UNIVERSITI TEKNOLOGI MARA

VISIONFIT: AI EYEWEAR TRY-ON ASSISTANT USING DEEP LEARNING APPROACH

MUHAMMAD RAFIQ AZIZI BIN ROSLAN

BACHELOR OF INFORMATION SYSTEMS (HONS.)
INTELLIGENT SYSTEMS ENGINEERING

JANUARY 2026

UNIVERSITI TEKNOLOGI MARA

VISIONFIT: AI EYEWEAR TRY-ON ASSISTANT USING DEEP LEARNING APPROACH

MUHAMMAD RAFIQ AZIZI BIN ROSLAN

Thesis submitted in fulfilment of the Requirements for Bachelor of Information Systems (Hons.) Intelligent Systems Engineering

Faculty of Computer and Mathematical Sciences

JANUARY 2026

SUPERVISOR APPROVAL

VISIONFIT: AI EYEWEAR TRY-ON ASSISTANT USING DEEP LEARNING APPROACH

By

MUHAMMAD RAFIQ AZIZI BIN ROSLAN

2024767483

This thesis was prepared under the supervision of the project supervisor, Dr. Mohd Zaki Zakaria. It was submitted to the Faculty of Computer and Mathematical Sciences and was accepted in partial fulfilment of requirements for the degree of Bachelor of Information Systems (Hons.) Intelligent Systems Engineering

Approved by

DR. MOHD ZAKI ZAKARIA

PROJECT SUPERVISOR

JANUARY 2026

STUDENT DECLARATION

I certify that this thesis and the project to which it refers is the product of my own work and that any idea or quotation from the work of other people, published or otherwise are fully acknowledged in accordance with the standard referring practices of the discipline.

MUHAMMAD RAFIQ AZIZI BIN ROSLAN

2024767483

JANUARY 2026

ACKNOWLEDGEMENT

First and foremost, I would like to sincerely thank Allah SWT for giving me the courage, perseverance and wisdom I needed to complete this final year project for my Bachelor of Information Systems (Hons.) in Intelligent Systems Engineering.

I want to express my deep thanks to my final year project supervisor which is Dr. Mohd Zaki Zakaria for all of his help, criticism and constant support throughout this project. His assistance and knowledge have been extremely helpful in determining the course and outcome of my project.

Additionally, Dr. Azlin Binti Ahmad that instructed and managed the CSP650 course has my sincere appreciation. I have been able to successfully manage the requirements and structure of this project thanks to her clear instructions, patience and support.

I also want to deliver my sincere gratitude to my examiner which is Dr. Farah Aqilah Bohani for taking the time to carefully review my work. Her suggestions and comments have greatly helped in this project development and improvement.

Last but not least, I would like to express my gratitude to my family, friends and everyone else who has helped me along the way whether directly or indirectly. I am so grateful for their understanding and support.

ABSTRACT

This project's goal was to develop an intelligent virtual eyewear try-on system that intended to help people choose eyewear more wisely. The standard approach of selecting eyewear was manual and trial and error when often the selection procedure is subjective, time consuming and unreliable. The already available virtual try-on tools do not have facial geometry analysis and therefore it gives lack or generic suggestions. To resolve this issue, this project integrated WebAR real-time 3D visualization, rule-based eyeglass mapping and facial shape classification into a single lightweight web application. This project used 5000 labelled photos of five different face shape groups from the publicly available Hugging Face dataset. VGG16 was used to make classification for the model. Various preprocessing steps were applied to the dataset such as face detection, resizing, maintaining aspect ratios and data augmentation. The creation of rule-mapping that associated all facial forms with suitable eyewear design based on aesthetic considerations and expert opinion. AR.js and A-Frame provided the basis for user interaction WebAR which overlays 3D eyeglass models on the user's face in real time. The model achieved an accuracy of 87% when classifying facials into five face shape categories. The AR try-on was responsive and visually clear but some of the frames required manual scaling. In conclusion, this project was able to offer an efficient solution that combines WebAR visualization, structured suggestions and AI classification. Future advancements might include automatic scaling of 3D frames, improved AR stability, expanding the data collection and testing the system on the real optical retail environment.

TABLE OF CONTENTS

CONTENT **PAGE**

SUPERVISOR APPROVAL	ii
STUDENT DECLARATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	ix
LIST OF TABLES	xi
LIST OF ABBREVIATIONS	xii
CHAPTER ONE	1
INTRODUCTION	1
1.1 Background of Study	1
1.2 Problem Statement	3
1.3 Research Question	4
1.4 Research Objectives	5
1.5 Research Scope	5
1.6 Research Significance	6
1.7 Summary	7
CHAPTER TWO	8
LITERATURE REVIEW	8
2.1 The Human Face and Face Shape Classification	8
2.1.1 Face Shape Definition and Categories	8
2.1.2 Feature Extraction Techniques	10
2.1.3 Classification Approaches	11
2.2 Face Detection and Landmark Extraction	15
2.2.1 Overview of Face Detection Techniques	15

2.2.2 Facial Landmark Detection Approaches 20
2.3 Eyewear and Facial Fitting 25
2.3.1 Significance of Proper Frame Selection 25
2.4 Eyewear Recommendation System 27
2.4.1 Rule-Based Recommendation Approaches 27
2.5 Augmented Reality (AR) in Virtual Eyewear Try-On 28
2.5.1 Overview to Augmented Reality 28
2.5.2 AR Tools and Technologies 30
2.5.3 AR Application in Virtual Eyewear Try-On 35
2.6 Related Work and Comparative Studies 38
2.6.1 Commercial Applications of AR Eyewear Systems 38
2.6.2 Comparative Analysis of Face Detection Methods, Face Shape Classification, AR and Recommendation Techniques 39
2.7 Summary 41

CHAPTER THREE 42
RESEARCH METHODOLOGY 42
3.1 Research Methodology Framework 42
3.2 Preliminary Studies 46
3.3 Knowledge and Data Acquisition 46
3.4 Exploratory Data Analysis and Image Preprocessing 47
3.4.1 Exploratory Data Analysis (EDA) 48
3.4.2 Image Preprocessing 50
3.5 Model Development 52
3.5.1 Dataset Preparation 52
3.5.2 Dataset Preparation 53
3.5.3 Model Configuration 53
3.6 Model Evaluation 54
3.7 Rule Mapping as Inference Engine 54
3.7.1 Design of Rule Mapping 54
3.7.2 Implementation 56
3.8 WebAR Development 57
3.8.1 3D Eyewear Model Preparation 57
3.8.2 Face Tracking and Model Alignment 58
3.8.3 Rendering and Visualization 59
3.9 Prototyping 59

