainstorming.

Table 7 - Summary of Finding

Finding	literatu	Surv	Brainsto	Prototot
The proposed system assists both ordinary people and stylists in selecting the ideal hair and beard style.	X	X	X	
The system is primarily aimed for both men and women over the age of 20.	X	X		
Architecture should be modular as to support extension of code and features.			X	X
Everyone expects the system to be simple and easy to use.		X	X	
All error scenarios require proper validation methods in the system.			X	X
Simple and easy to use UI		X		X
A sufficient set of well-cleaned and pre-processed dataset would be essential for the system's performance.	X			X
To improve system accuracy, hybrid techniques should be used.	X			

When the user attempts to enter female face images, for getting beard recommendation the system should notify him.		X	X	

4.7 Context Diagram

The context diagram is used to show the boundaries of the system and also it helps to understand the internal and external interferences. The following diagram shows the context of this project.

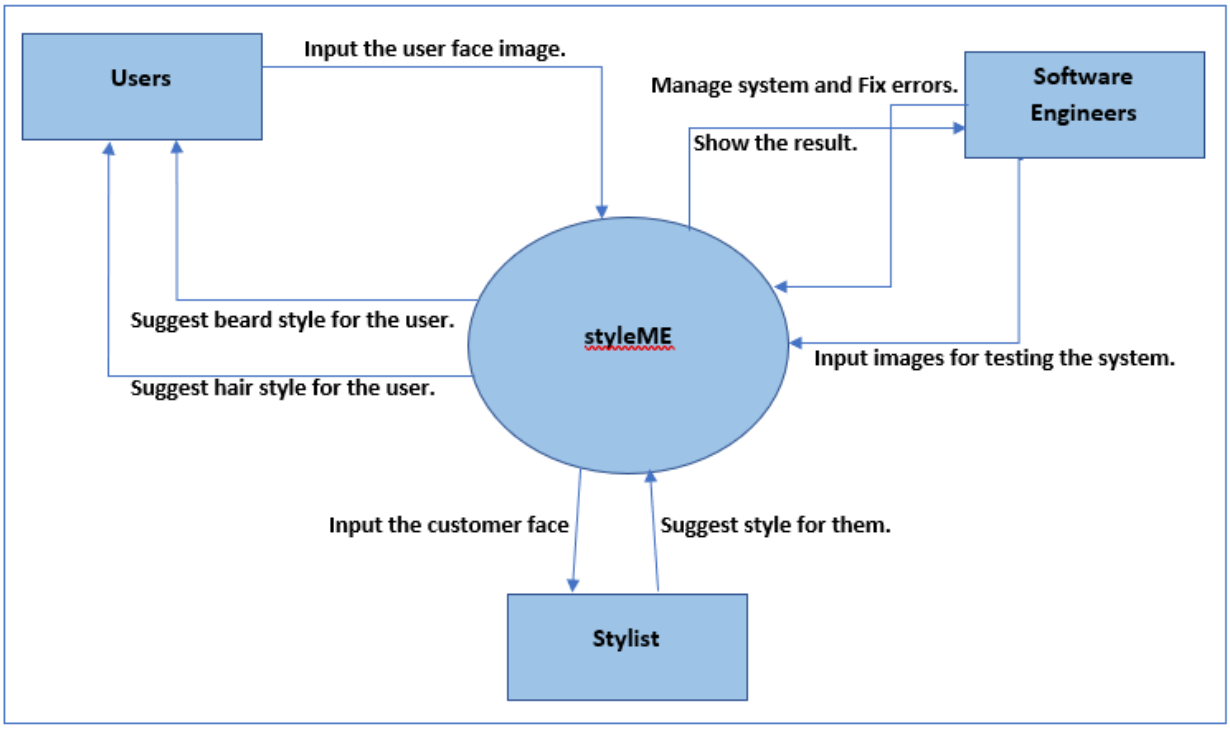


Figure 7 - Context Diagram

The context diagram shows how the StyleMe web application is used by different types of users. The general users interact with the system to access the styling services offered by the application, while the software engineers develop and maintain the application. Additionally, the stylists provide styling services through the application. All these stakeholders are connected to the application through a network, and the application interacts with an external database to store user information and styling data.

4.8 Use case diagram

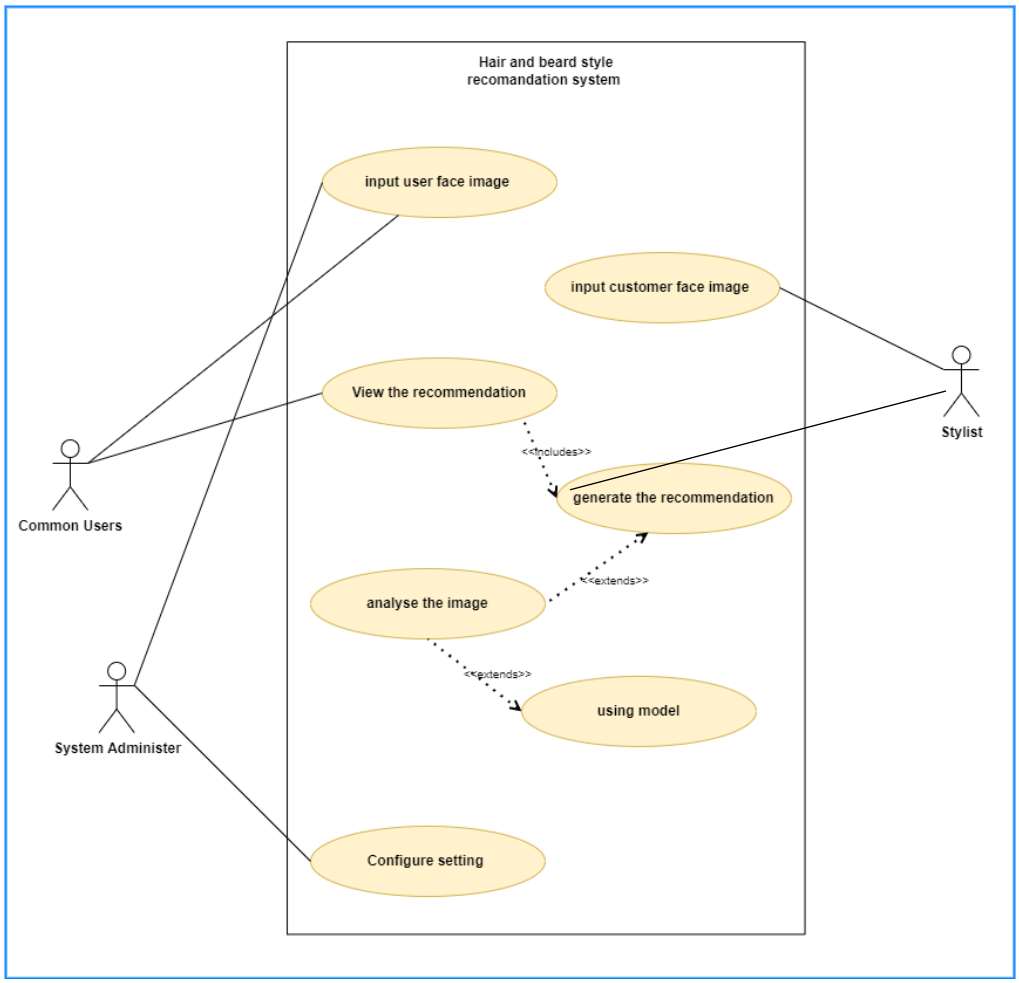


Figure 8 - Use case diagram

4.9 Use case descriptions

Table 8 - Use case descriptions

Use Case	View recommendation
Description	Display the suitable hair or beard style for the user
Primary Actor	Common user, Stylist
Supporting Actors (if any)	None
Stakeholder and	System admin

interests	
Pre-Conditions	Recommendation models need to be trained and working correctly and user need to add the image
Post condition	Generated recommendation needs to be presented to the user successfully
Trigger	When a user wants to generate a suitable hair or beard style for himself or others based on the shape of their face,
Main Flow	• Users open the system and add a face image. • Generate the recommendation

4.10 Functional Requirements

In this section, the author will provide details about the functional requirements of the project at four priority levels. To select the priority levels, the author used the MoSCoW technique, and those are provided below.

Table 9 - Priority Level Description

Priority Level	**Description**
Must have (M)	This level's requirement is a prototype's core functional requirement, and it must be implemented.
Should have (S)	Important requirements aren't necessary for the expected prototype to work, but they do add a lot of value.
Could have (C)	desirable requirements that are optional and aren't deemed essential critical to the project's scope.
Will not have (W)	The requirements that the system may not have and that are not considered a top priority at this time.

The founded functional requirements are provided in here **Appendix 13 - Functional Requirements**

4.11 Non-Functional requirements

Table 10 - Non-Functional requirements

NFR ID	Requirement	Description	Priority Level
1	Accuracy	The recommendation algorithms and the model need to have high accuracy.	Important
2	Performance	The system needs to generate the suggestion within a short period of time.	Important
3	Maintainability	The developer needs to follow the coding standards and best practices when he develops the system.	Desirable
4	Usability	The system needs to throw proper error messages when errors occur.	Desirable
5	Usability	The system must have a simple GUI that all users can understand, as well as a help page that explains the system to the user.	Important
6	Security	The application should prevent any attackers from manipulating results and extracting user inputs. also need to have proper login methods.	Desirable
7	Compatibility	The system needs to support all types of platforms.	Desirable

4.12 Chapter Summary

This chapter provided information about the project's stockholders as well as the project's requirements. The author provided a rich picture diagram that shows how society interacts with the system in this chapter, and the author also used Saunders' onion model to represent the stakeholders who are connected to the system. This chapter also provided a brief explanation of the requirements gathering techniques, and with the help of those methods, the author listed down the functional and nonfunctional requirements.

CHAPTER 5: SOCIAL, LEGAL, ETHICAL AND PROFESSIONAL ISSUES

5.1 Chapter Overview

This chapter is going to give a clear overview of social, legal, ethical and professional issues which can occur during the research period and also going to provide steps taken to mitigate those issues.

5.2 SLEP Issues and Mitigation

Table 11 - SLEP Issues and Mitigation

Social	Legal
• The questionnaire didn't get the personal details of respondents, only it asked questions which related to the research. • The system didn't get or save the personal details of the users.	• The usage of any programming languages, tools, or frameworks was subject to an open-source license. • The system was created in a way that ensures no personal information is needed to get the desired results.
Ethical	**Professional**
• The project and their involvement in it were explained to the participants who answered the questionnaires. • The author didn't used any plagiarism content in the thesis also and if author used any reference in the thesis also sited perfectly. • The academic behavior and policies of the University of Westminster are followed with regard to the proper citation and reference of research papers and conference connected to the study.	• The author followed the best practices and the best methods during the development of this project. • Throughout the whole research process, proper standards for research were followed.

5.3 Chapter Summary

Possible social, legal, ethical, and professional difficulties were identified in this chapter with study mitigation techniques.

CHAPTER 6: DESIGN

6.1 Chapter Overview

This chapter covers the diagrams and designs related to the implementation of the project. This chapter shows the high-level design diagram, the low-level design diagram, the class diagram, the sequence diagram, the UI wireframes, and also gives a brief understanding of the design goals. The discussions also document the various alternatives considered and the rationale for deciding on the final designs.

6.2 Design Goals

Design goals need to be defined before starting the design part of the projects, and those are provided in the below table.

Table 12 - Design Goals Description

Design Goal	**Description**
Performance	The image processing model takes a lot of time and computational power to generate a suitable recommendation. So, the design must effectively improve performance. Also, the design needs to improve the overall performance of the system by 2 or 3 times.
Correctness	The project's main goal is to create an image processing model that can be used to suggest hair and beard styles. So, the correctness of the suggestion is more important; if it didn't suggest it correctly, then there is no use of the model. so, the developer needs to put effort into designing a perfect model also developers need to ensure it will not suggest in correctly.
Usability	The main purpose of the project is to suggest hair and beard styles for all types of people, so the users of the project are from various scales, so the usability of the system must be important, so when the developer does designs, he needs to take care of it. Additionally, the system's graphical user interface (GUI) must be simple and easy to use.
Scalability	The system's basic architecture and main backbone should function well. in times when large datasets, complex algorithms are provided. Additionally, it should also be able to add new algorithms and preprocessing steps without much

	tweaks
Adaptability	Since the utilized recommendation models may have to be altered based on the available data and user requirements in the future, these models should be able to be easily swapped out for new ones while ensuring that the system won't break in the process of upgrading, with minimum changes.

6.3 High level Design

6.3.1 Tiered Architecture

The high-level architecture of the system is based on the selected layered architecture type. The tiered architecture diagram is a brief explanation of the three major tiers of the system. The author followed the three-tiered architecture, which is the presentation tier, logic tier, and data tier.

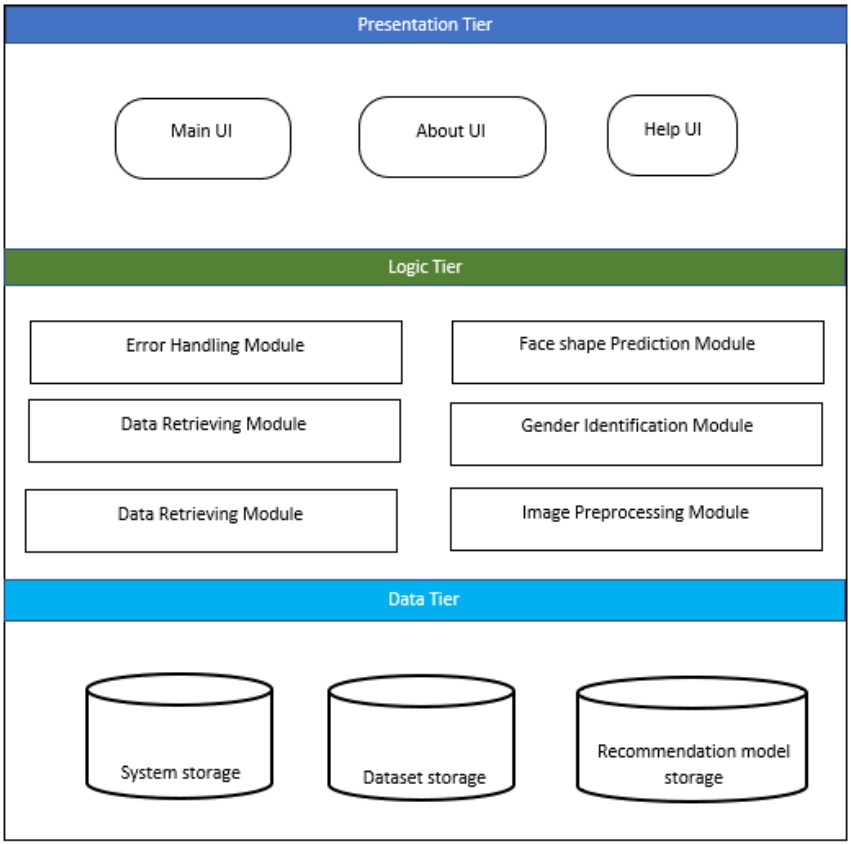


Figure 9 - Tiered Architecture

These three layers logically separate the functionalities of the system. The presentation layer interacts with the user. The logic layer acts as the middle layer, where all the main functionalities are defined. The data layer contains search space, model storage, searched architecture storage, and auditable configurations. A summary of each layer is presented below.

Presentation Tier

- User Input Wizard - This UI allows the user to upload a picture of their face.
- About UI – this is a static UI provides some important aspect about the system.
- Help UI – this is a static UI that provides basic system information and indicates how the user should use it.

Logic Tier

- Error Handling Module – the logic for detecting and handling errors or unexpected results in the melody generation process and providing appropriate feedback to users or developers.
- Data Retrieving Module – this module is responsible for retrieving data from the storage and sending to the front end.
- Face shape Prediction Module – the logic for getting the face image and using face shape prediction module to predict the face shape.
- Gender Identification Module - the logic for getting the face image and using gender identification module to predict the gender.
- Image Pre-processing Module - the logic for getting the face image and cropping the head from that image.

Data Tier

The data tier displays information regarding the storage system, and it is the third tier in the three-tiered architecture. In this tier, the author has identified three key components those are:-

- Dataset storage - The data that is required to train the recommender models which will be collected from the internet will be stored in this storage.
- Recommendation model storage - This component stores the recommendation models like face shape prediction system and gender identification models, which are used by the system to predict outcomes.

- System Storage - this storage stores the code and other things which are related to the system.

6.4 Low Level Design

6.4.1 Choice of design paradigm

The main considerations for the design paradigm are the Structured Systems Analysis and Design Method (SSADM) and OOAD (Object-Oriented Analysis and Design). Structured Systems Analysis and Design Method (SSADM) is approaching the project in a systematic way to analyze and design it. It provides a step-by-step process for the development of the system, from the pre-production stage to the end implementation of the system. Object-Oriented Analysis and Design (OOAD) is approaching the project in a software engineering way; it also models a system as a collection of objects, each representing a class, and the class describes the attributes and behavior of the object. OOAD is chosen for project development because the project's requirements will change dynamically day by day, and OOAD is flexible for that. Additionally, the project is based on a real-world problem, so the designer must handle real-world objects in the design so that OOAD can use the real-world objects. OOAD also improves reusability, flexibility, collaboration, and maintainability.

6.4.2 Data Flow Diagram

The tiered architecture of the system has been analyzed and the components within it have been meticulously chosen to be implemented as standalone modules. This approach not only promotes a more organized structure, but also allows for greater flexibility and easier maintenance in the future. A diagram has been created to visualize the interconnections between these components and to outline the flow of data throughout the system. With this diagram, it becomes much easier to comprehend the overall design and to identify areas for improvement if needed. It can be found in - **Appendix 19 - Data Flow Diagram**

6.4.3 UI Wireframes

The wireframes can be found in - **Appendix 03 - UI Wireframes**

6.4.4 Sequence Diagram

The two main flows related to this project are the creation of an image classification model for face shape prediction and the flow of the hair and beard style recommendation system. As a

result, the author decided to produce two sequence diagrams to represent these two flows. First, the flow of creating the image processing model is considered. The developer is required to locate an appropriate dataset, preprocess the selected dataset, load the configurations related to the project, train the model, and evaluate it. These steps are clearly described in the accompanying diagram.

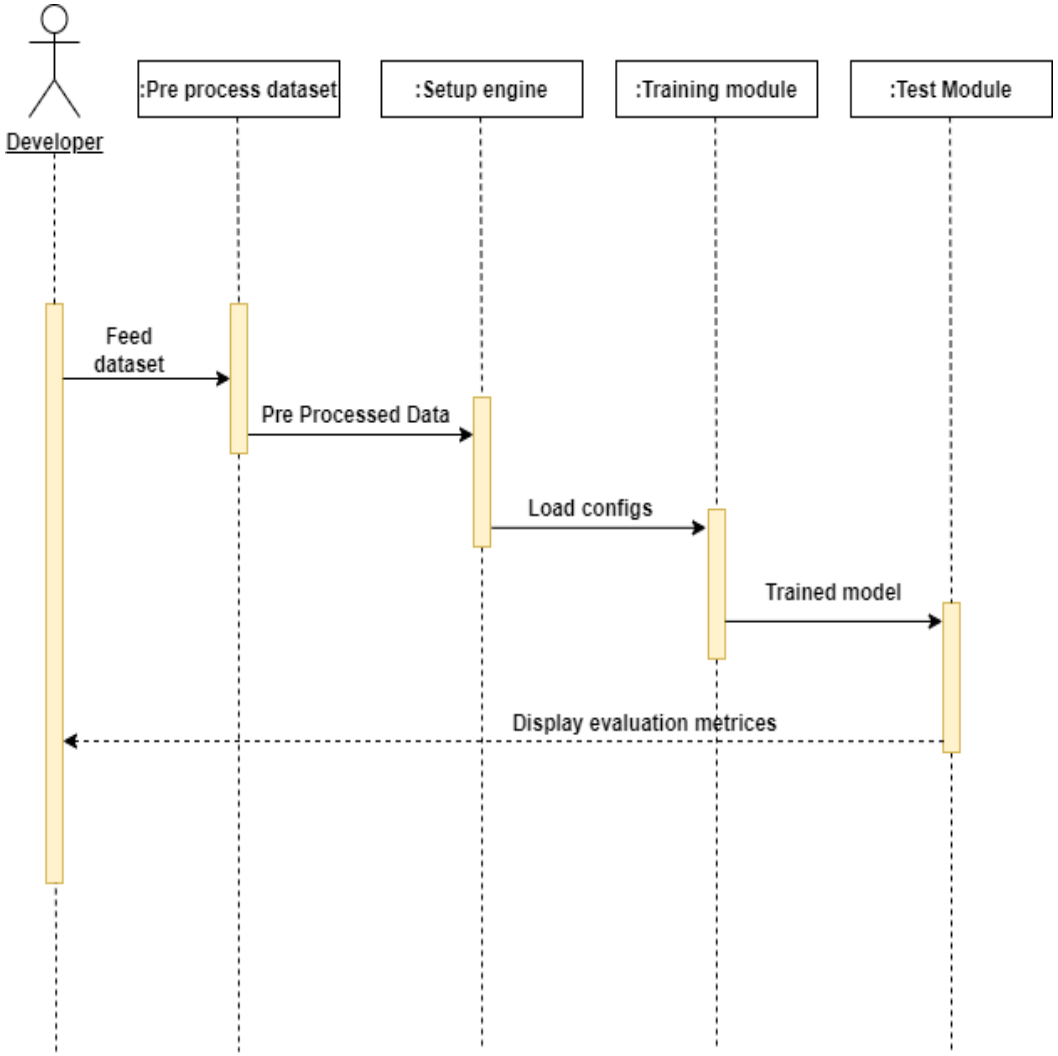


Figure 10 - Sequence Diagram of Model Creation

The primary system flow requires the user to upload a picture of their face to the web application. When you upload something, the system sends it to the backend. The image is processed and checked for errors in the backend. If an error is found, the backend sends an error message to the front end, where it displays the error to the user. If no errors are found, the

backend loads the image into the model, predicts the face shape, and suggests a suitable hair or beard style to the user based on that face shape.

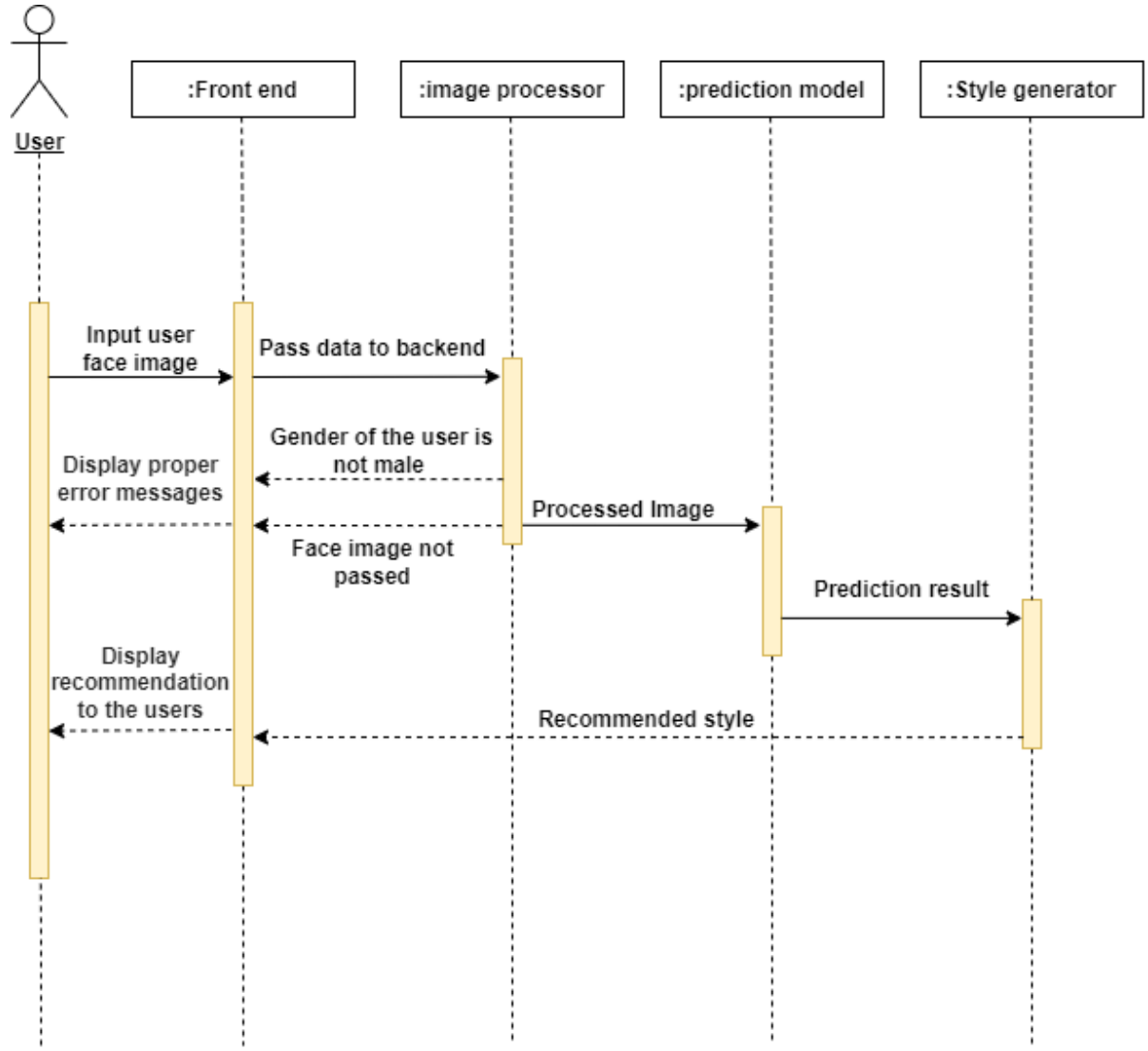


Figure 11 - Sequence Diagram of the System

6.4.5 System Process Flow Chart

The accompanying flowchart outlines the process of the algorithm and the operation of its decision-making structures. Since the system will primarily follow a procedural approach, this flowchart covers the majority of the system. The object-oriented aspect of the system is explained in further detail in subsequent sections. This flow chart clearly explains the basic flow of the system from start to end.

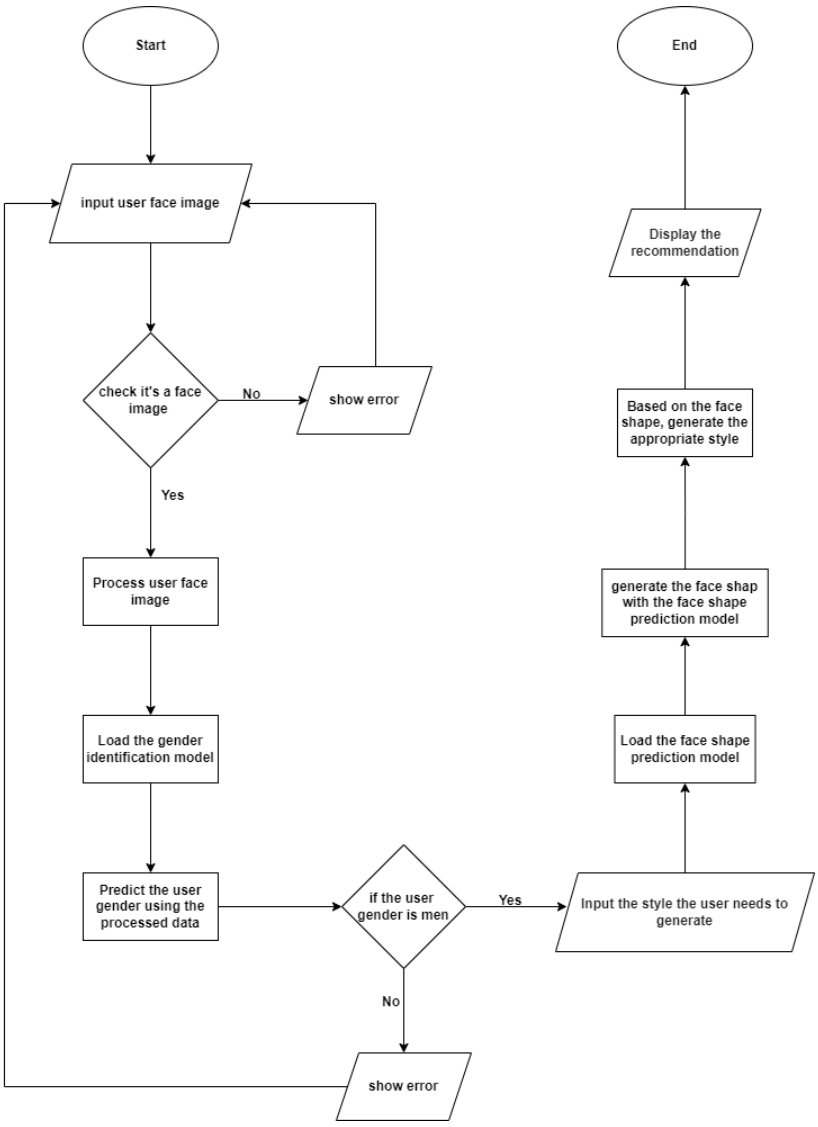


Figure 12 - System Process Flow Chart

6.5 Chapter Summary

This chapter serves as an overview of the design aspect of the project. It not only provides a high-level understanding of the system through clear and concise diagrams, but also delves into the specifics through low-level diagrams. The design goals have been meticulously outlined and explained in this chapter, offering a clear understanding of what the project intends to achieve. Furthermore, the chapter includes basic wireframes, offering a visual representation of the system's interface and how it intends to function. Overall, this chapter serves as a comprehensive guide to the design of the project, making it easier for readers to grasp the concept and approach.

CHAPTER 7: IMPLEMENTATION

7.1 Chapter Overview

This chapter is going to cover the details related to the implementation of the proposed system. It's going to provide a brief introduction about how the designs are going to be transformed into code. This chapter provides complete details on the technologies selected for the front end, back end, development tools, libraries, APIs, and datasets.

7.2 Technology Selection

4.2.1 Technology Stack

The technology stack was decided based on different layers of the system. Decided technologies as presented below.

Figure 13 - Technology Stack

7.2.2 Data set selection

Data is a very important part of the data science project, and based on this project, the system needs to find the face shape of the user. So, the author should first try to find the dataset for the project, and based on this project, the system needs to find the face shape of the user. So, the author should first try to find the dataset. For that, the author set some basic requirements for selecting the datasets.

1. A dataset must have capable images for training an image processing model.
2. Need to have enough and good images for training testing and validation.
3. Easy to access.
4. It should be identified as a complex dataset in the segmentation domain. Then, pixel-wise multi-class classification could be performed.
5. The dataset needs to mainly focus on the authors' domain.

Based on the above requirements, the author selected the **Face Shape Dataset** from the Kaggle website. This dataset contains 2000 images in 5 categories: heart, oblong, oval, round, and square, and these categories indicate the face shapes of men.

7.2.3 Development Frameworks

Python was selected by the author as the main programming language for this project. If we consider Python, Django and Flask to be the most well-known and efficient frameworks, the following table will examine these two frameworks.

Table 13 - Development Frameworks Comparison

Framework	Advantage	Disadvantages
Flask	• Lightweight and minimalistic • Flask provides a lot of flexibility. • Fast and simple. • Has excellent documentation • Supports multiple databases. • Scalable and large developer community.	• Flask is a minimalistic framework and does not come with many built-in features. • No strict structure. • No built-in security features.
Django	• Can handle complex projects. • rich feature set.	• Complex deployment. • Slow performance.

	• Strong security	• Complex structure. • Large memory footprint. • Inflexible.

The author weighed the benefits and drawbacks of both frameworks and chose flask for its simplicity, speed, and lightness.

7.2.4 Programming Languages

For the image processing model, the author considers various aspects and analyzes the advantages and disadvantages of every possible programming language, and based on that research, Python is selected for the development of this project. Some of the author's main reasons for choosing Python as the primary programming language Python has a large number of libraries for image processing, including OpenCV, Pillow, Scikit-Image, and more, making it easy to perform tasks such as image filtering, thresholding, and edge detection. Python is a good choice for developing the project because it is simple to use, ideal for prototyping, has a large developer community, and it can be easily integrated with other tools and libraries such as image processing frameworks and web frameworks.

Among the various front-end languages considered, JavaScript was chosen for the project's development because it is the most widely used programming language for web development and is supported by all major browsers, making it a good choice for creating dynamic and interactive web applications as well as single-page applications, dynamic, interactive and rich libraries and frameworks.

7.2.5 Libraries

Several libraries are required for the development of image processing models and the system. The below table is going to showcase those libraries.

Table 14 - Libraries Selection

Library	**Justification**
OpenCV	OpenCV is an open-source computer vision and image processing software library. It mainly deals with images, and it has a lot of in-built functions that the author can use for his development, for example, object detection, face recognition, and the feature extraction function, which can help the

	author in his development. Also, openCV has a big active community and is fast, simple, and has cross-platform compatibility.
Pillow	A pillow is a fork of the Python Imaging Library (PIL). It supports opening, manipulating, and saving many different images file formats. It can be used for basic image manipulation tasks such as resizing, cropping, and color correction.
Keras	Keras is a great choice for your deep learning project due to its ease of use, flexibility, large community, pre-built layers and models, compatibility, and performance. It's simple and user-friendly interface allows for easy prototyping and experimentation with different models and architectures. Keras is built on top of lower-level deep learning frameworks, making it a flexible and powerful tool for building a wide range of deep learning models. The large and active community of users and contributors means there are plenty of resources, tutorials, and examples available to support your project. Keras also supports multiple backends and GPU computation, allowing for efficient and fast model training and evaluation. Overall, Keras provides a powerful and easy-to-use solution for deep learning projects.

7.2.6 IDE

Table 15 - IDE Selection

Library	**Justification**
Google Colab	Google Colab is a free cloud-based software that is primarily used to create image processing models. It is simple to use, supports a large library, is integrated with Google Drive, and has access to GPUs and TPUs.
VSCode	Most developers use Visual Studio Code, a free and open-source IDE. It has many benefits, such as being simple to use and supporting multiple programming languages, frameworks, and libraries.

7.2.7 Summary of Technology Selection

Table 16 - Summary of Technology Selection

Components	**Selected technologies**

Programming Languages	Python, JavaScript
Development Framework	Flask
Libraries	OpenCV, Keras, Scikit-Image, Bootstrap
IDE	VSCode, Google Colab
Version Control	Git, Git Hub

7.3 Implementation of the Core Functionality

In this subheading, the author is going to provide the basic codes that are used for creating the hair and beard style system. As a result, the author will provide code snippets for the main functionalities as well as descriptions for those code snippets.

Data pre-processing

Before training the model, it's important to preprocess the available dataset to ensure that the model receives high-quality input data. Preprocessing involves various steps such as resizing the images to a standard size, normalizing the pixel values, and augmenting the dataset with additional images. By performing these steps, we can improve the accuracy of the model and make it more robust to variations in the input data.

```
import os

# Specify the directory containing the image files
image_dir = "/content/drive/MyDrive/DataSet/FaceShapeDataset/testing_set/Square"

# Get a list of all files in the directory
files = os.listdir(image_dir)

# Iterate through the files and check their existence and size
for file in files:
    file_path = os.path.join(image_dir, file)
    if os.path.exists(file_path):
        file_size = os.path.getsize(file_path)
        if file_size == 0:
            print(f"Warning: Empty file found: {file_path}")
    else:
        print(f"File not found: {file_path}")
```

Figure 14 - Data pre-processing Code

This code snippet is used to check the existence and size of image files in a specified directory. It starts by importing the "os" module and then defining the directory path of the image files. The

"os.listdir()" function is then used to get a list of all the files in the directory. A for loop is then used to iterate through the files in the list. For each file, the full file path is obtained using the "os.path.join()" function. The code then checks if the file exists using the "os.path.exists()" function. If the file exists, the code gets the file size using the "os.path.getsize()" function. If the file size is 0, the code prints a warning message indicating that an empty file was found. If the

```python
from PIL import Image
import os

# Specify the directory containing the image files
image_dir = "/content/drive/MyDrive/DataSet/FaceShapeDataset/testing_set/Square"

# Get a list of all files in the directory
files = os.listdir(image_dir)

# Iterate through the files and check for truncated (corrupted) images
for file in files:
    file_path = os.path.join(image_dir, file)
    try:
        with Image.open(file_path) as img:
            # Attempt to load the image to check for truncation
            img.load()
    except OSError as e:
        if "truncated" in str(e):
            print(f"Warning: Truncated image found: {file_path}")
        else:
            print(f"Error loading image: {file_path}, {e}")
```

Figure 15 - Code for Checking Images

file does not exist, the code prints a message indicating that the file was not found. This code is useful in preprocessing the image dataset by checking for any empty or missing files, which could affect the accuracy of the model.

This code snippet checks for truncated (corrupted) images in a directory using the Python Imaging Library (PIL). It imports the necessary Image class from PIL and the os module for directory and file handling.

Once we checked the entire dataset, then we need to crop faces from the dataset, save it, and create a new dataset. This below code snippet is using MTCNN (Multi-Task Cascaded Convolutional Neural Network) to detect faces in images and crop them. It reads the images from the specified dataset path and crops the faces from them using the crop_head() function. Then, it saves the cropped images to the specified save path. Here is an explanation of the code.

```
import cv2
import os
!pip install mtcnn
from mtcnn import MTCNN

face_detector = MTCNN()

def crop_head(image_path):
    img = cv2.imread(image_path)
    face = face_detector.detect_faces(img)
    for result in face:
        x, y, width, height = result['box']
        # head = img[y:y+height, x:x+width]

        head_width = int(1.2 * width)
        head_height = int(1.6 * height)
        head_x = int(x - 0.1 * width)
        head_y = int(y - 0.5 * height)
        head = img[max(head_y, 0):head_y+head_height, max(head_x, 0):head_x+head_width]
        return head

# Path to the dataset
dataset_path = '/content/drive/MyDrive/DataSet/FaceShapeDataset/training_set/Square'
# Path to the directory where cropped images will be saved
save_path = '/content/drive/MyDrive/DataSet/PreprocessedFaceShapeDataset/train/Square'

for root, dirs, files in os.walk(dataset_path):
    for file in files:
        if file.endswith('.jpg'):
            # Path to the image file
            image_path = os.path.join(root, file)
            # Crop the head from the image
            head = crop_head(image_path)
            # Save the cropped image to Google Drive
            save_file_path = os.path.join(save_path, file)
            try:
              cv2.imwrite(save_file_path, head)
            except:
              print(save_file_path)
```

Figure 16 - Code for Cropping images

Creating the face shape prediction model

The author used a stacking ensemble approach to improve the accuracy of the face shape prediction model. They trained three models using three different techniques: training from scratch, using EfficientNet, and using RegNet. These models were then combined using the ensemble approach, where the predictions from each model are stacked and used as inputs for a final classifier. The stacked predictions are used to train a meta-classifier, which outputs the final prediction. The author found that this approach improved the accuracy of the face shape prediction model compared to using a single model.