3.9.1 System Workflow 60
3.9.2 System Directory Structure 62
3.10 Documentation 63
3.11 Summary 63

CHAPTER FOUR 64
RESULT AND ANALYSIS 64
4.1 Face Shape Classification Experiments 64
4.2 Face Shape Prediction and Analysis 81
4.3 Rule-Based Eyewear Recommendation Results 82
4.4 WebAR Try-On Results 83
4.5 System Integration Results 85
4.6 Summary 89

CHAPTER FIVE 90
CONCLUSION AND RECOMMENDATION 90
5.1 Research Objective Achievement 90
5.1.1 Face Shape Classification Using VGG16 90
5.1.2 Rule-Based Eyewear Recommendation System 91
5.1.3 WebAR Virtual Try-On Implementation 91
5.2 Project Limitation 92
5.3 Future Work 92
5.4 Summary 93
REFERENCES 94

LIST OF FIGURES

FIGURE	TITLE	PAGE
3.1	Research Framework	43
3.2	Distribution Dataset Training	48
3.3	Distribution Dataset Testing	48
3.4	Distribution of Image Height and Width	49
3.5	Distribution of Image Aspect Ratio	49
3.6	Image Before Crop	51
3.7	Image After Crop	51
3.8	Augmentation Image	51
3.9	Eyewear with Face Shape	56
3.10	Snippet Illustrates the Structure	57
3.11	Snippet for AR Rendering Scene	59
3.12	Prototype Workflow	61
4.1	Training & Testing Loss Experiment 1	65
4.2	Training & Testing Accuracy Experiment 1	65
4.3	Confusion Matrix Experiment 1	65
4.4	Training & Testing Loss Experiment 2	66
4.5	Training & Testing Accuracy Experiment 2	66
4.6	Confusion Matrix Experiment 2	66
4.7	Training & Testing Loss Experiment 3	67
4.8	Training & Testing Accuracy Experiment 3	67
4.9	Confusion Matrix Experiment 3	67
4.10	Training & Testing Loss Experiment 4	68
4.11	Training & Testing Accuracy Experiment 4	68
4.12	Confusion Matrix Experiment 4	68
4.13	Training & Testing Loss Experiment 5	69
4.14	Training & Testing Accuracy Experiment 5	69
4.15	Confusion Matrix Experiment 5	69
4.16	Training & Testing Loss Experiment 6	70
4.17	Training & Testing Accuracy Experiment 6	70

4.18	Confusion Matrix Experiment 6	70
4.19	Training & Testing Loss Experiment 7	71
4.20	Training & Testing Accuracy Experiment 7	71
4.21	Confusion Matrix Experiment 7	72
4.22	Training & Testing Loss Experiment 8	72
4.23	Training & Testing Accuracy Experiment 8	72
4.24	Confusion Matrix Experiment 8	73
4.25	Training & Testing Loss Experiment 9	73
4.26	Training & Testing Accuracy Experiment 9	73
4.27	Confusion Matrix Experiment 9	74
4.28	Accuracy, Precision, Recall, F1-Score Baseline Model	76
4.29	Optimizer Effect	77
4.30	Epoch Effect	78
4.31	Batch Size Effect	80
4.32	Random Image Test	81
4.33	Result Rule-Based	83
4.34	Result Rule-Based on AR	84
4.35	Prototype Index Interface	85
4.36	Try-On Interface	86
4.37	Result Interface When Face Exist	86
4.38	Result Interface When Face Non-Exist	87
4.39	AR Interface	88
4.40	Prototype Description	88

LIST OF TABLES

TABLE	TITLE	PAGE
2.1	Comparison of Classification Methods	14
2.2	Comparison CNN Architecture Accuracy	17
2.3	Comparison of Face Detection Techniques	18
2.4	Comparison Tools and Libraries	23
2.5	Comparison AR Tools and Technologies	34
2.6	Comparative Academic Research	39
3.1	Research Methodology	44
3.2	Summary of Eyewear with Face Shape	55
3.3	System Directory in src Folder	62
3.4	Details of Documentation Phase	63
4.1	Experimental Setup	64
4.2	Metric Table Presentation	75

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AR	Augmented Reality
CNN	Convolutional Neural Network
SVM	Support Vector Machine
HOG	Histogram of Oriented Gradients
VGG	Visual Geometry Group
RF	Random Forest
KNN	K-Nearest Neighbours
SDG	Sustainable Development Goals
SDK	Software Development Kit
SLAM	Simultaneous Localization and Mapping
API	Application Programming Interface
UI	User Interface
MSE	Mean Squared Error
NME	Normalized Mean Error
MAFL	Multi-Attribute Facial Landmark
AFLW	Annotated Facial Landmarks in the Wild
ML	Machine Learning
AR.js	Augmented Reality JavaScript framework
WebRTC	Web Real-Time Communication
IDE	Integrated Development Environment
3D	Three Dimensional
2D	Two Dimensional
OpenCV	Open-Source Computer Vision Library
Dlib	C++ toolkit for ML and data analysis
MindAR.js	Lightweight Web AR framework
RetinaFace	Face Detector with Landmark Detection

CHAPTER ONE

INTRODUCTION

1.1 Background of Study

Face shape classification is focusing on recognizing the structure and blueprint of a person's face from an image which is an important task in computer vision (Hossam et al., 2021). It is widely used in fields like beauty and fashion for recommendation (Salim et al., 2023). A notable application is in selecting suitable eyewear, as the choice of glasses is highly related to a person's face shape (De Luna et al., 2024). Experts suggest tailoring frame styles to specific face shapes will help optimal balance (Suriyalakshmi et al., 2023) and automated systems that have been developed can provide personalized eyewear recommendations based on the classification (Grd et al., 2024). Furthermore, face shape classification can be integrated with Augmented Reality (AR) (Sagarika & Mokashi, 2024). This is because AR can help to serve as a filter, allowing users to virtually try on items such spectacles and visualize how the item will appear on the specific face shape (Sagarika & Mokashi, 2024), this is improving applications such as virtual eyewear try-on.