This code imports necessary modules for creating a face shape prediction model using CNN.

```
[ ]  from keras.models import Sequential
     from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, MaxPool2D
     from tensorflow.keras.preprocessing.image import ImageDataGenerator
     from keras.callbacks import EarlyStopping, LearningRateScheduler, ModelCheckpoint, ReduceLROnPlateau
     from keras.optimizers import Adam
     from tensorflow.keras import optimizers
```

Figure 17 - Code for Importing Library

The below code is setting up ImageDataGenerators for the training and testing sets of the face shape dataset. The training data is augmented with random rotations, horizontal shifts, and horizontal flips to improve the model's ability to generalize. The validation data is created by splitting off 20% of the training data. The testing data is not augmented or shuffled. The data is resized to 224x224, and the pixel values are normalized to be between 0 and 1.

```
train_path = '/content/drive/MyDrive/DataSet/PreprocessedFaceShapeDataset/train'
testing_path = '/content/drive/MyDrive/DataSet/PreprocessedFaceShapeDataset/test'

train_datagen = ImageDataGenerator(
    rescale=1./255, # Normalize pixel values
    rotation_range=20, # Rotate images randomly within a range of 20 degrees
    width_shift_range=0.1, # Shift images horizontally by a maximum of 10% of the image width
    height_shift_range=0.1, # Shift images vertically by a maximum of 10% of the image height
    horizontal_flip=True, # Flip images horizontally randomly
    vertical_flip=False, # Do not flip images vertically
    validation_split=0.2 # Split 20% of data for validation
)

test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    train_path, # Path to training data
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='training' # Use 80% of data for training
)

validation_generator = train_datagen.flow_from_directory(
    train_path, # Path to training data
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='validation' # Use 20% of data for validation
)

test_generator = test_datagen.flow_from_directory(
    testing_path, # Path to test data
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    shuffle=False # Do not shuffle the test data
)
```

Figure 18 - Code for Image Augmentation

```
input_tensor = Input(shape=(224, 224, 3))

base_model = RegNetY160(input_tensor=input_tensor, include_top=False, weights='imagenet')

x = Flatten()(base_model.output)
output_tensor = Dense(5, activation='softmax')(x)
model = Model(inputs=input_tensor, outputs=output_tensor)

loss = 'categorical_crossentropy'
optimizer = Adam(lr=1e-4)

model.compile(loss=loss, optimizer=optimizer, metrics=['accuracy'])
```

Figure 19 - Code for implementing the RegNet.

Above code is showing creating a model using RegNet in here the author using the RegNet version of RegNetY160 architectures

```
[ ]  # Define input image size
     input_size = (224, 224)

     # Load pre-trained EfficientNetV2M model
     base_model = EfficientNetV2S(input_shape=(input_size[0], input_size[1], 3), include_top=False, weights='imagenet')
```

```
     # Add global average pooling and dense output layers
     x = base_model.output
     x = GlobalAveragePooling2D()(x)
     x = Dense(256, activation='relu')(x) # Add additional dense layer for feature extraction
     predictions = Dense(5, activation='softmax')(x) # Output layer with 5 classes for face shapes
```

```
[ ]  # Create the model
     modelEfficientNetV2S = Model(inputs=base_model.input, outputs=predictions)
```

Figure 20 - Code for implementing the EfficientNet.

Above code is showing creating a model using EfficentNet in here the author using the EfficentNetV2-S version of EfficentNetV2 architectures

```
modelCNN = Sequential()
modelCNN.add(Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)))
modelCNN.add(MaxPooling2D((2, 2)))
modelCNN.add(Conv2D(64, (3, 3), activation='relu'))
modelCNN.add(MaxPooling2D((2, 2)))
modelCNN.add(Conv2D(128, (3, 3), activation='relu'))
modelCNN.add(MaxPooling2D((2, 2)))
modelCNN.add(Conv2D(128, (3, 3), activation='relu'))
modelCNN.add(MaxPooling2D((2, 2)))
modelCNN.add(Flatten())
modelCNN.add(Dropout(0.5))
modelCNN.add(Dense(512, activation='relu'))
modelCNN.add(Dense(5, activation='softmax'))
```

Figure 21 - Code for implementing Sequential.

This is a simple CNN model with 4 convolutional layers and 2 dense layers. Here is a brief explanation of each layer.

```
# Defining model file name
filePath ='EfficientNetV2-M'+ '_BestModel.h5'

# Defining model checkpoint for saving best model
bestModelCheckpoint = ModelCheckpoint(filePath, monitor='val_loss', verbose=1, save_best_only=True, save_weights_only=False, mode='auto', period=1)

# Defining decaying learning rate callback
decayLR = ReduceLROnPlateau(monitor='val_loss', factor=0.2, verbose=1, patience=3)

# Defining early stopping callback
earlystopCheckpoint = EarlyStopping(monitor="val_loss", min_delta=0, patience=9, verbose=1)

# Preparing list of all callbacks
callbacksList = [bestModelCheckpoint, decayLR, earlystopCheckpoint]
```

Figure 22 -Code of Model Creation

The above code is the same for all three models. This code is used to define three different types of callbacks for training a machine learning model: ModelCheckpoint, ReduceLROnPlateau, and EarlyStopping. ModelCheckpoint is a callback that saves the weights of the model at specific intervals during training, based on some metric such as validation loss. The saved model can then be loaded and used later for prediction or further training. ReduceLROnPlateau is a callback that reduces the learning rate when the validation loss plateaus, with the aim of allowing the optimizer to converge to a better minimum. EarlyStopping is a callback that stops the training process if the validation loss does not improve for a certain number of epochs, with the aim of avoiding overfitting. The code then creates a list of all three callbacks to be used during training.

```
# Train the model with the callback
history = modelEfficientNetV2S.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    epochs=100, # Number of training epochs
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // validation_generator.batch_size,
    callbacks=[callbacksList] # Add the callback to the training process
)
```

Figure 23 - Training the Model

This above code is similar for all three models in here we are calling the fit method to train a model. Once we trained these three models then we needed to use the ensemble technic to merge three models and create one model.

To create the ensemble method, we used three different architectures as mentioned above and merged all three models using the stacking method. To do this, the first step was to load the three

models. We used the load_model method from the Keras library to load each model. The code snippet below demonstrates this process:

```
model1 = load_model('/content/drive/MyDrive/FYP/face_shape_classification_model_RegNetY160_Version_01_Best_Only.h5')
model2 = load_model('/content/drive/MyDrive/FYP/face_shape_classification_model_CNN_Version_06_V2.h5')
model3 = load_model('/content/drive/MyDrive/FYP/face_shape_classification_model_modelEfficientNetV2S_Version_01_Best_Model.h5')
```

```
# Concatenate the outputs
concatenated = Concatenate()([outputs1, outputs2, outputs3])

# Add a dense layer
dense = Dense(units=5, activation='softmax')(concatenated)

# Create the ensemble model
ensemble_model = Model(inputs=inputs, outputs=dense, name='ensemble_model')

# Compile the model
ensemble_model.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
```

Figure 24 - Code for Ensemble Method

This code snippet shows how the concatenated outputs from the three models are passed to a dense layer and the ensemble model is created using the Functional API of Keras. The concatenated outputs are passed to a dense layer with 5 units and softmax activation function. The resulting model is then compiled using Adam optimizer with a learning rate of 0.001, categorical cross-entropy loss function, and accuracy as the evaluation metric. Once these things happened, the author used the fit method and successfully trained the model.

Creating a gender identification model

The code can be found in **Appendix 14 - Implementing the Gender Identification Model**

7.4 Implementation of the System

The code and explanation can be found in **Appendix 18 – Implementation of the System**.

7.4 User Interface

The user interface can be found in - **Appendix 05 - User Interface**

7.5 Chapter Summary

This chapter provided a brief explanation of the technologies used in the project. It is clearly stated what technologies were considered for development, which technologies were chosen, and why those technologies were chosen. Also, it provided the code and the user interface that were used in the development of this project.

CHAPTER 8: TESTING

8.1 Chapter Overview

This chapter is going to discuss the testing methodology used in the testing of the hair and beard style recommendation system. also going to discuss how functional and non-functional requirements are tested. This chapter also discusses the model evaluation methods and the results of those tests.

8.2 Objectives and Goals of Testing

The goal of testing is to determine whether the real system satisfies the expected requirements and to maintain the system's quality. The following testing objectives were established in order to improve the testing process.

- Verify that the system satisfies the functional requirements.
- Verify that the system satisfies the non-functional requirements.
- Verify the author followed the best coding practice while writing the codes.
- Verify that all system models are operating with high accuracy.
- Identify bugs and feature defects undetected in the development phase.

8.3 Testing Criteria

Author considered two testing criteria to test the system and the models those are :-

- Functional testing – Testing the functional requirements to make sure all the functions are working as expected.
- Structural testing- Testing the code developed to validate whether it is aligned with software engineering best practices.

8.4 Model Testing

8.4.1 Face Shape Prediction Model Testing

In order to create the face shape prediction model, the author used the ensemble method, which involved combining multiple models with various architectures. However this chapter, the author will only be testing the final ensemble model. There are several metrics that can be used to evaluate the performance of an image classification model. One of those is training and validation accuracy curve and training and validation loss curve. These curves provide insights

into the model's learning process during training and can help identify if the model is overfitting or underfitting. The training and validation accuracy curves show the accuracy of the model on the training and validation data over each epoch during training. The curves should be compared to see if the model is overfitting or underfitting. The training and validation loss curves show the loss of the model on the training and validation data over each epoch during training. The curves should be compared to see if the model is overfitting or underfitting. Because of that, analyzing these curves is an important part of evaluating the performance of an image classification model.

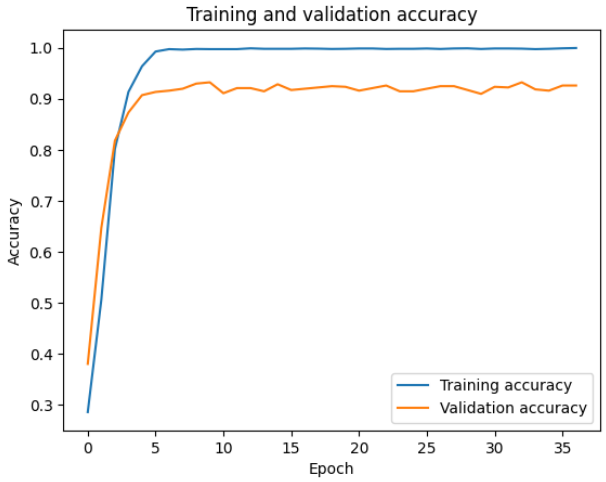


Figure 25 - Accuracy Curves

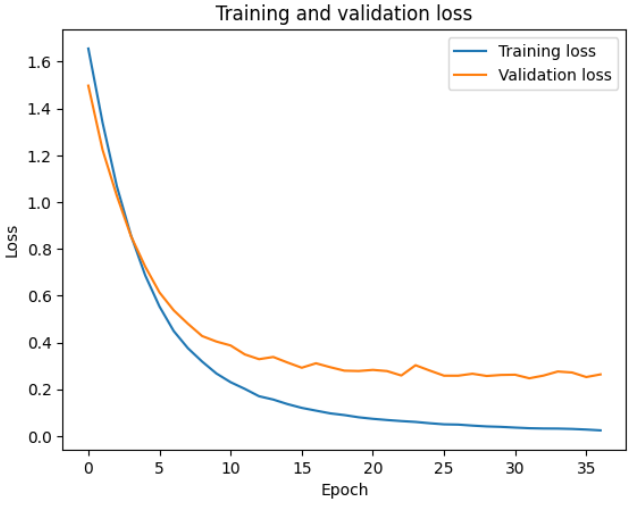


Figure 26 - Loss Curve

Based on the above curves, we can conclude that the model has a good performance. The difference between the training and validation curves is relatively small, indicating that there is

no significant overfitting or underfitting. Both the training and validation accuracy curves show a consistent increase over the epochs, with the validation accuracy staying close to the training accuracy. Similarly, the training and validation loss curves show a consistent decrease over the epochs, with the validation loss staying close to the training loss. This suggests that the model is generalizing well to unseen data and is not overfitting or underfitting. Therefore, the model can be considered suitable for the task of face shape prediction.

Accuracy is an important metric used to evaluate the performance of a machine learning model. It measures the percentage of correct predictions made by the model out of all predictions made. Accuracy is calculated by dividing the number of correct predictions by the total number of predictions and multiplying the result by 100 to get a percentage.

```
y_true = test_generator.classes
y_pred = modelMy.predict(test_generator)


accuracy = accuracy_score(y_true, y_pred.argmax(axis=-1))
score = accuracy*100
print(f"Accuracy: {(accuracy*100):.1f}%")
```

```
32/32 [==============================] - 17s 498ms/step
Accuracy: 89.4%
```

Figure 27 - Accuracy for Dataset 01

```
y_true = test_generator.classes
y_pred = modelMy.predict(test_generator)


accuracy = accuracy_score(y_true, y_pred.argmax(axis=-1))
score = accuracy*100
print(f"Accuracy: {(accuracy*100):.1f}%")
```

```
32/32 [==============================] - 23s 525ms/step
Accuracy: 89.5%
```

Figure 28 - Accuracy of Dataset 02

The above codes were used to calculate the accuracy of the model using the accuracy_score() method for calculate the accuracy. As the author had limited datasets for testing, most of the datasets were used for training, so author have two datasets for testing both resulted around 89.5%. Although this accuracy may seem relatively low, predicting a face shape is a complex task, and an accuracy of 89.5% is still considered good in many applications. Furthermore, the

model was trained on a limited dataset. It is worth noting that when compared with other existing models, this model's accuracy is significantly better.

Precision, Recall, and F1-Score are important metrics used in evaluating the performance of classification models. These metrics provide a more detailed understanding of the model's performance beyond just accuracy. Precision measures the proportion of true positive predictions over all positive predictions, while recall measures the proportion of true positive predictions over all actual positive instances. F1-Score is the harmonic mean of precision and recall and provides a single metric to evaluate the balance between precision and recall. By using these metrics, we can have a better understanding of how well the model is performing and where it may need improvements. All of these metrics can be calculated using the confusion matrix, precision, recall, and F1-score can be calculated as follows:

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1-score = 2 * ((Precision * Recall) / (Precision + Recall))

where TP is the number of true positive predictions, FP is the number of false positive predictions, and FN is the number of false negative predictions. In below code is used to calculate the Precision, Recall, and F1-Score for the ensemble model in here author using precision_score( ), recall_score( ), f1_score( ) methods to calculate the Precision, Recall, and F1-Score scores

```
y_true = to_categorical(test_generator.labels, num_classes=5)
y_pred = modelMy.predict(test_generator)

y_pred_class = y_pred.argmax(axis=1)
y_true_class = y_true.argmax(axis=1)

precision = precision_score(y_true_class, y_pred_class, average='weighted')
recall = recall_score(y_true_class, y_pred_class, average='weighted')
f1 = f1_score(y_true_class, y_pred_class, average='weighted')

print('Precision:', precision)
print('Recall:', recall)
print('F1-score:', f1)
```

```
32/32 [==============================] - 16s 499ms/step
Precision: 0.8945753064655834
Recall: 0.8946840521564694
F1-score: 0.8944356947105938
```

Figure 29 - Getting F1 Score

Based on the precision, recall, and F1-score values obtained from the evaluation of the model, we can conclude that the model has performed well in predicting the face shapes. The precision value of 0.894 indicates that out of all the predicted face shapes, 89.4% were correctly classified. The recall value of 0.894 also indicates that out of all the actual face shapes, 89.4% were correctly identified by the model. The F1-score of 0.894 suggests that the model has achieved a good balance between precision and recall. These metrics provide a more comprehensive understanding of the performance of the model beyond just accuracy and can help identify areas for improvement in future iterations of the model. Overall, these results suggest that the model is performing well and can be used for face shape prediction.

	precision	recall	f1-score	support
Heart	0.94	0.91	0.92	200
Oblong	0.91	0.92	0.91	200
Oval	0.84	0.81	0.83	198
Round	0.87	0.90	0.88	200
Square	0.91	0.93	0.92	199
accuracy			0.89	997
macro avg	0.89	0.89	0.89	997
weighted avg	0.89	0.89	0.89	997

Figure 30 - Classification Report

The **classification_report** function was used to generate a summary of the Precision, Recall, and F1-Score metrics for each class. This function is a valuable tool for assessing the model's performance for each individual class and detecting any class-specific issues that may require attention. the above image shows the classification report of the face shape prediction model. The results of the analysis showed that the model performed well for the Heart, Oblong, Round, and square face shapes, with Precision, Recall, and F1-Score scores above 0.90. However, for the oval class, the scores were lower, indicating that further improvements are necessary. Therefore, the report suggests that more focus should be given to enhancing the model's performance in identifying the oval face shape.