The current trends and technologies in face shape classification strongly rely on machine learning and deep learning approaches, particularly Convolutional Neural Networks (CNN) (Sagarika & Mokashi, 2024). The advanced CNN architectures like GoogLeNet, MobileNetV3, ResNet50, Sequential CNN, InceptionV4, InceptionV3, Vit Small, DenseNet121, VGG16, EfficientNetV2S and Swin Transformer are being explored for their performance (Grd et al., 2024) as some achieving high accuracy like GoogleNet at 91.03%, VGG-16 at 96.83% in some studies (Sagarika & Mokashi, 2024). Next, transfer learning, utilizing pre-trained CNN models is a common strategy to improve classification accuracy and model performance (Grd et al., 2024). The detection of facial landmarks, predefined points on the face are often important for feature extraction and classification (Hossam et al., 2021) for example it uses models like RetinaFace and ARCFace (Loukil et al., 2023). These technologies support a lot of real-world applications including the personalized recommendation for hairstyles, makeup, and eyewear designs (Salim et al., 2023). Face shape and landmark detection are also relevant in dentistry and cosmetic surgery, forensic science for identification, gaming and AR industries (Sagarika & Mokashi, 2024).

Other than that, even with these advancements there are still several problems and gaps persist in face shape classification finding and its application. In order to handle various face orientations, different poses and obstructions such as masks, glasses, or hats remain a significant challenge (De Luna et al., 2024), as there is also performing classification under poor lighting conditions (De Luna et al., 2024). Accurately detecting facial feature points is challenging under a wide range of face deformation including rigid changes like scaling and rotation and non-rigid changes like expressions and aging (Grd et al., 2024). Gathering sufficient information and representative training datasets to cover all possible face shapes across different demographics is difficult (Loukil et al., 2023). Many existing datasets may lack diversity where some focus on specific groups or be limited in size (Hossam et al., 2021) and the process of labeling face shapes is time consuming and prone to human error (De Luna et al., 2024). This may lead to ethnic or racial biases in some classification methods (Loukil et al., 2023). There is also no consistent number of face shapes universally used across studies (Grd et al., 2024). The technical challenges like accurately segmenting the hairline are complicated by barriers from hair, accessories of facial hair (Grd et al., 2024) and some classification methods may require significant computing resources that limit their use on devices with limited power. To be truly effective for recommendation systems, the precision of classification needs to be advanced (De Luna et al., 2024) and future research may need to incorporate more features beyond just shape such as lineaments in order to enhance the accuracy of recommendations (Loukil et al., 2023).

The possibility to work out a framework, such as the one that is potentially operating on that is it will target classification of face shapes to be applied in the area of eyewear suggestions that would be developed on the basis of the technologies and studies is outlined (Suriyalakshmi et al., 2023). The project will deal with the implementation of deep learning models, probably CNN or any architecture similar to them which will estimate faces in accordance with the categories of face shapes that are relevant (Sagarika & Mokashi, 2024). It demands searching the identified challenges like insisting on resilient performance regardless of changes in the quality of the image, facial pose or partial occlusion (De Luna et al., 2024). Collecting or creating an appropriate and broad enough labelled dataset is also an important factor, which deals with the restrictions on data in the existing studies (Loukil et al., 2023). Thus, the purpose of this project is to create face shape detection system that has a spectacle recommending engine and an interface that displays a virtual fitting. The system utilizes deep learning to classify and detect facial landmarks so as to provide personalized and interactive user experience.

1.2 Problem Statement

Face shape classification is a base task with widespread application in the fields such as computer vision, fashion, beauty, eyewear selection and even security (Sagarika & Mokashi, 2024). It provides an important beginning step for advising personalized recommendations for items like hairstyles, makeup or eyeglasses (Rifat et al., 2023). On the other hand, achieving accurate and reliable facial shape classification in real-world scenarios present significant challenges (Loukil et al., 2023). These challenges are from the inherent variability and complexity of the human face, the conditions under which images are captured and limitations in existing methodologies (Loukil et al., 2023). By addressing these challenges is essential for developing robust automated facial classification systems that can effectively leverage face shape information for practical application (Rifat et al., 2023).

Incorrectness and inconsistency in face shape classification techniques are one of the challenges (Hossam et al., 2021). The absence of large, diverse and correctly labelled data sets is one of the most important reasons (Grd et al., 2024). Most of the studies require smaller and some even utilize personal datasets or capture them online where it can cause underrepresentation of a demographic and bias (Grd et al., 2024). Besides that, face shape is by far a branding venture and a time consuming and at times subjective process that might involve the intervention of various professionals. It also tools in inconsistency and human errors in datasets (Sagarika & Mokashi, 2024). Moreover, it has also been proved that the traditional systems of face categorization such as the one proposed by Poch are quite redundant in terms of morphology and has low values of consistency in both intra and inter observer, and results in a hard time of classifying faces, especially those that have average arrangement. This variance in classifying faces affects the security of the systems that are made to rely on proper face shape recognition (Salim et al., 2023).

The effectiveness of systems of face shape identification especially for personalized recommendation like selecting spectacles is further limited by the lack of integrated and real-time platforms that can handle the complexities of real-world use (Rifat et al., 2023). As system targeting to recommend eyeglasses based on face shape, they are facing challenges such as achieving sufficient accuracy for real-time identification (Rifat et al., 2023) and handling variations like different face orientations or any other obstacles like mask and glasses (De Luna et al., 2024). By implementing virtual try-on capabilities requires systems that can adapt to random user postures and handle varying image conditions (Ishikawa & Ikenaga, 2022). Moreover, simply classifying faces shape is often insufficient for providing highly accurate personalized recommendations. The effective recommendations also depend on other factors like facial characteristics (Suriyalakshmi et al., 2023). Incorporating these diverse elements into an integrated, real-time system needs to add significant complexity that will pose a challenge for the development of truly comprehensive and user-friendly recommendation platforms (Suriyalakshmi et al., 2023).

1.3 Research Questions

I. How can a machine learning model be developed to accurately classify face shapes based on facial images?

II. How can a system recommend suitable spectacle frame styles according to classified face shapes?

III. How can an augmented reality (AR) module be implemented to overlay recommended spectacles onto a user's face in real time?

1.4 Research Objectives

I. To develop a machine learning model that can classify face shapes based on facial image data using VGG16.
II. To create a recommendation system that maps face shapes to suitable spectacle frame styles.
III. To implement an augmented reality (AR) module that overlays the recommended spectacles onto the user's face in real-time using webcam input.