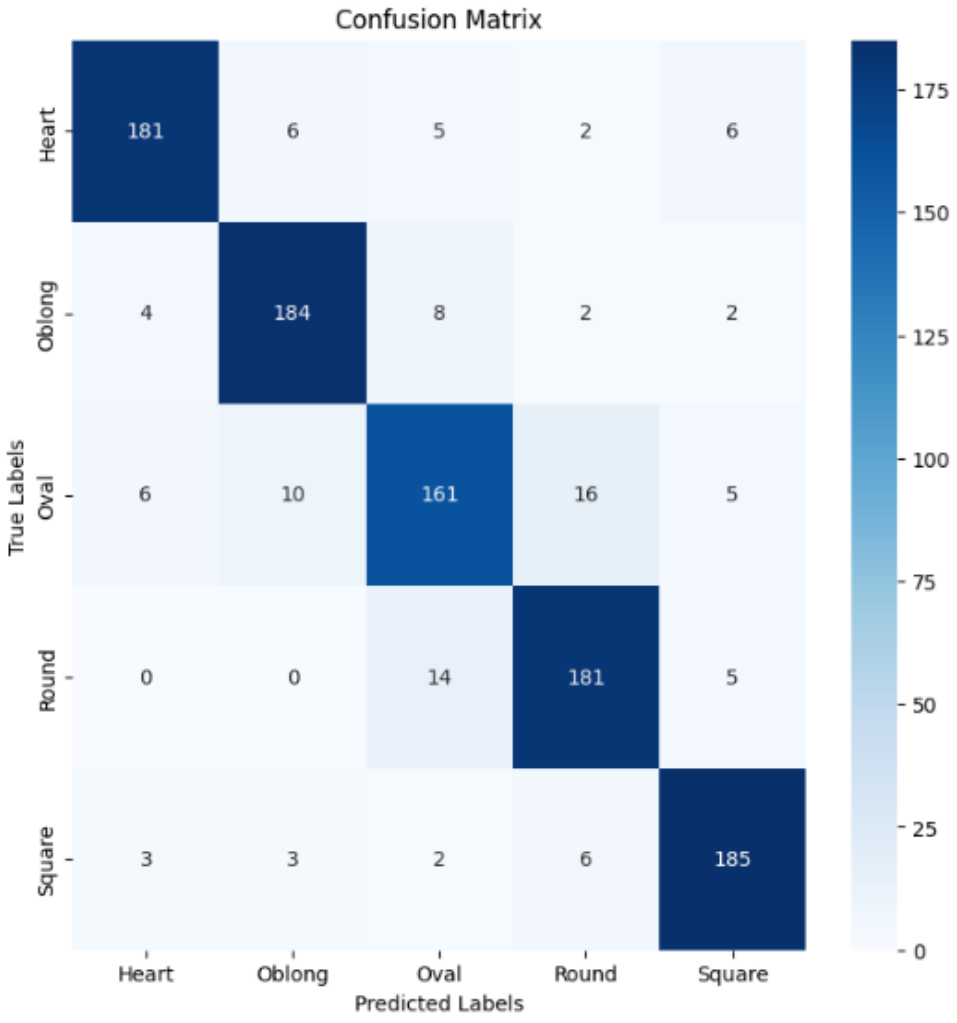


Figure 31 - Confusion Matrix

The Confusion Matrix depicted in the above image provides an evaluation of the model's classification performance. A Confusion Matrix is a table that showcases the true and predicted classes of a classification model. It helps in visualizing the classification results, enabling a better understanding of the model's accuracy. The results of the Confusion Matrix indicate that, apart from the oval face shape, the model has accurately predicted more than 180 images out of the approximately 195 images per class. This suggests that the model is highly reliable and can be effectively utilized for face shape prediction. The below code is used to generate the confusion matrix.

```python
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

# Generate predictions for the test set
y_pred = ensemble_model.predict(test_generator, steps=test_generator.samples // test_generator.batch_size + 1)
y_pred_classes = np.argmax(y_pred, axis=1)

# Get the true labels for the test set
y_true = test_generator.classes

# Compute the confusion matrix
conf_mat = confusion_matrix(y_true, y_pred_classes)

# Plot the confusion matrix as a heatmap
fig, ax = plt.subplots(figsize=(8, 8))
sns.heatmap(conf_mat, annot=True, cmap='Blues', fmt='d', ax=ax)
ax.set_xlabel('Predicted Labels')
ax.set_ylabel('True Labels')
ax.set_title('Confusion Matrix')
ax.xaxis.set_ticklabels(['Heart', 'Oblong', 'Oval', 'Round', 'Square'])
ax.yaxis.set_ticklabels(['Heart', 'Oblong', 'Oval', 'Round', 'Square'])
plt.show()
```

Figure 32 - Code for Confusion Matrix

An ROC (Receiver Operating Characteristic) curve is a graphical representation of a binary classification model's performance. It illustrates the trade-off between sensitivity (the true positive rate) and specificity (the true negative rate) for different classification thresholds. The area under the curve (AUC) is a measure of the model's ability to distinguish between positive and negative classes. A higher AUC indicates better performance, with an AUC of 1 representing a perfect classifier. The ROC curve is a useful tool for evaluating and comparing classification models and it is presented below.

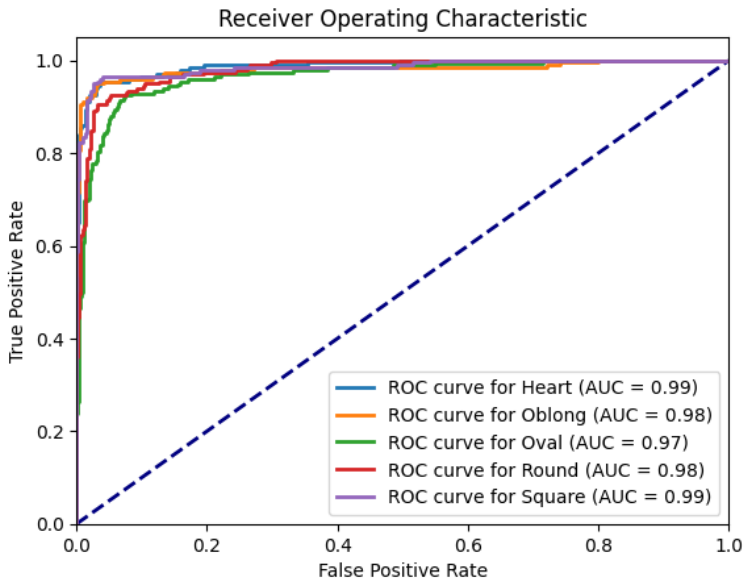


Figure 33 - ROC curve

In this case, the model demonstrated a strong ability to distinguish between positive and negative classes, as evidenced by the curve's steep rise and high area under the curve (AUC) score. This suggests that the model is effective at predicting the target variable and has good discriminatory power. Additionally, the curve hugs the top-left corner, indicating that the model has a low false positive rate and a high true positive rate. Overall, the ROC curve analysis indicates that the model is well-suited for the task of binary classification and performs at a high level of accuracy.

8.4.2 Gender Identification Model Testing

The Gender identification model testing can be found in **Appendix 15 - Gender Identification Model Testing**

8.5 Benchmarking

The benchmarking process is a crucial part of testing the image classification model for face shape prediction. The author used two different methods to benchmark the model. The first method involved trying various methods and architectures to achieve a high-quality model for face shape classification. For this, the author used the same Kaggle dataset for both training and testing the model. The dataset consisted of 800 images for training and 200 images for testing for each class.

During the testing process, the author used various metrics to evaluate the best performing model. These metrics included accuracy, precision, recall, F1 score, and mean average precision (mAP). After evaluating the different models, the author selected accuracy as the primary metric to determine the best model. Overall, the benchmarking process was rigorous, and the findings revealed the best model for face shape prediction with a high level of accuracy.

Table 17 - Benchmarking with Different Architecture

Method	**Description**	**Accuracy**
Sequential (model 01)	Using a Sequential method and training the model from scratch	71%
Sequential (model 02)	Using a Sequential method with more layer and training the model from scratch	76%
Inception	Using inception version 01 to train a CNN model	67%
InceptionV3	Using inception version 03 architectures to train	75.5%

	a CNN model	
SVM	Using Support Vector Machine algorithm to train a model	71%
GAN	Using Generative Adversarial Networks to train a deep learning model	77%
Functional API	Using Functional API to train a model from scratch	57%
VGG16	Using Vgg16 architectures to train a CNN model	71%
VGG19	Using Vgg19 architectures to train a CNN model	80.4%
EfficentNetV2-M	Using EfficentNetV2-M architectures to train a CNN model	86%
EfficentNetV2-S	Using EfficentNetV2-S architectures to train a CNN model	88%
RegNetX160	Using RegNetX160 architectures to train a CNN model	86%
RegNetY160	Using RegNetX160 architectures to train a CNN model	87.2%
Ensemble model 01	Using Vgg19, Inception and MobileNet architectures and used the ensemble technology and trained a model	87.6%
Ensemble model 01 (final model)	Using RegNetY160, EfficentNetV2-S and model trained from scratch using Sequential and used the ensemble technology and trained a model	89.5%

Based on the table presented, it can be concluded that ensembling models yields the highest accuracy among all the techniques. This is consistent with previous research that has shown that ensembling can significantly improve the performance of machine learning models.

Comparing the accuracy of a model with previously published research is another method to benchmark the model. This allows you to evaluate the performance of your model in comparison to other state-of-the-art models.

Table 18 - Benchmarking with Existing System

Method	Description	Accuracy
(Abdullah et al., 2022)	Ensemble method using VGG16	86.5%
(IEEE Computational Intelligence Society and Institute of Electrical and Electronics Engineers, no date)	Used ANN to do deep learning	60%
(Rahmat et al., 2018)	Used PNN (Probabilistic neural network)	80%
(Hossam et al., 2021)	Used SVM algorithm	82%
(Sánchez-Escobedo and Castelán, 2013)	Used PLS to increase the accuracy	75%
(Ratna Wati, Dinar Mutiara and Puji Widodo, 2021)	Used Random Forest modeling	82%
(Emmanuel and Tio, 2019)	Used inception v3 to create the model	84.4%
Proposed Model	Using RegNetY160, EfficentNetV2-S and model trained from scratch using Sequential and used the ensemble technology and trained a model	89.5%

Based on this also author proposed model is worked well

8.6 Functional Testing

To do the functional testing, the author conducts black box testing, and the results of those tests are presented in a table format.

Table 19 - Functional Testing

Id	User Action	Expected Outcome	Actual Outcome	Status
01	User uploading a heart face shape image of men and click the predict hair style button	System suggests hair style of men which is suitable for that face shape	System suggests hair style of men which is suitable for that face shape	Pass

02	User uploading an oblong face shape image of men click the predict hair style button	System suggests hair style of men which is suitable for that face shape	System suggests hair style of men which is suitable for that face shape	Pass
03	User uploading an oval face shape image of men click the predict hair style button	System suggests hair style of men which is suitable for that face shape	System suggests hair style of men which is suitable for that face shape	Pass
04	User uploading a round face shape image of men click the predict hair style button	System suggests hair style of men which is suitable for that face shape	System suggests hair style of men which is suitable for that face shape	Pass
05	User uploading a square face shape image of men click the predict hair style button	System suggests hair style of men which is suitable for that face shape	System suggests hair style of men which is suitable for that face shape	Pass
06	User uploading a heart face shape image of women click the predict hair style button	System suggests hair style of women which is suitable for that face shape	System suggests hair style of women which is suitable for that face shape	Pass
07	User uploading an oblong face shape image of	System suggests hair style of women which is suitable for that face shape	System suggests hair style of women which is suitable for that face shape	Pass

	women click the predict hair style button			
08	User uploading an oval face shape image of women click the predict hair style button	System suggests hair style of women which is suitable for that face shape	System suggests hair style of women which is suitable for that face shape	Pass
09	User uploading a round face shape image of women click the predict hair style button	System suggests hair style of women which is suitable for that face shape	System suggests hair style of women which is suitable for that face shape	Pass
10	User uploading a square face shape image of women click the predict hair style button	System suggests hair style of women which is suitable for that face shape	System suggests hair style of women which is suitable for that face shape	Pass
11	Click the predict button without add the image	Display a suitable error message.	Display a suitable error message.	Pass
12	Adding a file format that is not allowed by the system	Display a suitable error message.	Display a suitable error message.	Pass
13	Adding an image which is not contain any faces	Display a suitable error message.	Display a suitable error message.	Pass
14	Adding an	Display a suitable error	Display a suitable error	Pass

	image which is containing multiple faces	message.	message.	
15	User uploading a heart face shape image of men and click the predict beard style button	System suggests beard style of men which is suitable for that face shape	System suggests beard style of men which is suitable for that face shape	Pass
16	User uploading an oblong face shape image of men click the predict beard style button	System suggests beard style of men which is suitable for that face shape	System suggests beard style of men which is suitable for that face shape	Pass
17	User uploading an oval face shape image of men click the predict beard style button	System suggests beard style of men which is suitable for that face shape	System suggests beard style of men which is suitable for that face shape	Pass
18	User uploading a round face shape image of men click the predict beard style button	System suggests beard style of men which is suitable for that face shape	System suggests beard style of men which is suitable for that face shape	Pass
19	User uploading a square face shape image of men click the predict beard style button	System suggests beard style of men which is suitable for that face shape	System suggests beard style of men which is suitable for that face shape	Pass

20	User uploading a face shape image of Women and click the predict beard style button	Throw a proper error message	Throw a proper error message	Pass

8.7 Module and Integration Testing

Table 20 - Module and Integration Testing

Module	Input	Expected Output	Actual Output	Status
input validator	Wrong format image or non-image file type	Display error message	Display error message	Pass
	Image which not containing a face	Display error message	Display error message	Pass
	Correct image with one face	Move the image to image-process module	Move the image to image-process module	Pass
	Image which containing more than one faces	Display error message	Display error message	Pass
Image processing module	Face image	Return the processed and cropped image	Return the processed and cropped image	Pass
Gender identification module	Processed face image	Return the correct gender	Return the correct gender	Pass
Face shape identification module	Processed face image	Return the correct face shape	Return the correct face shape	Pass
Display Module	Face shape and gender	Suggest the hair or beard style	Suggest the hair or beard style	Pass

8.8 Non-Functional Testing

Non-functional testing was aligned to the non-functional requirements defined in Chapter 4.

8.8.1 Accuracy

Accuracy is a crucial aspect of image classification models, and it determines the effectiveness of the model. The testing chapter of this project has provided different ways to check the accuracy of both the face shape prediction and gender identification models. After rigorous testing, the models have shown good accuracy levels, which is promising for real-world usage. The face shape prediction model has an accuracy rate of 89.5%, while the gender identification model has an accuracy rate of 97.5%. These results indicate that the models can perform well in practical applications, although more data may be needed to improve their accuracy further.

8.8.2 Performance Testing

The performance of an image classification model is a crucial factor in its real-world application. To evaluate the performance of the models, the author conducted performance testing by measuring the time taken for the models to predict an image. Specifically, 100 images were used to calculate the total time, which was then divided by 100 to obtain the average time taken to predict one image. This provides insight into the real-time efficiency of the models and their ability to process images in a timely manner.

```
start_time = time.time()
prediction = model.predict(image)
end_time = time.time()
inference_time = end_time - start_time
total_time = total_time + inference_time
```

Figure 34 - Cod for Performance Testing

The code provided above calculates the time taken to make predictions on images. The face shape prediction model takes around 5 seconds to make a prediction, which may seem slightly longer. However, considering the limited resources and the need to predict the face shape, it is acceptable. On the other hand, the gender identification model takes only 0.25 seconds to make a prediction, which is a good performance. Based on these results, we can conclude that both models are suitable for real-world use.

Another important factor is the performance of the GUI in order to do that author used the Lighthouse tool which available in Google Chrome.

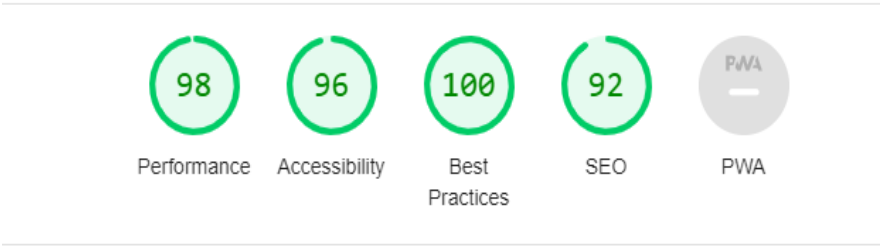


Figure 35 - Rainbow Testing Result

In here all the scores are above 90 percentage, and the performance is 98% which is very good performance in the real world. GUI performance, best practices, accessibility, and SEO all had good scores.

8.8.3 Security

Security is an essential aspect of any software system, and the StyleMe system developed by the author follows industry-standard coding practices and guidelines to ensure its security. The system does not access any user data, which reduces the risk of data theft. Moreover, the author has thoroughly tested the system for vulnerabilities and has taken measures to address any security issues identified during the testing phase. Overall, the system is secure and safe to use.

Other non-functional requirements were also tested and those also met the academic standards. The usability of the system was tested with various end users, and it was confirmed that the system is highly usable. The maintainability of the system was also tested, and it was found it is high because the author followed all the best practices and added comments for each code segment, making it easy to edit the codes in the future. Furthermore, to test the compatibility of the system, the author tried to run the system on different browsers, including Google Chrome, Microsoft Edge, and Firefox, and it was successful on all three browsers. Overall, the system met the academic standards for non-functional requirements such as usability, maintainability, and compatibility.

8.9 Limitations of the Testing Process

Although different testing methodologies may be followed to test the system to ensure that it is working as expected, there are certain limitations that may prevent the identification of bugs. One limitation is that the testing process is typically conducted on a limited number of devices or hardware configurations, which may not represent the full range of devices on which the system may be used. Additionally, the testing process may not be able to account for all possible user

behaviors and edge cases that could cause errors or unexpected results. Another limitation is that the testing process may not be able to fully simulate real-world scenarios, such as high traffic loads or network connectivity issues, which could impact system performance and stability. Finally, the testing process may be constrained by time and resource limitations, which could prevent the testing team from conducting comprehensive testing across all system components and features.

8.10 Chapter Summary

This chapter detailed every form of test that was run on both the research component and the finished product. In order to find the models with the best performance while ensuring that it is not biased, extensive model testing was performed using a variety of hyperparameter combinations. the training data being overfit. The system was then benchmarked using the best-performing models that were generated, including a generalized model that generalizes well across a variety of circumstances. Additionally, functional, integration, and non-functional tests were run to ensure that the research and application meet the established standards and function as expected.

09 EVALUATION

9.1 Chapter Overview

This chapter gives a summary of the comments received from researchers, academic experts, technical experts, and domain experts regarding many facets of the research and prototype, such as the novelty of the idea, the caliber of the finished product, etc. There is also a presentation of the author's assessment of many issues. To make sure the system complies with the anticipated requirements, functional and non-functional needs were examined.

9.2 Evaluation Methodology and Approach

A project's evaluation is used as a measure of its success. Feedback was gathered on the research challenge, the project's design, execution, testing, and other key aspects. The major objective of the research project StyleMe is to suggest an image classification model for CNN-based face shape prediction. As a result, the author has decided to use both qualitative and quantitative methods to assess the project's progress. Deductive reasoning theme analysis was applied to the qualitative material to be analyzed. In addition to the qualitative evaluations, quantitative evaluation input was also received during the evaluation phase and was analyzed using descriptive analysis before being provided here. To get feedback, surveys were given out and domain experts were contacted for interviews.

9.3 Evaluation Criteria

The following evaluation criteria were defined according to the themes identified for the evaluation.

Table 21 - Evaluation Criteria

Criteria	**Purpose to evaluate**
Choice of research topic and domain	To validate the significance of the choices of topic, domain, research gap, and depth undertaken in this research.
Research contribution	To validate the authors research contribution in the domain and technical part of the research.
System design, architecture and implementation	Evaluate whether the system design and architecture design are completed up to the proper standards
Solution and Prototype	To evaluate the implemented prototype acts as a proof of

	concept to the proposed solution
possible improvements	To unveil possible improvements that could be worked on as Future Work related to the conducted research.

9.4 Self-Evaluation

The self-evaluation is done by the author according to the evaluation criteria mentioned in Chapter 9.3. It's presented below.

Table 22 - Self-Evaluation

Criterion	**Author's Self-evaluation**
Choice of research topic	The fashion industry is one of the most significant and rapidly growing industries worldwide, with an increasing focus on digital technologies. The integration of deep learning technologies into the fashion industry has opened up new avenues for growth and innovation.
Research contribution	The author's research contribution can be divided into two parts: technical and domain specific. From a technical standpoint, the author made several contributions to the suggestion system. Firstly, the author utilized an ensemble approach to create a high accuracy model for face shape prediction. Secondly, the author employed a new CNN architecture, EfficientNetV2-M, which had not been previously used for gender identification and achieved high accuracy. In terms of domain-specific contribution, the author's system offered a novel combination of suggesting hair and beard styles, which is a unique feature. Additionally, the system suggested hairstyles for both men and women, which is a valuable feature as previous systems often focused on one gender only. Overall, the author's research has contributed to the field of fashion industry by utilizing deep learning technology to create a comprehensive suggestion system for hair and beard styling

	that caters to both genders.
System design, architecture and implementation	The system design, architecture, and implementation of the project met the expectations of the research. The system design was well thought out and played a significant role in creating a high-performing system. The majority of functional and non-functional requirements were met, and the code quality and industrial standards were considered throughout the design and implementation phase. The features, design/architecture, and implementation were completed at a satisfactory level, with a focus on meeting the research objectives and delivering a reliable system. Overall, the system design, architecture, and implementation were completed to a high standard, reflecting a thorough and diligent approach to the research project.
Solution and Prototype	The prototype web application developed as a part of this research project has been designed and implemented with utmost care and attention to detail. It offers a user-friendly interface with smooth navigation and seamless integration of deep learning technologies to provide accurate and personalized fashion recommendations. During the development process, various testing methods were employed to ensure that the system is functional, secure, and reliable. The testing chapter in the report provides a detailed account of the various tests that were conducted, including unit testing, integration testing, and system testing. The results of the tests were highly positive, confirming that the system's majority functionality is working correctly and meeting the expectations set for it. The prototype is capable of delivering accurate and personalized hair or beard style recommendations to users,

	making it a valuable tool for the fashion industry.
possible improvements	There are several possible improvements that can be made to the system. One of the main limitations is the availability and quality of datasets. If a larger, more diverse, and high-quality dataset is used, the system could achieve even higher accuracy and performance. Additionally, since the system is intended for use in the fashion industry, there is potential to add more functionality, such as integrating with social media platforms or providing personalized recommendations based on user preferences and history. Moreover, further research and experimentation could be conducted on the use of more advanced deep learning models or techniques to improve the accuracy and efficiency of the system. Furthermore, enhancing the user interface and experience could also make the system more user-friendly and appealing to a wider audience.