1.5 Research Scope

This project focuses on the development of an integrated system that will classify a person's face shape and recommends the suitable spectacles for them. It is also targeted to enhance with the real-time virtual try-on feature. The scope of this project is as follows:

- Face Shape Classification

The system will classify the face shapes into predefined categories such as oval, round, heart, square and oblong that is based on facial landmarks extracted from front-facing images using computer vision and deep learning techniques. The classification will be limited to adult human faces and accuracy may differently depending on head pose constraints and lighting.

- Spectacles Shape Recommendation

Based on the detected face shape, the system will give list of suggestions for suitable eyewear frame styles. The recommendations will follow general design guidelines from the optical fashion principles and will not include user preference learning or optical prescription matching. The spectacle shapes used are round, rectangular, cat-eye, wayfarer, square, aviator, geometric, browline, and oval.

- Virtual Try-On Implementation

A real-time virtual try-on module will be developed using a live camera feed. It will overlay spectacles on the user's face using basic 3D AR techniques that allow the virtual eyewear to stay aligned with facial landmarks even if the user slightly moves their head. The face tracking is a limited range of head motion but excludes complex 3D.

- Dataset Acquisition

The development will utilize 5000 rows of face shape datasets from Hugging Face https://huggingface.co/datasets/bkprocovid19/face_shape. Every face has a thousand of rows. The dataset was selected due to its relevance to real-world facial diversity, balanced class distribution and good image quality.

1.6 Research Significance

This research can contribute to the academic and technological development in the area of computer vision, artificial intelligence and augmented reality. This is because when exploring the integration of face shape classification with real-time on virtual reality systems. It also adds value to future studies in facial analysis especially in improving the accuracy under real-world conditions such as different head poses and facial obstacles. The system may demonstrate how machine learning models and augmented reality can be combined to solve practical problems in personalized product recommendation. This aligns with the current trends in AI research that focus on human-centred design and interaction.

From the point of view of an industry and user perspective, this project can offer a solution that can enhance online shopping experience particularly in the eyewear market. By providing real-time and personalized recommendations users can have better choices of spectacles that are suitable for their face shape that can increase their satisfaction and reduce return rates. The system also supports the goal such as Sustainable Development Goal (SDG 9) which is industry, innovation and infrastructure by promoting smart digital tools. Other than that, it supports the SDG12 which is responsible for consumption and production by helping the users to make informed choices that reduce the unnecessary product waste.

1.7 Summary

This chapter introduces the research project that focuses on developing a system that can identify a person's face shapes, recommend suitable spectacles and offer a real-time try-on experience using augmented reality (AR). It begins with the background of the study which explains the significance of face shape classification in fashion and eyewear applications. Then, the problem statement highlighted the current challenges in accuracy, integration and real-time performance of existing systems. The research question and objectives are outlined to guide the development of a good and user-friendly solution. While the scope defines the boundaries of the study such as using front-facing images and limiting the platform to web-based only. Lastly, the significance of the research is discussed in terms of this project contributions to the academic, industry and end-users.

CHAPTER TWO

LITERATURE REVIEW

This chapter provides an overview of the problems that this project is based on. It starts by looking at the human face and the major structures of the face, the definitions of the face shape, and classification methods that are a part of the literature that is already available. It further examines face detection and facial landmark extraction as fundamental processes of studying facial structure. The chapter moves on to a summary of the types of eyewear and the complementary effect of various frames to particular facial features, and the methods of rules-based recommendation. Virtual try-on technologies based on Augmented Reality (AR) are also discussed. Lastly, there are related works and comparative studies that bring out the present capabilities, limitations, and gaps in research.

2.1 The Human Face and Face Shape Classification

The face is partitioned into clear parts such as the eyes, the bridge of the nose, the jaw, the cheeks and the forehead which all have particular morphological features that are highly different among the genders and ethnicities. These areas collaborate to create a system of aesthetics which identifies the final look and appeal of the face.

2.1.1 Face Shape Definition and Categories

Face form is defined by an object geometric structure and appearance that is one of the crucial components of a person physical appearance (Salim et al., 2023). Due to internal and external factors then it comes in a variety of shapes and size with differentiation calls for a wide range of characteristics (Hossam et al., 2021).

Classifying facial structures into distinct categories is known as face shape classification (Salim et al., 2023).

It also importance in eyewear recommendation because classifying face shapes is a common fashion challenge for appropriate eyeglasses (Hossam et al., 2021). The choice of eyewear greatly affects facial symmetry because it uses for best balance and the frame types are determined by each face shape (Sagarika & Mokashi, 2024). Uncomfortable, dizziness or blurred eyesight might result from poorly fitted eyewear (Tian & Ball, 2023a). Comprehending facial shapes facilitates customized eyewear recommendations and enhancing customer satisfaction and self-assurance (Rifat et al., 2023). Here are descriptions of the face types and the specific recommendations include:

a) Oval

   It is distinguished by its narrow width and a bit long width (Hossam et al., 2021). The sides of the face are curved and the chin is narrower and rounder than the forehead (Hossam et al., 2021). Oval faces have a longer vertical axis and balanced proportions (Sagarika & Mokashi, 2024). They are thought to have natural characteristics of flexibility and balance (Rifat et al., 2023). Round, cat-eye, wayfarer, square, aviator, geometric, and browline frames all work well with oval faces (Suriyalakshmi et al., 2023).

b) Oblong

   It is distinguished by a height that is significantly longer than its breadth (Hossam et al., 2021). Oblong faces usually have straight sides and are longer than wide (Grd et al., 2024). To provide depth and contrast, oblong faces need width-enhancing designs like wayfarer, browline, oval, or geometric forms (Sagarika & Mokashi, 2024).

c) Round

   The forehead, cheeks, jaws, and height are all roughly comparable in size (Hossam et al., 2021). They are softly rounded and roughly equal in length and width (Sagarika & Mokashi, 2024). So round face is complemented by sharp-angled frames like rectangular, square, aviator and geometric styles for prolonging (Suriyalakshmi et al., 2023).

d) Square

It distinguished by broad foreheads that are as wide as the jawline and cheekbones, as well as strong, angular jaws (Hossam et al., 2021). This type of face shape is regarded as one of the rarest(Hossam et al., 2021). A firm jawline and equal width across the forehead, cheeks, and jaw are characteristics of square faces (Sagarika & Mokashi, 2024). As square faces are softened by round or oval frames, while cat eye, wayfarer, and browline styles add elegance (Suriyalakshmi et al., 2023).

e) Heart

It distinguished by a narrower chin and a larger forehead (Hossam et al., 2021). It has the characteristic heart-like form of the forehead, which tapers down to the thin chin (Hossam et al., 2021). The heart-shaped faces are balanced with cat eye, rectangular, and oval frames, with wayfarer and browline adding boldness. Bottom-heavy or rimless frames can also be suitable (Suriyalakshmi et al., 2023).

2.1.2 Feature Extraction Techniques

Face shape categorization that relies on extracting different facial features is a popular task in beauty and fashion (Hossam et al., 2021). The target of this system is to offer tailored suggestions for eyewear that applied a person unique face characteristic (Hossam et al., 2021).

a) Feature Measured

Faces are categorized using the computer vision techniques that find the face oval, round, square, heart and oblong shapes (Sagarika & Mokashi, 2024). Due to the differences in size and shape brought on by internal and external factors has made human face is complicated and requires a wide range of characteristic for identification (Hossam et al., 2021). The commonly measure are face height and width, forehead width, jawline width and min-jaw width, cheekbone width and the angles of different facial landmarks with respect to the chin landmark (Grd et al., 2024).For common use, 68 facial landmarks found on a frontal view image are frequently used to calculate these measurements (Sagarika & Mokashi, 2024). Even most landmarks may be found with a typical library, it still can be hard to determine the important spots like the forehead and may require further computations like estimating the entire face height by measuring the distance from chin and adding half of that distance to a central point (Hossam et al., 2021). Facial features are often expressed as ratios and angles derived from distances between these landmarks. Examples of such ratios involve ratio between face height and width (Grd et al., 2024), ratio of jawline width to face width (Hossam et al., 2021) and ratio of mid-jaw to jawline width (Hossam et al., 2021).

b) Challenges in Face Shape

Despite the advancements, there several challenges like inconsistency in poses, lighting and expressions. This happen because of scaling, rotation, translation, and other hard face deformations, as well as emotion and age, can all have significant effects on how well facial feature points are detected (Algaraawi et al., 2024). Next, the data quality and quantity that need for more diverse, larger and properly labelled face shape datasets (Grd et al., 2024). While small datasets are common and labelling face shapes is consuming time and subjective process that may lead to human error (Grd et al., 2024). Other than that, the computational resources as the complex algorithm may need significant computing resources and limit the applicability on devices with limited capabilities (Loukil et al., 2023). Last but not least, the hairline segmentation where to determine the boundary between skin and hairline is a persistent problem as this region is frequently hidden by accessories or hair (Grd et al., 2024).

2.1.3 Classification Approaches

There are rule-based and machine learning techniques know as classification methods that are used to group data according to input features into predetermined classes or labels. Decision trees, K-nearest neighbours (KNN), support vector machine (SVM) and deep learning models such as convolutional neural network (CNN) are examples of popular methods. Table 2.1 shown the easy understanding comparison among the classification methods.

a) Rule-Based Classification

In ruled-based categorization the face shapes are identified by applying predetermined geometric criteria. For example, if the width is equal to height and jaw is round then it is a round face could be a rule. Because the rules are clear, this approach is simple to use and understand. It is suitable for this recommendation system as the face is already been detected in the early step. By apply the rule based on the face identification will make the program faster. However, when given the complexity and diversity of human faces it might not be very generalizable. In the past, these methods might have been in line with the manual measurements and subjective evaluations that professionals like to utilized (Sagarika & Mokashi, 2024).

b) Machine Learning Models

i. Support Vector Machine (SVM)

SVM is a supervised machine learning technique that is applied to regression and classification. Each data item is plotted as a point in an n-dimensional space where n is the number of characteristics in classification and a hyperplane that best distinguishes between two or more classes is found (Hossam et al., 2021). New samples are more accurately classified as the farther the hyperplane is from the support vectors which is nearest points. SVM is thought to perform well with high-dimensional data (Hossam et al., 2021).

Based on the kernel and particular implementation, SVM classifiers have demonstrated a range of accuracies. In one comparison investigation, SVM with a radial basis function (RBF) kernel had the highest overall accuracy of 82% (Hossam et al., 2021). Additional findings include SVM-Linear (SVM-LIN) obtaining 70% test accuracy and SVM with RBF kernel reaching 72% accuracy for hairstyle recommendations (Hossam et al., 2021). SVM classified faces into four forms with an accuracy of 73.68% when applied to 3D human data. When compared to other methods, SVM-LIN achieved 55.2% accuracy and SVM with RBF kernel achieved 50.6% accuracy in a different study (Hossam et al., 2021).

ii. Decision Tree/ Random Forest (RF)

The machine learning algorithm Random Forest uses a group of decision trees. The classifier is produced by voting for the most popular class after each tree is created in accordance with random vector restrictions (Hossam et al., 2021). In general, these models are simple to understand. At 70% accuracy, the Random Forest algorithm demonstrated its peak test performance (Hossam et al., 2021). In another work, a similar ensemble approach called a Gradient Boosting Tree Classifier produced an accuracy of 70% (Salim et al., 2023).

iii. K-Nearest Neighbours (KNN)

KNN is a widely used and simple-to-implement to non-linear classification technique. It learns the training dataset by heart rather than inferring a function from it. It uses the 'k' closest points to classify a new input, and the categorization output is decided by a majority voting system (Hossam et al., 2021). When similar faces organically cluster, KNN performs well. In one example comparison, KNN's accuracy was 64.6% (Hossam et al., 2021).

c) Convolutional Neural Networks as Deep Learning in Classification

A subclass of deep learning models called CNNs is employed for tasks involving picture recognition and classification. CNNs can learn from raw data and automatically extract complicated features from unprocessed images, unlike older approaches (Loukil et al., 2023). They are made up of layers that gradually pick up complex patterns, including convolutional, pooling, and fully linked layers (Sagarika & Mokashi, 2024). Although CNNs can perform better than conventional machine learning models, they usually need more data and processing power to train efficiently. Advanced CNN architectures have achieved high accuracies in face shape classification. For example, by using Inception modules to obtain multi-scale features, GoogLeNet was able to attain the best accuracy of 91.03% (Sagarika & Mokashi, 2024). Next, MobileNetV3 that is designed for lightweight, real-time mobile applications has come in second at 89.74% (Sagarika & Mokashi, 2024). ResNet50, which uses residual networks with skip connections to capture complicated features, achieved 88.54% (Sagarika & Mokashi, 2024). The accuracy of a sequential CNN was 80.02% (De Luna et al., 2024). With data augmentation, the Swin Transformer model's accuracy was 86.34% (Salim et al., 2023). When used on the entire dataset in one investigation, Inception v3 showed excellent performance, with an overall accuracy of 97.8% (Rifat et al., 2023). EfficientNetV2S, a transfer learning technique, produced an exceptional 96.32% total accuracy (Grd et al., 2024).