9.5 Limitations of Evaluation

The limitations of the evaluation conducted in this research include the relatively small sample size of evaluators and the limited scope of the evaluation. Due to time and resource constraints, only a small number of evaluators could be included, which may not be representative of the wider user base. Additionally, the evaluation focused only on specific functionalities and aspects of the system, which may not provide a comprehensive understanding of the system's overall performance and usability. Furthermore, the evaluation was conducted in a controlled environment and may not accurately reflect the system's performance in real-world scenarios.

9.6 Evaluation on Functional Requirements

The breakdown of the evaluation of functional requirements is presented in **Appendix 06 - Evaluation on Functional Requirements**

9.7 Evaluation on Non-Functional Requirements

The evaluation of non-functional requirement is presented in **Appendix 07 - Evaluation on Non-Functional Requirements**

9.8 Chapter Summary

The evaluation elements of the research were reviewed in this chapter. The methods used for evaluation were presented along with the justification for each choice. Prior to the author's self-assessment and comments from the assessors, the evaluation criteria were established. The The evaluators' opinions were categorized into topics and presented according to the pre-established standard. The functional and non-functional needs were assessed in the end.

10 CONCLUSIONS

10.1 Chapter Overview

The final chapter of this report serves as a conclusion to the research conducted, summarizing the key findings, and outlining the project's contribution to the research community. The chapter also discusses the project's objectives and how they were achieved, highlighting the challenges faced along the way. Additionally, the author reflects on the knowledge and skills gained throughout the project, including the utilization of prior knowledge and the modules of the degree program. These new insights are documented and analyzed in this chapter.

10.2 Achievements of Research Aims & Objectives

The achievement of the aim and objectives were discussed below.

10.2.1 Achievements of aim

The aim of this research is to design, develop, and evaluate a recommendation system that can analyze the face of a user and predict the face shape of the user. Then, according to the face shape, it will recommend the hair or beard style for the users.

The aim of this research was to develop an image classification system to predict a face shape using Convolutional Neural Networks (CNNs) and transfer learning, and to improve its performance through ensembling. The aim was successfully achieved by implementing transfer learning using a pre-trained CNN model and fine-tuning it on a custom dataset. The system was then further optimized through ensembling, which involved combining the outputs of multiple CNN models to generate a final prediction. The proposed approach was tested and evaluated on a custom dataset, and the results confirmed the effectiveness of the system for image classification. The use of transfer learning and ensembling resulted in improved performance, and the proposed approach can be generalized for use in other image classification tasks.

10.2.2 Research Objectives

Achievement of research objectives can be found in **Appendix 08 - Achievements of Research Objectives.**

10.3 Utilization of Knowledge from the Course

The below table will clearly explain how the software engineering course modules helped the author successfully complete the research project.

Table 23 - Utilization of Knowledge from the Course

Module(s)	**Utilized Knowledge**
Programming Principles 1, 2 & Object-Oriented Programming	These modules gave a basic understanding of programming languages and best practices that needed to flow during the implementation of the project. Also, the author learned the basics of Python during these modules, which helped him during the creation of the module. He also used the Flask framework for creating the web app, for which a good knowledge of Python is required.
Web Design and Development, Advanced Server-Side Web Programming & Server-side Web Development	The research project is based on a web app, so these modules helped the author learn about HTML, CSS, JavaScript, web design libraries, and frameworks, which helped during the development of the StyleME website. During the Advanced Server-Side Web Programming module, the author learned about the MVC pattern, which helped to build a prototype with Flask. The prototype was made with the UI/UX guidelines taught in this module.
Software Development Group Project	This module is almost identical to the final year project module; the only difference is that it's a group project, while the final year project is an individual project, so it helped the author in the several stages of projects like report writing, implementation, literature review, evaluation, and testing.
Algorithms: Theory Design and Implementation	The most crucial information about algorithms and data structures was supplied in this module. This was beneficial for using logical reasoning to design crucial algorithms with minimal time complexity.
Client Server Architecture & Computer Systems Fundamental	These modules gave a basic understanding about data science and also helped to build a system structure.

10.4 Use of Existing Skills

Skills developed during the software engineering degree program were utilized to complete the project. The skills attained are as follows,

Python: - This is a data science project that uses CNN, so knowledge of Python and Python-related libraries is very important. During the academy periods, the author learned Python, which

helped him train the image classification model and preprocess the data. Also, the author builds the system with Flask, which is a Python-based framework that the author learned during the software development group project module.

Industrial Placement: - The internship at Creio360 gave a practical knowledge of industry standards for developing systems and choosing architectural and implementation options. Working in an agile environment with industry professionals has given a experience in effectively producing programs in a set amount of time.

Web Design and Development: - During the course of the academy program, the author gained extensive knowledge in a variety of programming languages and web development tools such as HTML, CSS, JavaScript, web-based frameworks, and web-based libraries. This knowledge was instrumental in developing a custom website for the project. The author was able to leverage their understanding of UI/UX guidelines and best practices to ensure that the website was user-friendly and visually appealing.

10.5 Use of New Skills

Author learned lot of new skills for this final year project

Deep Learning: - The author of this project did not have prior experience in deep learning. To overcome this, the author learned deep learning skills through online courses on Udacity, Coursera, and Udemy. These courses covered essential topics such as Convolutional Neural Networks (CNNs) and transfer learning. The author gained both theoretical knowledge and practical experience through the development and testing of the image classification system. With these newly acquired skills, the author was able to complete the project successfully and gained valuable knowledge for future deep learning projects.

Data pre-processing: - Data pre-processing plays a critical role in the success of any machine learning project. The author of this project recognized this and devoted significant effort to pre-processing the image dataset before training the image classification system. The pre-processing steps included data cleaning, resizing, and normalization. Data cleaning involved removing any corrupt or irrelevant images from the dataset to ensure that only relevant and high-quality data was used for training. The images were then resized to a standard size to reduce computational complexity during training. Finally, normalization was performed to ensure that the pixel values

of the images fell within a standardized range, enabling better model convergence during training. Through these pre-processing steps, the author was able to obtain a high-quality image dataset that was well-suited for training the image classification system.

Image Classification Model: - Prior to working on this project, the author had no prior experience with Convolutional Neural Networks (CNNs), which are commonly used for image classification tasks. To develop the image classification system for predicting face shape, the author extensively studied the theoretical foundations of CNNs, including their architecture and training techniques. The author also explored their application to image classification tasks, specifically for classifying faces into five distinct shapes. Through this research, the author gained a solid understanding of the underlying concepts and was able to apply this knowledge to develop the image classification system successfully. This experience provided the author with valuable knowledge and skills in the field of image classification and deep learning, which can be applied to future projects in the same field.

10.6 Achievement of Learning Outcomes

Achievement of Learning Outcomes can be found in **Appendix 16 - Achievement of Learning Outcomes**

10.7 Problem and Challenge Faced

Table 24 - Problem and Challenge Faced

Problem and Challenge	**Mitigation**
Vast scope of the project.	When coming to suggest the hair or beard style author need to consider lot of factors like face shape, type of hair, thickness of hair and lot more so author limited the project to the face shape
Lack of dataset: there is no proper dataset for face shapes, and the existing datasets are small and not accurate.	In order to fix that, the author merged two datasets, removed the unwanted images, and cropped the faces of the images to create new data sets for training purposes.
Lack of knowledge in the research topic	The research topic is related to deep learning, especially CNN, but the author didn't have any experience in this sector, so he enrolled in some courses to get an understanding of these topics.

High end hardware requirements.	Since the image classification models used in this project required a lot of resources to train efficiently, the author needed to find a solution to this issue. To overcome this challenge, the author decided to use Google Colab, which is a free cloud-based platform that provides access to high-end hardware to train deep learning models. By using this platform, the author was able to train the models efficiently without needing expensive hardware.
Time constraints	The author faced some challenges while working on this project. One of the major challenges was managing time effectively, as the project had to be completed within a certain deadline. To overcome this challenge, the author had to limit the scope of the project and prioritize essential features. The author also had to balance the workload from other modules alongside the project. Despite these challenges, the author was able to develop a functional image classification system for predicting face shape using CNNs. This experience helped the author learn valuable lessons about project management and time management.

10.8 Deviations

The project was developed according to the planned design without any significant deviations. However, during the development stage, the author encountered a challenge regarding the gender specificity of the hair and beard styles suggested by the system. The initial design was meant to suggest hair and beard styles for men only. However, the author recognized the need to incorporate a gender identification feature to make the system more inclusive and comprehensive. To address this issue, the author developed a gender identification model with high accuracy, which could determine the gender of the user from their facial image.

Incorporating this feature improved the overall functionality of the system. By identifying the gender of the user, the system could suggest the appropriate hair and beard styles that align with the user's gender. This not only made the system more inclusive but also provided a more personalized experience for the user. The gender identification model was trained using deep

learning techniques, specifically using convolutional neural networks (CNNs). The model achieved a high level of accuracy, which made it suitable for integration into the hair and beard style suggestion system.

Overall, this addition to the system demonstrated the flexibility and adaptability of the project to accommodate unforeseen challenges and enhance the functionality of the system.

10.9 Limitation of the Research

While this research has successfully achieved its aim of suggesting hair and beard styles based on face shape prediction, it's important to note its limitations. In this section, the author will explore the specific limitations of the research and discuss their potential impact on the results and conclusions.

- One limitation of this research is that it only considers face shape as a factor in suggesting hair and beard styles. While face shape is an important factor in determining suitable styles, it may not be the only factor that users consider when choosing a style. Other factors such as hair type, face color, and personal preferences may also play a role. However, due to time constraints, the author chose to focus on face shape as the primary factor in the system. Future research could expand on this work by considering additional factors to provide more personalized and accurate recommendations.

- One of the limitations of this research is the limited dataset used for training the model. The dataset contains only 800 images for each of the five face shape classes, with 200 images for validation and 200 images for testing. Although the model achieved satisfactory performance, a larger and more diverse data set could potentially lead to even better results.

- Although the prediction model has achieved a high accuracy rate of around 90%, it is possible to further improve the accuracy with the use of a larger dataset and more powerful hardware. With more data and resources, the accuracy of the model can potentially be increased up to 95%.

- The limitation of the current system is that it is only integrated as a web page due to time constraints and limited knowledge. However, it would be beneficial to create a mobile app for this system in the future to increase its accessibility and reach a wider audience.

- The system was trained to recognize only the 5 major face shapes among more than 7 types that exist in the real world. This was due to the limited availability of data for other face shapes. Therefore, the system may not be as accurate in identifying and suggesting hairstyles for people with other face shapes.

- During the development of the image classification model for face shape detection, the author attempted to use various techniques available for Convolutional Neural Networks (CNNs) to improve the accuracy of the model. However, due to time constraints and limited knowledge, it is possible that some techniques have not been utilized, which could have further improved the accuracy of the model.

10.10 Future Enhancements

In this section, we will discuss some possible feature enhancements that could be made to the system to improve its functionality and user experience.

- One potential feature enhancement for this research could be to expand beyond just considering face shape when suggesting hair and beard styles. While face shape is an important factor, it may not be the only consideration for users. Other factors such as hair type, face color, and personal preferences could also play a significant role in determining the best style for an individual. While time constraints limited the author's ability to consider these factors in this research, future work could aim to incorporate them to provide more personalized and accurate recommendations.

- One feature enhancement that could be implemented in the future is to transform the proposed system from a website to a mobile app. While the website is accessible through various devices, a mobile app could provide more convenience for users, allowing them to easily access the system on-the-go. This could potentially increase user engagement and

satisfaction with the system. The author plans to explore the development of a mobile app for StyleMe in the future.

- To enhance the performance and accuracy of the system, future research can explore various deep learning techniques and methods. The use of more advanced hardware, such as GPUs or TPUs, can also improve the efficiency and speed of the training process. Moreover, training the model with a larger dataset can further improve the accuracy of the predictions. By incorporating more diverse data, the model can learn to recognize and suggest styles that are suitable for a wider range of individuals.

- The proposed system currently only suggests hair and beard styles based on the user's face shape. However, there are many other style choices that can complement these recommendations, such as eyeglasses, earrings, or other accessories. In future work, the author plans to expand the system to include recommendations for these additional style choices, providing users with a more complete and personalized fashion experience.

10.11 Achievement of the contribution to body of knowledge

By conclude this project the Arthur by concluding the research project, the author has managed to make contributions in the domain of hair and beard style industry, the technology of face prediction system, and towards the research process.

10.11.1 Technical Contribution (Classification Systems)

- The author proven ensembling the EfficentNetV2-S architecture, RegNetY160 architecture and CNN model using sequential (from the scratch) architecture resulting in a highly accurate model for image classification.
- Author proven EfficientNet architecture and RegNet can be used for face shape prediction task which will give more accuracy compared to old architecture.
- Author proven EfficientNet architecture can be used for gender identification task which will give more accuracy compared to old architecture.
- The author combined face shape prediction model and gender identification model to give a more personalized suggestion.

10.11.2 Domain Contribution

The author created a recommendation system for combining the hair style and beard style suggestion also it will suggest based on the gender of the user.

10.12 Concluding Remarks

In conclusion, the proposed StyleMe system utilizes deep learning techniques to suggest hair and beard styles based on a user's face shape. While the system has shown promising results with an accuracy rate of nearly 90%, there are limitations to the research, such as a limited dataset and the focus on only face shape as a factor in recommending styles.

To enhance the system's accuracy and usability, future work could involve expanding the dataset, considering additional factors such as hair type and color, and developing a mobile app version of the system. Moreover, trying different techniques from deep learning and improving the hardware can also lead to better performance.

Overall, the proposed system has the potential to revolutionize the way people choose their hair and beard styles. With further research and development, the StyleMe system can provide more personalized and accurate recommendations, making it a valuable tool in the beauty and grooming industry.

REFERENCES

- Abdullah et al. (2022). Hybrid Based Model Face Shape Classification Using Ensemble Method for Hairstyle Recommender System. *Lecture Notes in Networks and Systems*. 2022. Springer Science and Business Media Deutschland GmbH, 61–68. Available from https://doi.org/10.1007/978-981-16-9480-6_7.

- Alzahrani, T., Al-Nuaimy, W. and Al-Bander, B. (2021). Integrated multi-model face shape and eye attributes identification for hair style and eyelashes recommendation. *Computation*, 9 (5). Available from https://doi.org/10.3390/computation9050054.

- Ao, S.-I., Rieger, B. and Amouzegar, M.A. (eds.). (2010). Machine Learning and Systems Engineering. 68. Available from https://doi.org/10.1007/978-90-481-9419-3 [Accessed 5 May 2023].

- Baixian, Z. et al. (2018). Image classification with a new kind of shape representation. 29 October 2018. SPIE-Intl Soc Optical Eng, 67. Available from https://doi.org/10.1117/12.2513989.

- Bell, J. (2022). What is machine learning? *Machine Learning and the City: Applications in Architecture and Urban Design*, 209–216. Available from https://doi.org/10.1007/978-3-319-18305-3_1/COVER [Accessed 5 May 2023].

- Bird, J.J. and Faria, D.R. (2018). A Study on CNN Transfer Learning for Image Classification Combining NLP and Image Processing for Multimedia Retrieval (KTP UK) View project AMS-HMI: Assisted Mobility Supported by Shared-Control and Advanced Human-Machine Interfaces View project. Available from https://www.researchgate.net/publication/325803364 [Accessed 5 May 2023].

- Emmanuel, A. and Tio, D.C. (2021a). *Face shape classification using Inception v3*. Available from https://github.com/adonistio/inception-face-shape-classifier.

- Emmanuel, A. and Tio, D.C. (2021b). *Face shape classification using Inception v3*. Available from https://github.com/adonistio/inception-face-shape-classifier

- *GLOBAL GARMENT AND TEXTILE INDUSTRIES Workers, Rights and Working Conditions*. (2018).

- Here Are the Best Haircuts for Your Face Shape. (no date). Available from https://www.instyle.com/hair/find-perfect-cut-your-face-shape [Accessed 5 May 2023].

- Hossam, M. et al. (2021). A Comparative Study of Different Face Shape Classification Techniques. *2021 International Conference on Electronic Engineering (ICEEM)*. Available from https://doi.org/10.1109/ICEEM52022.2021.9480638 [Accessed 4 May 2023].

- How To Determine The Shape Of Your Face And 6 Different Types. Available from https://www.stylecraze.com/articles/know-your-face/ [Accessed 5 May 2023].

- IEEE Computational Intelligence Society and Institute of Electrical and Electronics Engineers. (no date). *2016 Eighth International Conference on Advanced Computational Intelligence (ICACI) : 14-16 Feb. 2016.*

- Jason, B. (2022). When to Use MLP, CNN, and RNN Neural Networks . Available from https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/ [Accessed 6 May 2023].

- Kothari, R. (2019). Research Mathodology : Methods and Techniques. University of Rajasthan. Available from http://lms.aambc.edu.et:8080/xmlui/bitstream/handle/123456789/212/Research%20Methodology%20Methods%20and%20Techniques%20(%20PDFDrive%20).pdf?sequence=1 [Accessed 8 November 2022].

- Liu, J. et al. (2020). EfficientNet based recognition of maize diseases by leaf image classification. *Journal of Physics: Conference Series*, 1693 (1), 012148. Available from https://doi.org/10.1088/1742-6596/1693/1/012148 [Accessed 6 May 2023].

- Liu, Z. et al. (2019). *Design and Implementation of Hair Recommendation System Based on Face Recognition*.

- Mahesh, B. (2018). Machine Learning Algorithms-A Review Self Flowing Generator View project Machine Learning Algorithms-A Review View project Batta Mahesh Independent Researcher Machine Learning Algorithms-A Review. *International Journal of Science and Research*. Available from https://doi.org/10.21275/ART20203995 [Accessed 5 May 2023].

- Mayank Mishra. (2021). Convolutional Neural Networks . Available from https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939 [Accessed 5 May 2023].

- Mehta, A. et al. (2021). Human Face Shape Classification with Machine Learning Human Face Shape Classification with Machine Learning View project The Story Behind the 2020 U.S. Presidential Election: A Big Data Analysis Using Twitter View project Human Face Shape Classification with Machine Learning. Available from https://doi.org/10.13140/RG.2.2.28282.57282.