Table 2.1 Comparison of Classification Methods

Feature	Rule-Based Classification	Machine Learning Models (SVM, Random Forest, KNN)	CNN (Convolutional Neural Network)
Need for Feature Engineering	High (based on predefined geometric rules)	High (rely on hand-crafted features like landmark distances and angles)	Low/Automatic (learns features directly from raw images)
Accuracy	Limited generalization (implied by design)	Varied, generally lower than deep learning (e.g., SVM up to 82%1, RF 70%25, KNN 64.6%17)	Generally highest (e.g., EfficientNetV2S 96.32%36, Inception v3 97.8%17)
Interpretability	High (rules are explicit and understandable)	Moderate to High (Decision Trees and KNN are interpretable; SVM's hyperplane	Low (often considered a "black box" due to complex internal representations)

		is conceptually clear)	
Best Use Case	Simple, well-defined scenarios; foundational understanding	High-dimensional data (SVM), effective with moderate datasets and good for problems where features can be engineered	Large, diverse datasets, complex image patterns, real-time applications; superior generalization for varied face appearances

2.2 Face Detection and Landmark Extraction

In face study applications, face detection and face landmark extraction are important preprocessing steps. These processes are complex as to categorize the face shape, emotion identification and virtual object placement that can enable the systems to recognize and interpret human facial features. For ensure proper positioning of virtual glasses on the user's face during the virtual eyewear try-on process, it is necessary to precisely locate the face and its important landmarks. This section is covered for face detection with the followed of approaches for facial landmark extraction

2.2.1 Overview of Face Detection Techniques

One of key task in computer vision is face detection where it recognises and proper locate of human faces in digital images or video streams. It is an essential initial step for many applications of face analysis like facial landmark detection, face reconstruction, express recognition and face shape classification (X. Li et al., 2024). This face detection can be divided into two section such as traditional face detection techniques and modern face detection techniques.

a) Traditional Face Detection

i. Haar Cascade Classifier (Viola-Jones Algorithm)

The Haar Cascade classifier is a machine learning based method for object detection that including faces and it is frequently linked to the Viola-Jones algorithm (Suriyalakshmi et al., 2023). The Haar-like features is a straightforward rectangular feature that capture pattern like edges, lines and four-rectangle features. The features are found by examining the pixel structure of grayscale images (Sagarika & Mokashi, 2024). After that, a cascade of classifier is basically a sequence of increasingly more complicated decision stages that evaluate these features (Suriyalakshmi et al., 2023). A region is categorized as a face and a circle of boundaries is made around it if it passes all phases. Usually, libraries like OpenCV are used to implement this approach (Salim et al., 2023) Haar Cascade classifier area often quick and easy to use but they have struggle in handling the non-frontal facial positions, low lighting and changes in head angles (Rifat et al., 2023). Additionally, they are not as resistant to obstacles (Rifat et al., 2023).

ii. Histogram of Oriented Gradients with Support Vector Machine (HOG + SVM)

This technique involves of extracting detailed edge and gradient structures from facial regions using the histogram of oriented gradient (HOG). After that, a support vector machine (SVM) classifier to distinguish between facial and non-facial regions is fed the HOG features (Sanjar et al., 2024). This technique is frequently applied to face detection in preprocessing stages. Even though HOG and SVM techniques are often more reliable than Haar Cascade Classifier it still may have limitations when it comes to identifying faces in diverse positions or obstacles (Rifat et al., 2023).

b) Modern Face Detection

i. Convolutional Neural Networks (CNN) - based Detectors

A particular kind of deep learning model that been made for image processing application is called as convolutional neural network (CNN) (Suriyalakshmi et al., 2023). This model does not require manually engineered features because they can automatically learn hierarchical features straight from raw image data (Huang et al., 2024). CNN can now handle difficult situations including different stances, different light and partial obstacles with far more accuracy and robustness thanks to these capabilities (Zou et al., 2025). The CNN architectures ResNet50, MobileNetV3, GoogLeNet and VGG16 are widely used for face detection and classification (Sagarika & Mokashi, 2024). CNN are frequently made for real-time applications and are capable of achieving cutting-edge accuracy performance (Sagarika & Mokashi, 2024). Table 2.2 shows that VGG16 is the best to be use.

Table 2.2 Comparison CNN Architecture Accuracy

Architecture	Accuracy
RestNet50	88.54 %
MobileNetV3	89.74 %
GoogLeNet	91.03 %
VGG16	96.83 %

ii. RetinaFace

RetinaFace is an effective tool for a variety of computer vision applications like augmented reality (AR), facial identification and emotion analysis. This architecture is a cutting-edge single-shot face detector that uses deep learning (Loukil et al., 2023). It is quite effective and can do many things at once like identifying faces and estimating the five main landmarks which are mouth, corner, nose and eyes in a single pass (Loukil et al., 2023). It is also capable in identifying the person's gender (Loukil et al., 2023). Other than that, with the help of anchor boxes and multi-scale feature pyramid this architecture can reliably identifying faces with varying scales and orientations (Loukil et al., 2023). Even under difficult situations it has known for the excellent precision and efficiency (Loukil et al., 2023). This method is a complete head detection method since it targets the entire head region including the ears, hair, forehead and neck (Loukil et al., 2023). This is because its simultaneous ability to deliver precise facial detection and landmark localization (Loukil et al., 2023).

Table 2.3 Comparison of Face Detection Techniques

Method	Accuracy	Speed	Pose Handling	Suitable for AR
Haar Cascade	Lower particularly in varied conditions	Fast and simple	Struggles with non-frontal angles	Less suitable due to limitations
HOG + SVM	Moderate and better than Haar Cascade	Computationally efficient	Limited with different poses	Less suitable due to limitations
CNN-based and RetinaFace	High	Real-time capable and highly efficient	Robust can handles large pose variations well	Highly suitable for precise virtual item placement

The relevance methods to the project that focused on face shape classification, the modern deep learning techniques as shown in table 2.3 like CNN models and specifically RetinaFace is important over traditional methods.

a) Better Landmark Alignment and feature extraction

Accurately detecting faces and important facial landmarks in a variety of situations like various orientations, poor lighting or obstacles that make traditional techniques such as Haar Cascade and HOG + SVM has limitation (Rifat et al., 2023). This is a major disadvantage because reliable facial features such as angles and distance ratios that determine face shape can only be extracted with precise landmark detection (Hossam et al., 2021). On the other hand, CNNs are able to automatically extract complex, hierarchical features from unprocessed images and produce feature that more accurate and robust (Sagarika & Mokashi, 2024).

b) Real-time performance

Real-time processing is needed for many face shape classification applications like smartphone apps for fashion and beauty advice or internet guides (Sagarika & Mokashi, 2024). Despite its speed, Haar Cascade is less appropriate in "in-the-wild" situations due its accuracy limits (Rifat et al., 2023). High accuracy and real-time capability are features of modern CNNs especially optimized architectures like RetinaFace that allow for smooth user experiences(Sagarika & Mokashi, 2024).

c) Robustness to occlusion and pose variation

In the real-world situations, faces are frequently hidden by various head positions and obstacles like mask, glasses and hats (De Luna et al., 2024). In these situations, traditional approaches frequently fall short or perform poorly (De Luna et al., 2024). RetinaFace is quiete successful in these difficult situations because of its capacity to simultaneously recognize faces and landmarks as well as it robust to different scales and orientations (Loukil et al., 2023). For face shape categorization to be very accurate even when facial features are partially hidden this robustness is crucial (De Luna et al., 2024).

d) Enhanced face shape robustness

The main objective is to accurately characterize face shapes. Lower accuracy and dependability in future face shape categorization are directly related to the limits of conventional face detection techniques (Grd et al., 2024). The project can gain from using the advanced CNN-based model like RetinaFace because of its high accuracy with strong feature learning and adaptability to a variety of real-world scenarios. This leads in more accurate and dependable face shape classification and better personalized recommendations (Sagarika & Mokashi, 2024).

2.2.2 Facial Landmark Detection Approaches

Finding specific crucial places on the face is known as facial landmark detection. The eye corners, nose tip, mouth corners, chin, jawline and eyebrows are some of face important features. Some research also relies on 81 landmarks or sometimes it up to 468 landmarks for detailed face grids (Grd et al., 2024) and there are 68-point facial landmarks among the typical standards (Sanjar et al., 2024). These positions can be estimated with mm-level accuracy using recent methods like those that use milimeter-wave (mmWave) radar (Y. Li et al., 2024).Facial landmark detection is used to derive face geometry and align overlays like virtual glasses:

i. Face geometry

It makes the face geometric structure visible and help in comprehending the shape of the object which is difficult for human faces to distinguish and needs a variety of characteristics (Huang et al., 2024). This is essential for classifying faces shapes because a task that is widely utilized for many applications (Hossam et al., 2021). It provides the foundation for research on facial technology by enabling precise localization of every facial feature (Huang et al., 2024).

ii. Align overlays

Virtual objects can be precisely positioned during the Augmented Reality (AR) interactions because of landmarks. They are crucial for virtual try-on systems, especially when it comes to choosing eyeglasses (Luidolt & Zhou, 2023). So, the individual facial landmarks and 3D scanned data can be used to create custom-fit eyeglass frames for the applications (Tian & Ball, 2023a).

The importance of key points is it essentially for aligning virtual objects and defining facial characteristics. For instance, eye position helps align glasses horizontally, the nose bridge helps with vertical placement and the jawline help define face shape. Classifying face makes it easier to choose appropriate eyewear (Hossam et al., 2021). Certain frame styles that complement various facial form are recommended by expert (Sagarika & Mokashi, 2024). Besides that, in custom eyewear design the association between face shape and eyewear component like arm size, nose bridge size and frame size are established using ten secondary measurements and a set of 32 basic facial landmarks (Tian & Ball, 2023a). A crucial component of contact fit is the nose bridge where the curves should match the user nose bridge shape as determined by particular landmarks (Tian & Ball, 2023a).

Other than that, a bad augmented reality experience or incorrect face classification can result from misplaced landmarks or incorrect face shape classification. Face shape determination using AI may result in misclassification (Hossam et al., 2021). Next, Eyewear that fits poorly could result in pain, dizziness, or obscured vision (Tian & Ball, 2023a). The low quality of landmark detection directly downgrades the final system performance. Temporal aspects may be affected by internal bouncing sounds in face landmarks (R. Wang et al., 2024). Errors in virtual try-on systems might result in faulty outfit transfers due to incorrect data capture or obstacles like hair or beards (Luidolt & Zhou, 2023).

a) Tool and eLibraries

i. Dlib

This pretrained model are accessible and it is widely utilized, quick and accurate. It is a modern toolset with tools and methods for machine learning (Sanjar et al., 2024). There are 68 facial landmarks, face identifiers and a frontal face detector are specifically utilized to acquire face shape predictors (Sanjar et al., 2024). The important landmarks such as the eyes, eyebrows, nose, mouth, chin, lips, forehead and general form of the face can be detected by this library (Sanjar et al., 2024). It is used to recognize faces in applications for the blind and visually barrier with and impressive accuracy of 95.64% (Sanjar et al., 2024). Dlib also uses Euclidean distance to identify faces while providing a computationally effective method that enables quick and accurate recognition even with a small number of training photos(Sanjar et al., 2024). It also considered to have user-friendly interfaces (R. Wang et al., 2024).

ii. MediaPipe

It is a well-known library for being lightweight, real-time and appropriate for both web and mobile apps. MediaPipe provides a very detailed mesh of the face because of 468 landmarks, making it good for AR alignment. With a Normalized%-MSE of 5.01 on the MAFL dataset and 6.80 on the AFLW dataset in comparative tests, it demonstrates good accuracy (Zou et al., 2025).

iii. OpenFace

This library is an open-source library for analysing facial expressions. It is more frequently utilized in research and typically less focused on real-time detection than direct landmarks detectors. The ability of deep learning-based facial landmark extraction techniques to greatly increase recognition speed has led to their widespread integration into toolkits such as this library (R. Wang et al., 2024).