- Mehta, A. et al. (no date). Human Face Shape Classification with Machine Learning Human Face Shape Classification with Machine Learning View project The Story Behind the 2020 U.S. Presidential Election: A Big Data Analysis Using Twitter View project Human Face Shape Classification with Machine Learning. Available from https://doi.org/10.13140/RG.2.2.28282.57282.

- Pasupa, K., Sunhem, W. and Loo, C.K. (2019a). A hybrid approach to building face shape classifier for hairstyle recommender system. *Expert Systems with Applications*, 120, 14–32. Available from https://doi.org/10.1016/j.eswa.2018.11.011.

- Pasupa, K., Sunhem, W. and Loo, C.K. (2019b). A hybrid approach to building face shape classifier for hairstyle recommender system. *Expert Systems with Applications*, 120, 14–32. Available from https://doi.org/10.1016/j.eswa.2018.11.011.

- *PROCEEDINGS PROCEEDINGS III RESULT*. (no date). Available from https://visualstudio.microsoft.com/vs/.

- Rahmat, R.F. et al. (2018). Probabilistic neural network and invariant moments for men face shape classification. *IOP Conference Series: Materials Science and Engineering*. 1 October 2018. Institute of Physics Publishing. Available from https://doi.org/10.1088/1757-899X/420/1/012095.

- Ratna Wati, W., Dinar Mutiara, K.N. and Puji Widodo, A. (2021). Random Forest For Hijab Style Selection Based on Face Shape Using Morphological Facial Index. *2021*

International Conference on Artificial Intelligence and Computer Science Technology (ICAICST), 96–101. Available from https://doi.org/10.1109/ICAICST53116.2021.9497814 [Accessed 4 May 2023].

- Sánchez-Escobedo, D. and Castelán, M. (2013). 3D face shape prediction from a frontal image using cylindrical coordinates and partial least squares. *Pattern Recognition Letters*, 34 (4), 389–399. Available from https://doi.org/10.1016/j.patrec.2012.09.007.

- Science Council of Asia. Conference (16th : 2016 : Colombo, S.L., National Science Foundation (Sri Lanka) and National Academy of Sciences - Sri Lanka. (no date). *16th Conference of the Science Council of Asia : abstracts and proceedings.*

- Shah, H.N. and Hao, J. (no date). *3D-GUIDED FACE SHAPE CLASSIFICATION A Project*.

- Smith, A.M. (2012). Research Methodology: A Step-by-step Guide for Beginners. *Nurse Education in Practice*, 12 (3), e25. Available from https://doi.org/10.1016/j.nepr.2011.11.008.

- Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104, 333–339. Available from https://doi.org/10.1016/j.jbusres.2019.07.039.

- Sun, Y. et al. (2021). Automatically Designing CNN Architectures Using Genetic Algorithm for Image Classification. Available from https://doi.org/10.1109/TCYB.2020.2983860 [Accessed 5 May 2023].

- Vittal, S.R., Gowda, N.H. and Sankapal, P.R. (2022). FACE SHAPE CLASSIFIER USING DEEP LEARNING. *International Research Journal of Engineering and Technology*. Available from www.irjet.net.

- Wang, Y. et al. (2022). Cascade Regression-Based Face Frontalization for Dynamic Facial Expression Analysis. *Cognitive Computation*, 14 (5), 1571–1584. Available from https://doi.org/10.1007/s12559-021-09843-8.

- Xu, J. et al. (2022). RegNet: Self-Regulated Network for Image Classification. *IEEE Transactions on Neural Networks and Learning Systems*, 1–6. Available from https://doi.org/10.1109/TNNLS.2022.3158966 [Accessed 6 May 2023].

- Young, S. et al. (2019). Eyeglasses frame selection based on oval face shape using convolutional neural network. *ICIC Express Letters, Part B: Applications*, 10 (8), 707–715. Available from https://doi.org/10.24507/icicelb.10.08.707.

APPENDIX

Appendix 01 – Research Objectives

Table 25 - Research Objectives

Objective	**Description**	**Learning Outcomes**
Literature Survey	Analyze the previews works and critically evaluate them. • **R01:** Conduct a preliminary study on existing Models, Systems & Algorithms. • **R02:** Analyze the hairs and beard style recommendations system technics. • **R03:** Analyze the face shape prediction system and algorithms related to the face shape prediction model	LO1, LO6
Design	designing an architecture and a system that can solve the identified problems with the recommended techniques. • **R04**: Design a System which can recommend a hair or beard style to user. • **R05**: Design an image processing model which can predict the users face shape. • **R06**: Design an image processing model which can identify the users face	LO1
Requirement Analysis	Carrying out an in-depth user requirement gathering phase for the following areas • **R07**: Gather the requirements of a Recommendations System and understand end-user expectations. • **R08**: Conduct the user survey and collect details about users' expectations regarding this system. • **R09**: Get the opinions of domine and	L02, L03, L04, L05, L06

	technology expect regarding the system and understanding the experts' expectations.	
Development	Implementing the system that capable to full fill all scopes • **R10**: Develop the recommendation system which can analyst the user face and according to that face it will suggest the hair and beard style. • **R11**: develop a graphical user interface which is easy to use for user.	L03, L04, L06
Testing and Evaluation	Testing the created system & Data science models with appropriate data and evaluating them with baseline techniques identified in the literature. • **R12**: To create test plan and perform various unit, integration and functional testing. • **R13**: Evaluating the face shape prediction model.	L04, L06

Appendix 02 - Gantt Chart

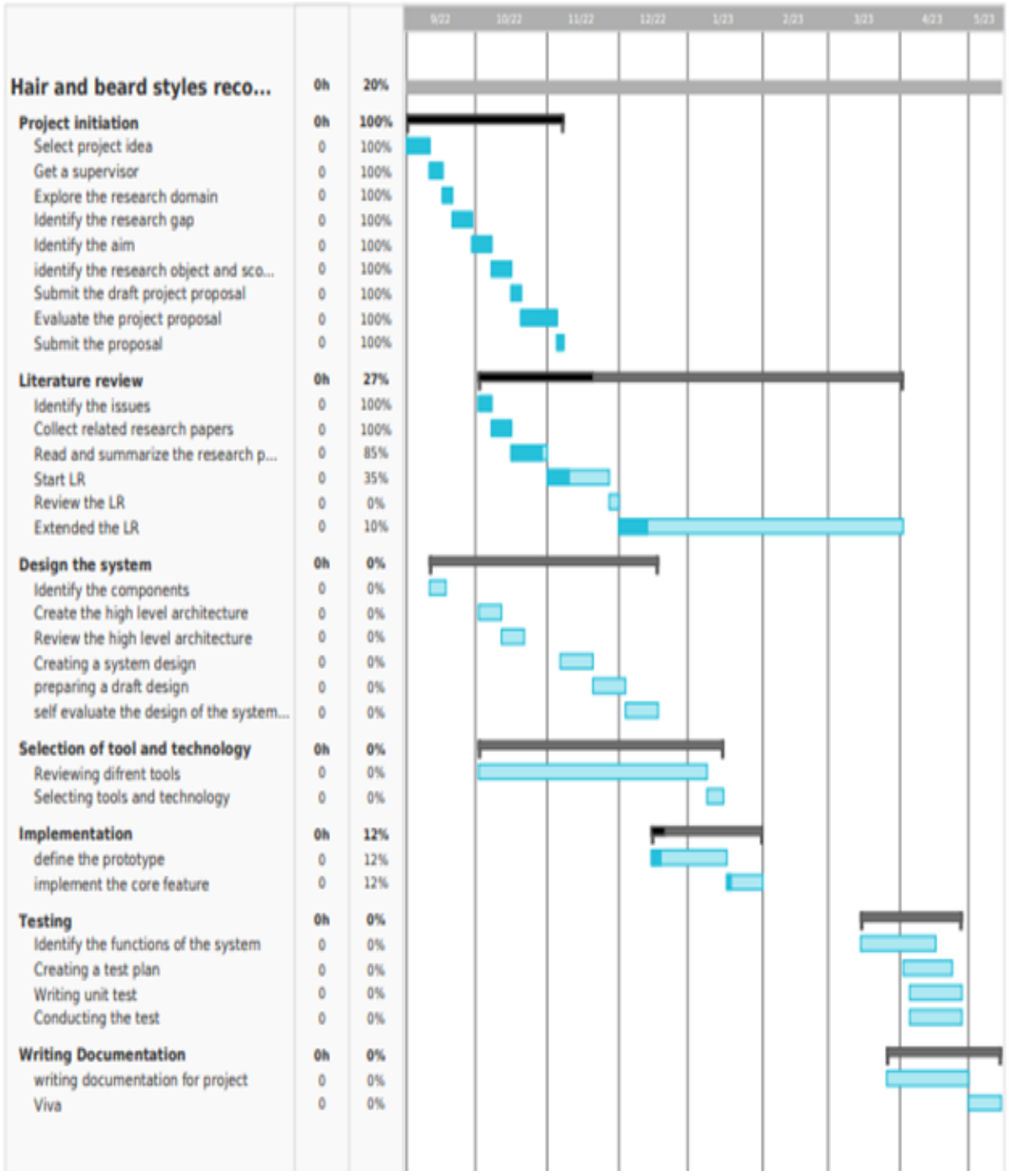


Figure 36 - Gantt Chart

Appendix 03 - UI Wireframes

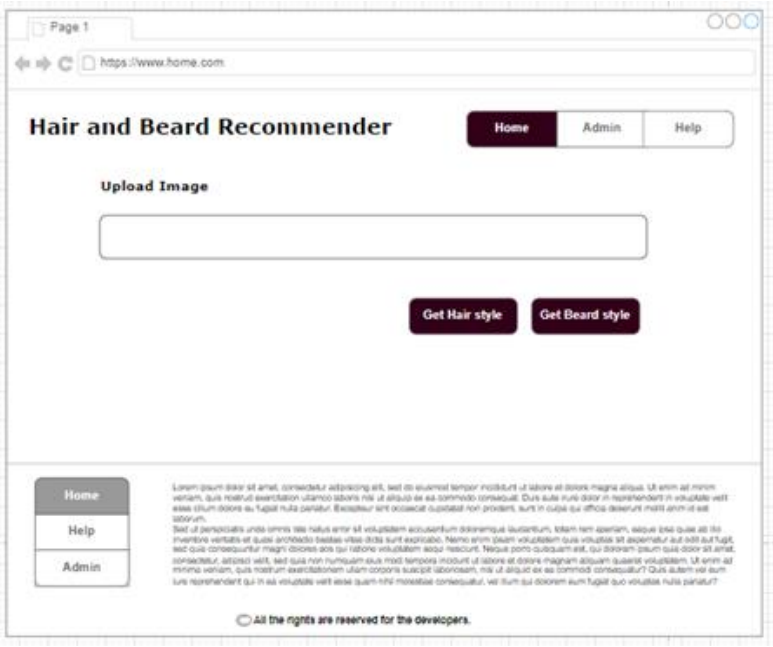


Figure 37 - UI Wireframes 01

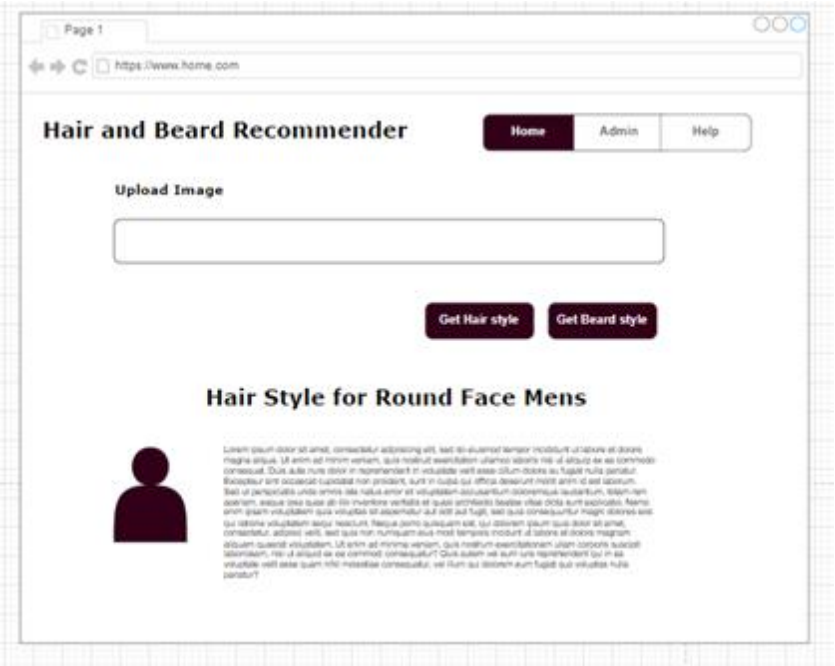


Figure 38 - UI Wireframe 01

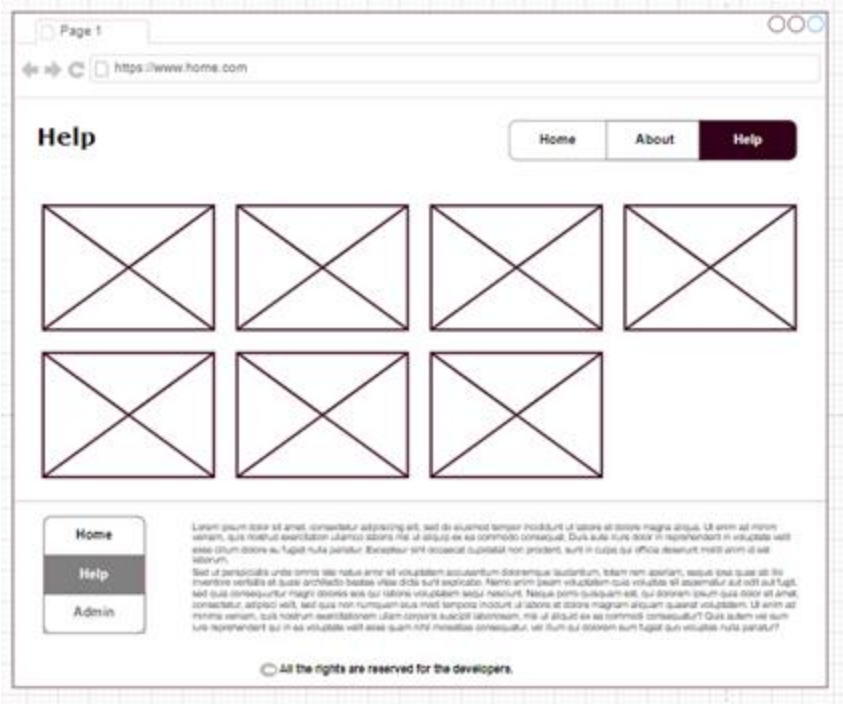


Figure 39 - UI Wireframe 03

Appendix 04 - Component Diagram

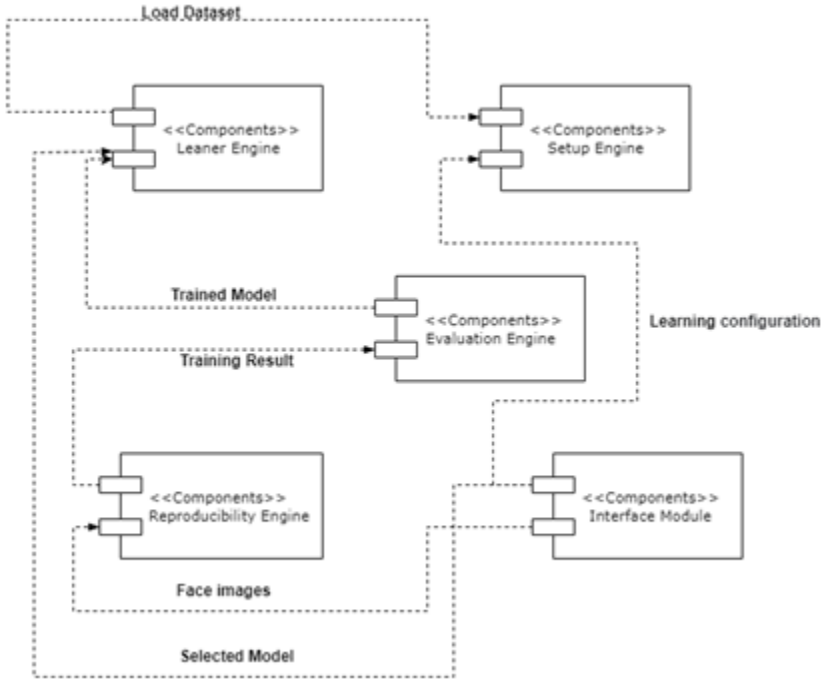


Figure 40 - Component Diagram

Appendix 05 - User Interface

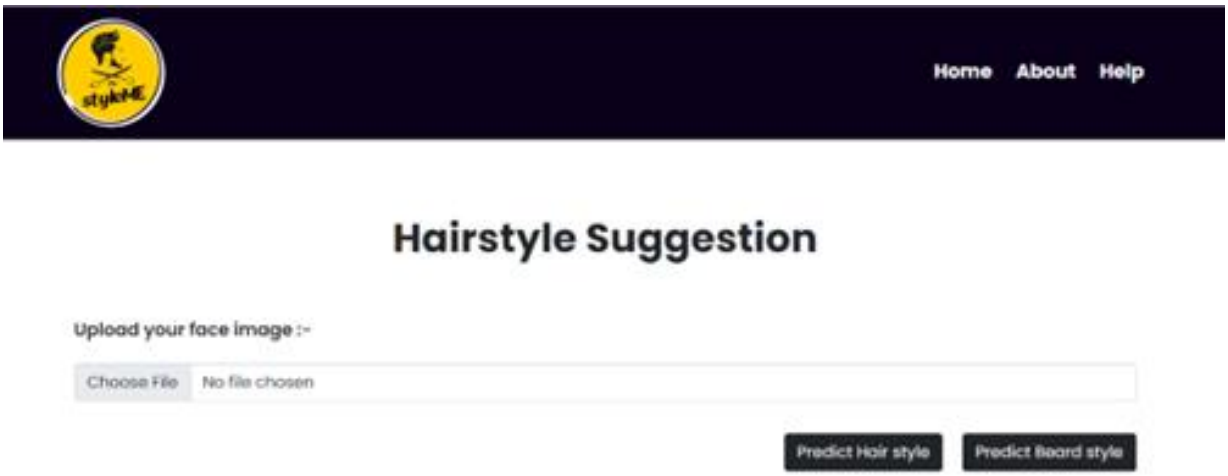


Figure 41 - User Interface 01

Figure 42 - User Interface 02

Men's Hairstyles for Round Faces

The round face shape has a lot of things in common with the square face shape, except for the edgy lines and angles. The round face has an equal dimension in length and width, thus you'll want to make it look slightly longer for added masculinity and sharpness. Wondering how can I make my face look thinner men? This can be achieved with short sides and back and voluminous top. In addition, regard asymmetrical hairstyles, like a comb over or a fringe swept to the side. Also, layering is key to visually narrowing your round face, as it makes the curved cheeks look sharper and more angled. There are two kinds of layers you can incorporate: regular layers cut all over your head and face-framing ones.

01 - The Spiky Short Hair

Figure 43 - User Interface 03

Beardstyles for Round Faces

A round face shape is characterized by fullness in the cheeks and a rounded jawline. The width and length of the face are roughly equal, with a soft and curved jawline. People with round face shapes often have softer facial features, such as a rounded chin and fuller cheeks. When it comes to selecting a beard style for a round face shape, it's important to choose a style that can help to create some balance and definition in the lower half of the face.

01 - The Boxed Beard

the boxed beard can be a great beard style for round face shapes. The boxed beard is a short, neatly trimmed beard that frames the jawline and chin, with a straight bottom edge and defined corners. It can help to add some structure and definition to the lower half

Figure 44 - User Interface 04

Appendix 06 - Evaluation on Functional Requirements

Table 26 - Evaluation on Functional Requirements

FR ID	Requirement	Priority Level	Status
FR1	The user can add the face image to the system, and according to that image, he can get the recommendation.	M	Implemented
FR2	The stylist can add his customer's images to the system and, according to that image, he gets a recommendation.	S	Implemented
FR3	The user can't add an image that contains multiple faces. If the user uploads an image like that, they need to get a proper error message.	C	Implemented
FR4	The user can upload a face image, and if it is not a face image, an error message will appear.	S	Implemented
FR5	If the user uploads a female face image for predicting a beard style system need to throw a proper error message	S	Implemented
FR6	The system can recognize a person's face and use it as input for the model.	C	Not Implemented
FR7	The system can check the input and find the gender of the image.	S	Implemented
FR8	The system can throw an error message if the user's gender is female.	W	Implemented
FR9	The system needs to have a GUI for the user to add an image and show the recommendation.	S	Implemented
FR10	The system needs to throw error for if user upload a not allowed format	C	Implemented

Functional Requirement Completion Percentage = $\frac{09}{10} \times 100$ = 90%

Appendix 07 - Evaluation on Non-Functional Requirements

Table 27 - Evaluation on Non Functional Requirements

NFR ID	Requirement	Description	Priority Level	Status
1	Accuracy	The recommendation algorithms and the model need to have high accuracy.	Important	Implemented
2	Performance	The system needs to generate the suggestion within a short period of time.	Important	Implemented
3	Maintainability	The developer needs to follow the coding standards and best practices when he develops the system.	Desirable	Implemented
4	Usability	The system needs to throw proper error messages when errors occur.	Desirable	Implemented
5	Usability	The system must have a simple GUI that all users can understand, as well as a help page that explains the system to the user.	Important	Implemented
6	Security	The application should prevent any attackers from manipulating results and extracting user inputs. also need to have proper login methods.	Desirable	Implemented - minimal
7	Compatibility	The system needs to support all types of platforms.	Desirable	Implemented

Non-Functional Requirement Completion Percentage = $\frac{07}{07} \times 100 = 100\%$

Appendix 08 - Achievements of Research Objectives

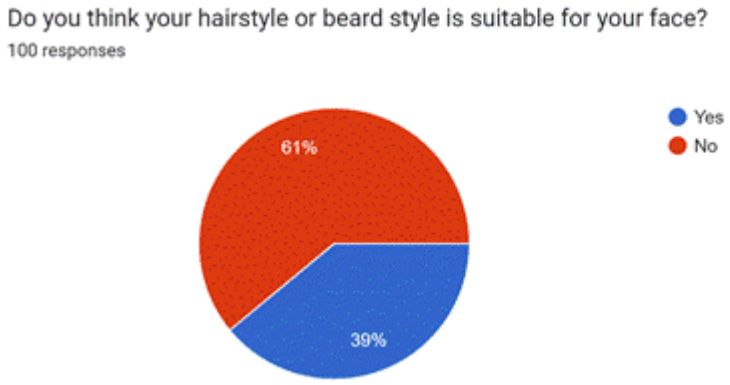
Table 28 - Achievements of Research Objectives

Objective	Description	Learning Outcomes
Literature Survey	Analyze the previews works and critically evaluate them. • **R01:** Conduct a preliminary study on existing Models, Systems & Algorithms. • **R02:** Analyze the hairs and beard style recommendations system technics. • **R03:** Analyze the face shape prediction system and algorithms related to the face shape prediction model	Completed
Design	designing an architecture and a system that can solve the identified problems with the recommended techniques. • **R04**: Design a System which can recommend a hair or beard style to user. • **R05**: Design an image processing model which can predict the users face shape. • **R06**: Design an image processing model which can identify the user's gender.	Completed
Requirement Analysis	Carrying out an in-depth user requirement gathering phase for the following areas • **R07**: Gather the requirements of a Recommendations System and understand end-user expectations. • **R08**: Conduct the user survey and collect details about users' expectations regarding this system. • **R09**: Get the opinions of domine and technology expect regarding the system and understanding the experts' expectations	Completed
Development	Implementing the system that capable to full fill all	Completed

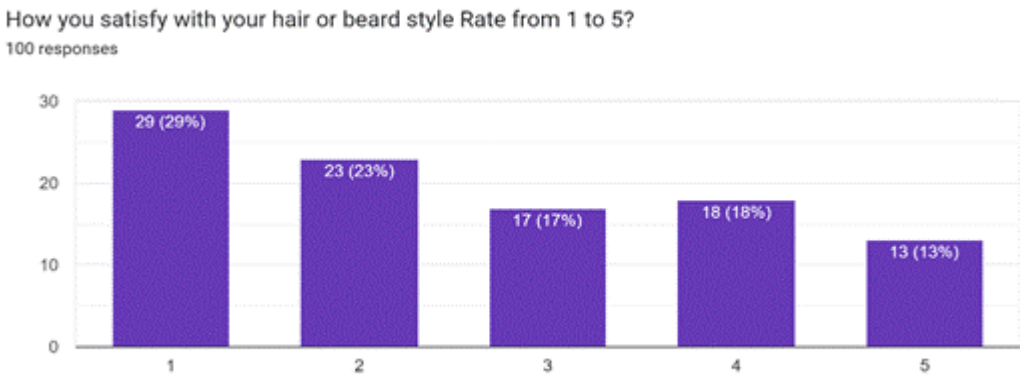
	scopes • **R10**: Develop the recommendation system which can analyst the user face and according to that face shape it will suggest the hair and beard style. • **R11**: develop a graphical user interface which is easy to use for user.	
Testing and Evaluation	Testing the created system & Data science models with appropriate data and evaluating them with baseline techniques identified in the literature. • **R12**: To create test plan and perform various unit, integration and functional testing. • **R13**: Evaluating the CNN model.	Completed

Appendix 09 - Survey

Table 29 - Survey for SRS

Question	**Do you think your hairstyle or beard style is suitable for your face?**
Aim of question	To identify users who were dissatisfied with their hair and beard styles.
Findings & Conclusion  Because the majority of users are dissatisfied with their hair or beard styles, we can conclude that this forum has successfully reached its intended audience.	
Question	**How you satisfy with your hair or beard style Rate from 1 to 5?**
Aim of question	to identify the people who are not satisfied with their hairstyle or beard style.

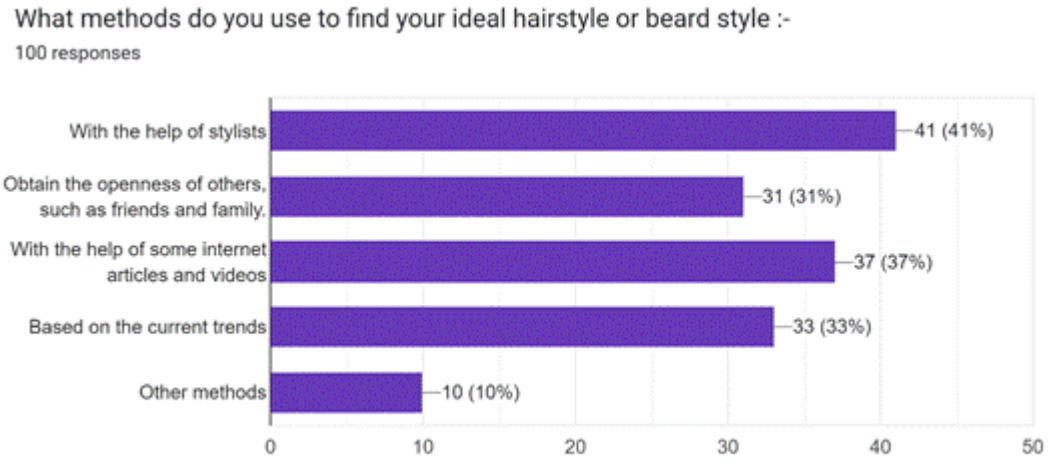
Findings & Conclusion



Among the users, most of them are not satisfied with their hair and beard styles. Twenty-nine percent of the users think their hair or beard style is not fit for them, and more than 40 percent of the users are also not fully satisfied with their hair and beard style. Also, 30 percent of the users are satisfied with their hair and beard styles. Based on the findings, we can conclude that more than 60% of people are not satisfied based on their hair or beard style, so the project's requirements can simply meet the needs of the users and provide a solution for them.

Question	**Do you think you can improve your hair style or beard style?**
Aim of question	To find what methods are peoples are using to find their suitable hair style

Findings & Conclusion

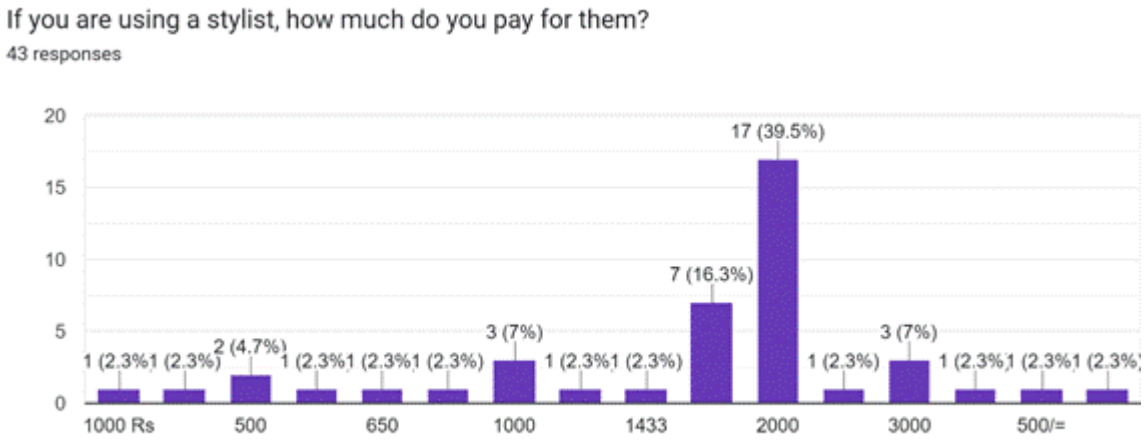


According to the findings, more than 40 percent of people still rely on stylists to choose their hair and beard styles, and around 30 percent choose their hair and beard styles based on current trends. In

addition, 37 people used the internet to choose their hair and beard styles, and 31 people sought advice from others. So based on the result, most of the people are using the old way to find suitable hair and beard styles, so if the author suggests new methods like a online system based on the image processing model, it will reach most of the people.

Question	**If you select other methods, please mention those down?**
Aim of question	To learn about the other methods users, use to find a hair and beard style that suits them,
Findings & Conclusion	Most users use the above options, so the author didn't get many answers for this question, but one person said he followed the hairstyles of celebrities like famous cricketers and cinema actors, and some said they selected hairstyles based on their favorite movie characters, but the author can't say if it's suitable for them or not, so it's better to go with some professional methods.
Question	**If you are using a stylist, how much do you pay for them?**
Aim of question	To find how much peoples are spending for find a suitable hair or beard style for them

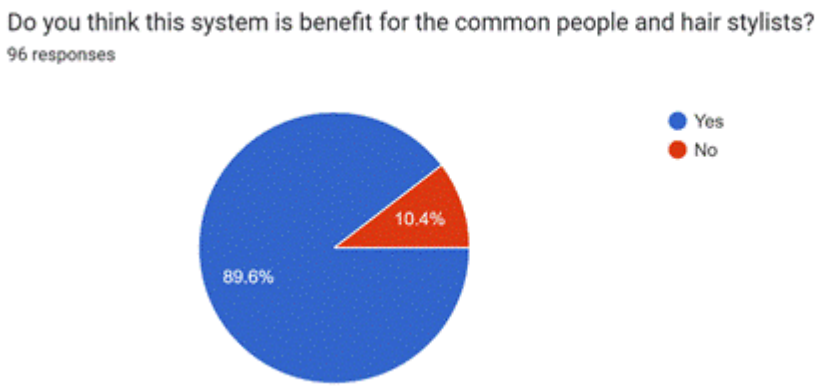
Findings & Conclusion



Based on the answers, around 17 percent of the people are spending 2000 for their hair and beard style, and most of the people are spending from 500 rupees to 3000 rupees, so the authors' system can help them reduce these costs.

Question	**Do you think this system is benefit for the common people and hair stylists?**
Aim of question	To check whether the proposed system is useful for both common users and professionals like stylists

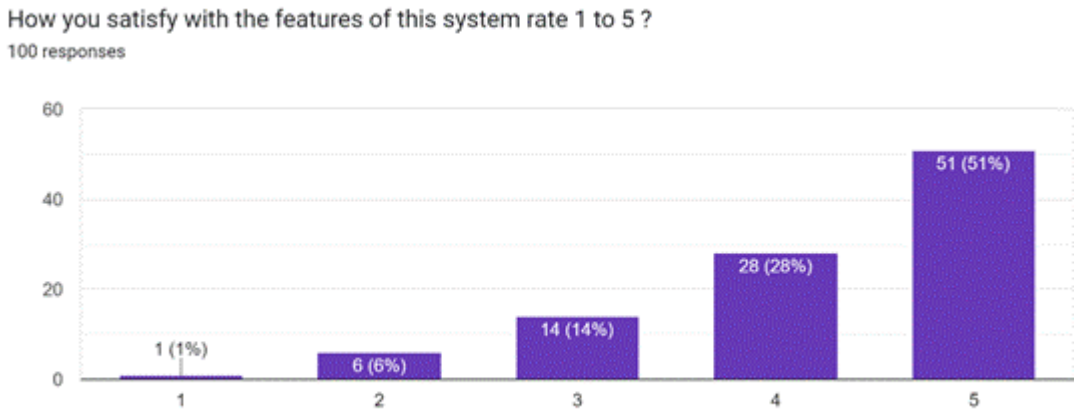
Findings & Conclusion



Most of the participants accept that the system is beneficial for both common people and professionals; almost 90 percent of the participants are happy with the system, so based on this result, we can conclude that developing a system like this is useful for society.

Question	**How you satisfy with the features of this system rate 1 to 5?**
Aim of question	To check how much the user can satisfy the requirements of the system

Findings & Conclusion



Based on the result, almost 80 percent of the users are fully satisfied with the requirements of the system, and 14 percent of the users have mixed reviews about the project requirements, so the above result shows that most of the participants are happy and fully satisfied with the requirements of the system. Based on these findings, if we fully fulfill these requirements and create the system, it will satisfy all types of users.

Appendix 10 - Research Methodology

Table 30 - Research Methodology

Research Methodology	
Research Philosophy	The research philosophy is primarily concerned with the origin, nature, and development of knowledge (Snyder, 2019), or, to put it another way, the methods by which data is collected, analyzed, and used. Among the possible candidates of pragmatism, positivism and interpretivism, pragmatism approach was chosen because the author going to research and try to implement various methods in a combined approach to find the best approach.
Research Approach	The approach of the research deals with the relevance of hypotheses to the study and methods of data collection and interpretation. We can divide the research approach in two major parts one is deductive and another one is Inductive. Among these two-methods author decided to flow the deductive approach because from this aim author going to implement the combination of the existing theories with simple modifications.
Research Strategy	The research strategy defines how the researcher is going to collect the data for the research. Among the main candidates of questionnaire, interviews, experiments, literature, observation and brainstorming author selected the questionnaire, interviews and experiments.
Research Choice	Choice is the fourth layer in the onion model. It will indicate whether we need to use quantitative, qualitative or both methodologies. Among the candidates of mono, mixed, and multi methods, the mixed method was chosen because in this method we can use the combine methodology of quantitative and qualitative methods. It will used for the surveys, interviews and experiments.
Time Horizon	The time horizon will describe the required time which need to complete the research. Among the candidates of

	longitudinal and cross-sectional, cross-sectional method was chosen.
Techniques and procedures	For collecting the data, mainly the organizational dataset, will be used. The questioners, interviews, and brainstorming sessions are also used to collect data

Appendix 11 - Risk management

Table 31 - Risk management

Risk	Probability of Occurrence	Magnitude of the loss	Mitigation Plan
Lack of knowledge in domain	2	5	Get the assistance of domain experts and developers and follow online courses and tutorials.
Lost the developed code	3	5	Save all the code in Git on a daily basis
Documents have the potential to be corrupted.	3	4	Save the copy of document in Google drive
Can't submit the deliverables within a deadline	3	5	Work hard to finish the deliverable within the deadlines and have the weekly goals

Appendix 12 - Deliverable

Table 32 - Deliverable

Deliverable	Date
Project proposal document The initial proposal document regarding the proposed project.	03rd November 2022
Ethics forms elaborate about the ethics related to the project	03rd November 2022
Software requirement specification document The document which will specify the requirements that are necessary to	24th November 2022

develop the prototype.	
System Design Document The document specifying the architecture of the proposed prediction system.	05th December 2022
Presentation of implementation and proof of concept	23rd December 2022
Initial project specifications design and prototype document	23rd February 2023
Test and evaluation report The document which will highlight the evaluations and tests that were carried out.	23rd March 2023
Draft Project Report The draft version of the thesis which will be submitted to the supervisor to get the feedback.	30th March 2023
Prototype The main working prototype of the proposed system.	
Final Thesis The final report which will discuss about all the findings, research and decisions.	24th April 2023

Appendix 13 - Functional Requirements

Table 33 - Functional Requirements

FR ID	Requirement	Priority Level
FR1	The user can add the face image to the system, and according to that image, he can get the recommendation.	M
FR2	The stylist can add his customer's images to the system and, according to that image, he gets a recommendation.	S
FR3	The user can't add an image that contains multiple faces. If the user uploads an image like that, they need to get a proper error message.	C
FR4	The user can upload a face image, and if it is not a face image, an error message will appear.	S
FR5	If the user uploads a female face image for predicting a beard style system need to throw a proper error message	S
FR6	The system can recognize a person's face and use it as input for the	C

	model.	
FR7	The system can check the input and find the gender of the image.	S
FR8	The system can throw an error message if the user's gender is female.	W
FR9	The system needs to have a GUI for the user to add an image and show the recommendation.	S
FR10	The system needs to have a GUI for the admin to manage the system	C

Appendix 14 - Implementing the Gender Identification Model

The below code shows a series of import statements from TensorFlow and its Keras API. Those are TensorFlow itself, EfficientNetV2M model from the Keras API's applications module for training the transfer learning model. ModelCheckpoint, ReduceLROnPlateau, and EarlyStopping functions. Dense and GlobalAveragePooling2D layers, which will be used to build the model.

```
import tensorflow as tf
from tensorflow.keras.applications import EfficientNetV2M
from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, EarlyStopping
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Figure 45 - Importing Libraries

In the code author provided, EfficientNetV2-M is used as the base model for transfer learning. The pre-trained weights of the EfficientNetV2-M model are used as the initial weights for the new model, and the last few layers of the model are replaced with a new dense layer for the specific image classification task. The model is then trained on the dataset using an image data generator and a set of call backs for monitoring the training process.

```
# Load pre-trained EfficientNetV2M model
base_model = EfficientNetV2M(input_shape=(input_size[0], input_size[1], 3), include_top=False, weights='imagenet')
```

```
# Add global average pooling and dense output layers
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(256, activation='relu')(x) # Add additional dense layer for feature extraction
predictions = Dense(2, activation='softmax')(x) # Output layer with 5 classes for face shapes
```

```
# Create the model
model = Model(inputs=base_model.input, outputs=predictions)
```

```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```
[ ] # Defining model file name
    filePath ='EfficientNetV2-M'+ '_BestModel.h5'

    # Defining model checkpoint for saving best model
    bestModelCheckpoint = ModelCheckpoint(filePath, monitor='val_loss', verbose=1, save_best_only=True, save_weights_only=False, mode='auto', period=1)

    # Defining decaying learning rate callback
    decayLR = ReduceLROnPlateau(monitor='val_loss', factor=0.2, verbose=1, patience=3)

    # Defining early stopping callback
    earlystopCheckpoint = EarlyStopping(monitor="val_loss", min_delta=0, patience=9, verbose=1)

    # Preparing list of all callbacks
    callbacksList = [bestModelCheckpoint, decayLR, earlystopCheckpoint]
```

Figure 46 - Code for Training

The below code is using the model.fit() function to train the gender identification model.

```
# Train the model with the callback
model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    epochs=100, # Number of training epochs
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // validation_generator.batch_size,
    callbacks=[callbacksList] # Add the callback to the training process
)
```

Figure 47 - Code for Using Fit method.

Appendix 15 - Gender Identification Model Testing

The gender prediction model is also an image classification model, so the same testing methods used for the face shape prediction model were used. for this model training accuracy is 99% and the validation accuracy is 97% which indicates this model is not facing any over fitting issues. Also, the accuracy also presented below.

```
scores = modelMy.evaluate(test_generator, steps=test_generator.samples // test_generator.batch_size + 1)
```

```
11/11 [==============================] - 104s 9s/step - loss: 0.0614 - accuracy: 0.9765
```

Figure 48 - Evaluation

The gender identification model achieved an accuracy of 97.6%, which is considered high for this type of model. This performance is notably better than other existing models. The high accuracy of the model indicates that it can effectively classify the gender of users based on the input image. Author used the classification_report method and generate the report wich is presented below:-

	precision	recall	f1-score	support
Men	0.97	0.98	0.98	170
Women	0.98	0.97	0.98	170
accuracy			0.98	340
macro avg	0.98	0.98	0.98	340
weighted avg	0.98	0.98	0.98	340

Figure 49 - Classification Report

The report indicates the precision is 0.98, recall is 0.98, fi-score is 0.98 and overall accuracy is 98% which is indicating this model have better performance.

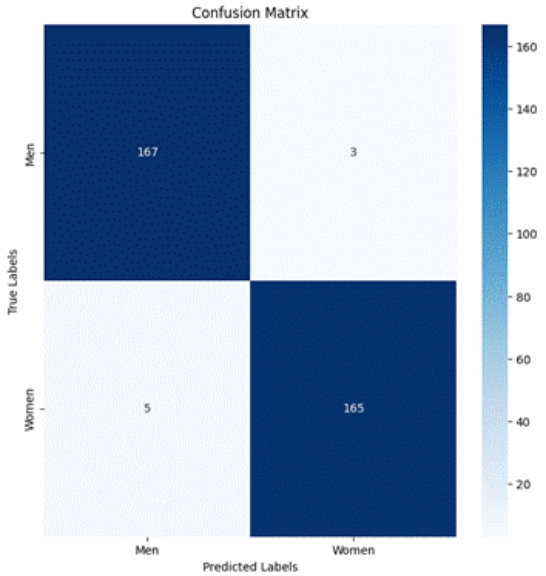


Figure 50 - Matrix

Once again, the confusion Metrix indicate that the models have a better performance.in here for both classes we used 170 images and amount those 170 images model predicted 165 images correctly. Also, the roc curve of this model also presented below.

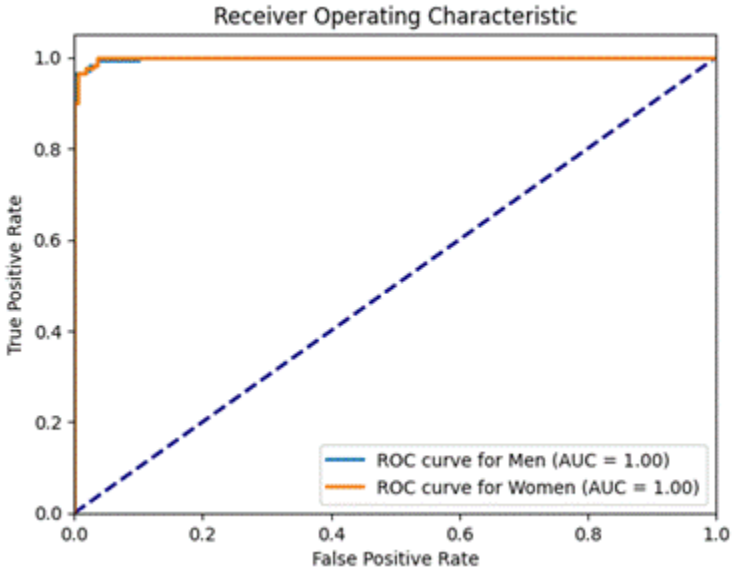


Figure 51 - Roc Curve

Appendix 16 - Achievement of Learning Outcomes

Table 34 - Achievement of Learning Outcomes

Learned Skills	Learning outcomes
Literature Review- A thorough study of the literature on research components and technologies	LO1, LO2, LO4, LO5, LO8
Requirement Analysis - Performing in-depth requirement gathering phase utilizing several techniques.	LO1, LO2, LO8
System Design and Architecture - Perform the required system design and prepare a system design specification.	LO1, LO2, LO5, LO8
Implementation - Develop the system with the help of factors identified with the requirement analysis and design phases.	LO5, LO8

Testing and Evaluation- Testing and evaluating the prototype to make sure the system is working fine	LO4, LO5, LO7, LO8

Appendix 17 - Concept Map

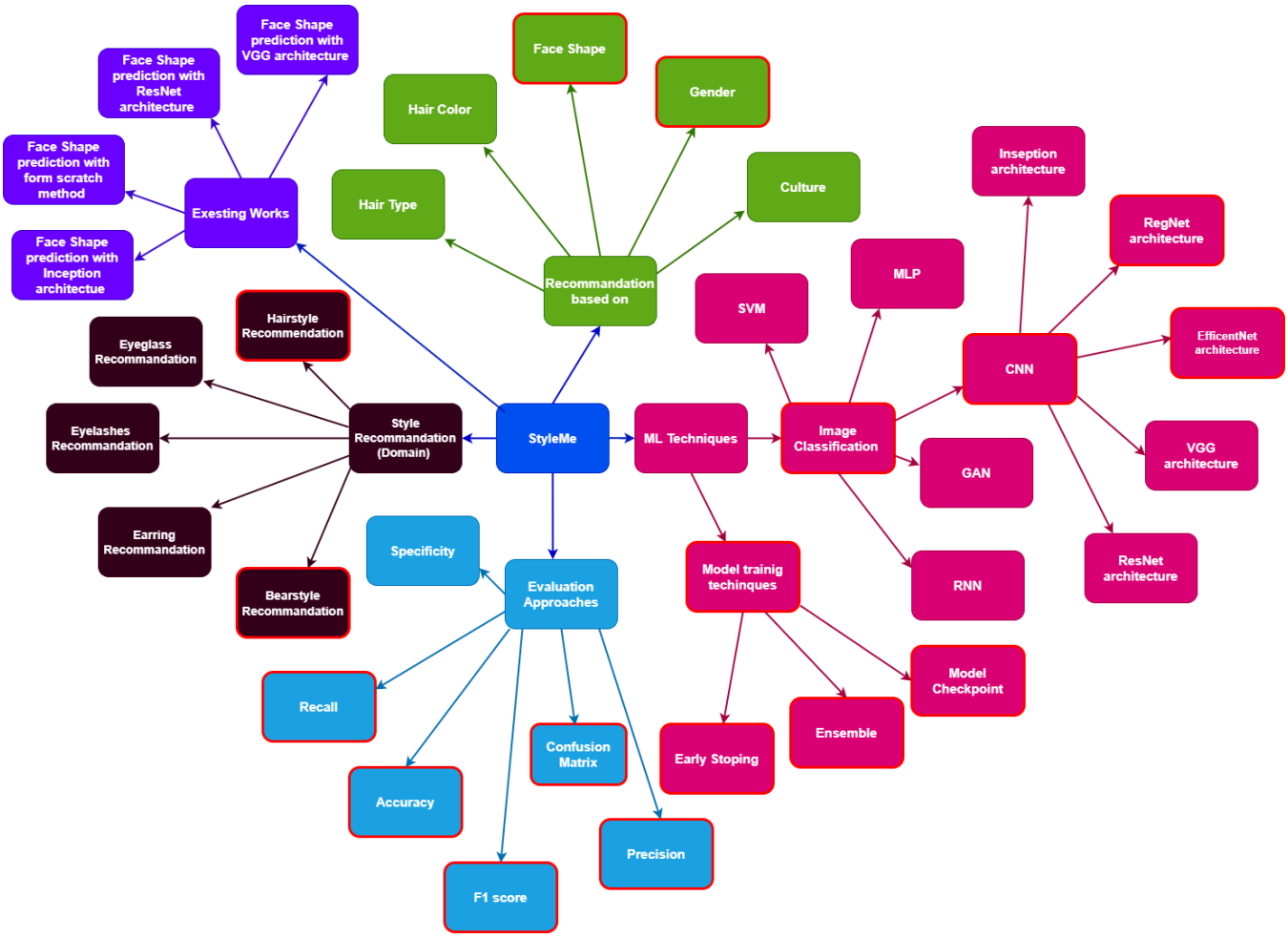


Figure 52 - Concept Map

Appendix 18 – Implementation of the System

This part provides the important code segments of the StyleMe codes (not every).

faceShape.py

```
# Loading a face prediction model
face_shape_predictor = load_model('models/face_shape_classification_model_Ensemble_Version_03.h5')

# List of face shapes
face_shape = ['Heart', 'Oblong', 'Oval', 'Round', 'Square']

# This method used to predict the face shape and return it
def predict_face_shape():

    # In here we are loading image and pre procesing
    image = load_img("static/images/preprocessed/preprocessed_image2.jpg", target_size=(224, 224))
    image = img_to_array(image)
    # Normalize pixel values
    image = image / 255.0
    image = np.expand_dims(image, axis=0)

    # Predicting the face shape
    predictions = face_shape_predictor.predict(image)
    index = np.argmax(predictions, axis=1)
    return index[0]
```

Figure 53 - faceShape.py

This file contains a code for predicting a face shape for the system. In here first author used the load_model method to loading the face shape prediction model. Then he created a predict_face_shape method which will return the face shape of the user. In here he loads the images and normalize is it and using the predict method to predict the face shape and returning it to the system.

gender.py

This file contains the code for identifying the gender of the user. First author loading the MTCNN model to find the faces from the images. Then he is loading the gender identification model using load_model method. This file contains one method named get_gender which will return the gender of the user. In this method the first author gets the image from the front end and cropping the head using the MTCNN model. Then normalized that image and predicting the gender of the user and return it to the system.

```python
# Loading the pre-trained face detector model
face_detector = MTCNN()

# Loading the gender identification model
gender_detector = load_model('models/gender_classification_model_EfficientNetV2M_Version_01.h5')

# This method used to identify the gender and return it.
def get_gender(faceImage):
    # This code used to detect the face and crop the head.
    face = face_detector.detect_faces(faceImage)
    for result in face:
        x, y, width, height = result['box']

        head_width = int(1.2 * width)
        head_height = int(1.6 * height)
        head_x = int(x - 0.1 * width)
        head_y = int(y - 0.5 * height)

        head = faceImage[max(head_y, 0):head_y+head_height, max(head_x, 0):head_x+head_width]

        # Saving the processed image.
        cv2.imwrite("static/images/preprocessed/preprocessed_image2.jpg", head)

    # In here we are loading image and pre procesing.
    image = load_img("static/images/preprocessed/preprocessed_image2.jpg", target_size=(224, 224))
    image = img_to_array(image)
    # Normalize pixel values
    image = image / 255.0
    image = np.expand_dims(image, axis=0)

    # Predicting the gender
    predictions = gender_detector.predict(image)
    index = np.argmax(predictions, axis=1)
    return index[0]
```

Figure 54 - gender.py

otherMethod.py

```python
# Loading the pre-trained face detector model
face_detector = MTCNN()

# List of allowed image extension
ALLOWED_EXTENSIONS = {'jpg', 'jpeg', 'png', 'gif'}


def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

# This method used to check the face count of the input image
def get_face_count(img):
    rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    faces = face_detector.detect_faces(rgb)
    if not faces:
        face_count = 0
    else:
        face_count = len(faces)
    return face_count
```

Figure 55 - otherMethod.py

This file contains the other methods of the system. This method contains two methods one of that is allowed_file which use to check the file formats and the get_face_count is used to return the faces in the user inputted images using the MTCNN model.

app.py

```
@app.route("/")
@app.route("/home")
def home():
    return render_template('home.html')

#open help page
@app.route('/help')
def help():
    return render_template('help.html')

#open about page
@app.route('/about')
def about():
    return render_template('about.html')
```

Figure 56 - Code for Routing

The above code is showing the routes of the StyleMe system which used to navigate the webpages. The blow code is responsible for getting user image from the front end return the hair or beard style suggestion based on that image. First author gets the image from the post request and checks if the file is in an allowed image format if not system return a error message. Then it will check the faces in that image and if the image count is not one then it will return the error. After that getting the gender based on that image and if user clicked the predict beard style button and the gender is women it through errors. Otherwise, it predict the face shape and based on the face shape and gender it retrieve the style and return to the front end like wise hair style also work in a same way.

```python
# Api for suggesting hair and beard style
@app.route('/predict', methods=['POST'])
def predict():
    if request.method == 'POST':

        # Get the uploaded file from the form data
        file = request.files['file']

        # Check the file format and if it not allowed it will return error
        if not allowed_file(file.filename):
            return render_template('home.html', message ='Invalid file format. Please upload an image file (jpg, jpeg, png, gif).')

        # Convert file data to a numpy array
        file_data = np.fromstring(file.read(), np.uint8)

        # Decode the image data to a Mat object
        img = cv2.imdecode(file_data, cv2.IMREAD_COLOR)
        cv2.imwrite("static/images/preprocessed/preprocessed_image.jpg", img)

        # Check if file has allowed extension
        if not allowed_file(file.filename):
            return render_template('home.html', message ='Invalid file format. Please upload an image file (jpg, jpeg, png, gif).')

        #getting face count for use the prediction
        faceCount = get_face_count(img)

        # Base on the face count show error messages
        if(faceCount == 0):
            return render_template('home.html', message = "No faces detected in the image.")
        elif(faceCount > 1):
            return render_template('home.html', message = "There are many faces detected! please add a image which contain one face")
        else:
            # Getting the user gender
            genderIdx = get_gender(img)

            # Checking if the user click the predict face shape button
            if request.form['action'] == "task1":
                # Getting the face shape
                faceShape_idx  = predict_face_shape()
                # Getting the hair style based on the users gender and face shape
                data = hair_style_detail[genderIdx][faceShape_idx]
                return render_template('home.html', prediction = data)
            else:
                # Checking if the user is male or not
                if genderIdx  == 0:
                    # Getting the face shape
                    faceShape_idx  = predict_face_shape()
                    # Getting the beard style based on the users face shape
                    data = beard_style_detail[faceShape_idx]
                    return render_template('home.html', prediction = data )
                else:
                    return render_template('home.html', message = "Please add men's images to suggest a suitable beard style.")
```

Figure 57 - Suggest Hair and Bear Style Method.

Appendix 19 - Data Flow Diagram

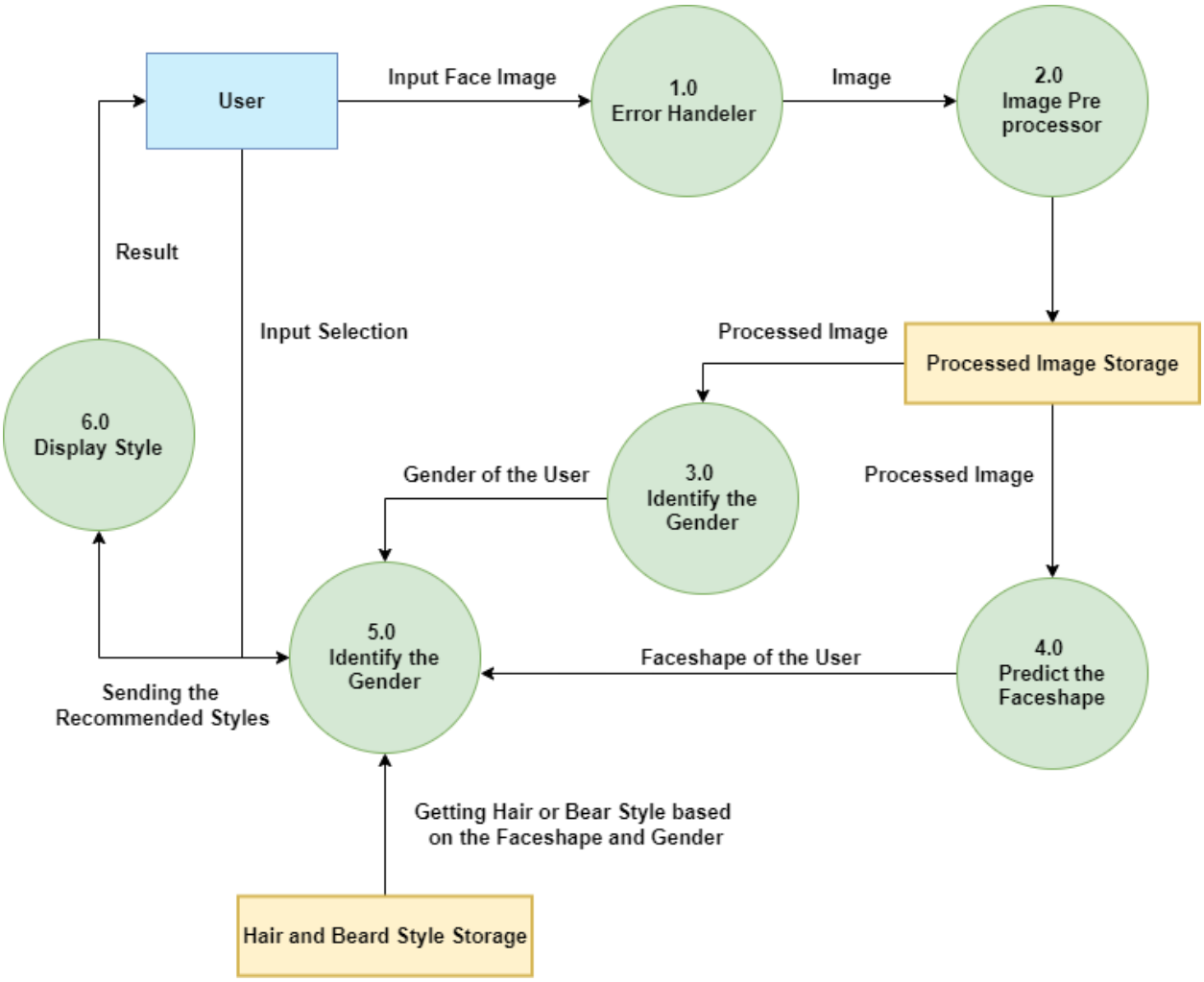


Figure 58 - Data Flow Diagram