Table 2.4 Comparison Tools and Libraries

Feature/Metric	Dlib	MediaPipe	OpenFace
Number of landmarks	68	468	Although OpenFace does not have clear specifications, it combines various techniques that identify different numbers of face landmarks.
Accuracy	Accurate: 95.64% accuracy in VIP facial recognition was attained. Research on stylized faces revealed an NME of 16.32%.	Shown 6.80% on AFLW and 5.01% Normalized%-MSE on MAFL.	Accuracy depends on the specific facial landmark detection methods integrated within the OpenFace framework.
Speed / Real-Time	Fast and computationally efficient. Allows for swift and precise recognition.	Real-time and lightweight, it is appropriate for both web and mobile platforms.	Can significantly boost recognition speed when deep learning-based facial landmark extraction methods are integrated. Generally, less real-time oriented for broader facial behaviour analysis.

Ease of use	Pretrained models are accessible and widely utilized. features interfaces that are easy to use.	Described as appropriate for both mobile and web.	Open-source library with interfaces that are easy to use.
Suitability for the project	Excellent for facial landmarking in general, particularly when 68 key points are enough. It offers high identity recognition accuracy and is effective for embedded devices with limited resources.	Great for tasks that need extremely precise facial geometry, such 3D face mesh reconstruction or sophisticated AR alignment.	More appropriate for research or projects involving face behaviour analysis where a thorough examination of facial expressions and action units is crucial.
Suitable for AR	Indeed, it may be applied to simple AR overlays and generic landmark detection.	Yes, its detailed geometry capabilities make it excellent for AR alignment. It allows virtual objects to be placed precisely.	Though it could need more real-time optimization for direct AR overlay, it could be used as a backend for facial feature extraction for AR apps that incorporate behaviour analysis.

2.3 Eyewear and Facial Fitting

Glasses nowadays are not only vision aids, but have to be comfortable, more appealing, and efficient. A good choice of frames is one that takes into account long term wear comfort, how much the design would work well with the features of the face and how much the frame would match the visual requirements of the user. The wrongly selected frames may bring pain to pressure, slipping, an unequal distribution of weight, and a poor image. Frames should also be able to fit various types of lenses, including reading glasses and progressives as well as safety eyewear. To accomplish optimal balance between comfort, looks and functionality, a methodological procedure, aligning the frame features with the facial geometry of the wearer and the demands of the usage, is required.

2.3.1 Significance of Proper Frame Selection

a. Comfort Considerations

Physical comfort represents a fundamental requirement in frame selection that directly impacts daily wearability and user satisfaction. Frame width plays a critical role, as frames that are too narrow compress lateral orbital tissues and create excessive pressure between the temple arms and head, leading to symptoms such as head distention and headaches (Bai et al., 2021). The temple arm design must also match individual ear anatomy, since improper length from the frame hinge to the ear contact point, combined with bending contours that don't align with the ear base outline, results in frames that cannot maintain stable positioning (Bai et al., 2021).

Vertex distance is distance between frame front and face that must be carefully optimized to get a compromise between comfort and functionality. Frames that are too near the face may cause irritation to eyelashes and face, whereas a distant vision may be worsened by too big a distance between the frame and face(Tian & Ball, 2023b). Another essential fit aspect, which should ensure secure contact with the head and should not cause uncomfortable points, is the arm width of the temple, and it should be adjusted depending on the preferences of the person and the frame should not lose the ability to hold the head in place (Tian & Ball, 2023b). The problem is that it is hard to fit the great diversity of the position and measurements of different landmarks on the face between people as these changes can produce large changes in the frame size and proportions to provide the maximum comfort (Tian & Ball, 2023b).

b. Functional Considerations

Functional considerations in frame selection centre on ensuring frames can reliably perform their intended purpose across different usage scenarios and prescription requirements. The relationship between frame dimensions and facial anatomy must be precisely calibrated to maintain both structural stability and optical effectiveness (Bai et al., 2021). Frame stability represents a critical functional requirement, as improper temple arm length or mismatched bending contours that don't align with ear anatomy can prevent frames from maintaining their intended position during use (Bai et al., 2021).

The challenge in functional frame design lies in accommodating the dramatic variations in facial landmark positions and measurements between individuals, as these differences require corresponding adjustments to frame structure and proportions to maintain consistent functional performance (Tian & Ball, 2023b). Modern approaches address this by establishing systematic relationships between body parameters extracted from facial measurements and adjustable frame parameters that can be modified to maintain optimal functionality across diverse facial geometries (Tian & Ball, 2023b).

c. Aesthetic Considerations

The challenge in aesthetic frame selection lies in accommodating the dramatic variations in facial structure between individuals, as different landmark positions and measurements can completely alter which frame proportions and styles will be most flattering (Tian & Ball, 2023b). Professional approaches now involve specialized evaluation by visualists and aestheticians who can systematically assess how different frame characteristics interact with specific facial geometries. This systematic approach addresses the growing demand for customization in eyewear design, where frames must be adapted not just for fit and function, but also to meet diverse aesthetic preferences and style requirements (Tian & Ball, 2023b).

2.4 Eyewear Recommendation System

The purpose of eyewear recommendation systems is to make it easier to find glasses that are both visually pleasing and fit properly. These systems typically assess human facial features using the computer vision techniques such as relying on Convolutional Neural Network (CNN). A key step includes face shape classification to identify the user's face type like oval, round, square, heart and oblong from an image. The algorithm suggests the appropriate eyeglass frames from a database bases on it research. So, the goal is to help provide personalized suggestions that could improve the user experience and helping them choose the frames that are visually compatible with their face shape.

2.4.1 Rule-Based Recommendation Approaches

Based on the research, rule-based methods for recommending eyeglasses generally depend on understanding out the form of user face first. After that, applying the predetermined style guidelines or standard to offer appropriate eyewear frames (Sagarika & Mokashi, 2024). This approach is based on common wisdom and recommendations from specialist in the field of beauty, fashion and optics that suggest the specific frame designs match specific facial shapes more effectively (Sagarika & Mokashi, 2024). The basic idea is to suggest frame that balance face features, highlight curves or improve the wearer's appearance (Sagarika & Mokashi, 2024). For example, square faces should have round or oval frames to soften their sharp features. Meanwhile for round faces should have rectangular or sharply angled frames and etc (Hossam et al., 2021). These well-established principles are based automated systems that can categorize a user face shape and use the associated style rules to filter or evaluate possible eyewear selections from a database (Sagarika & Mokashi, 2024).

However, there also point out that the flaws and difficulties of strictly rule-based face shape classification algorithms. One of the problems is that facial shape identification can be subjective, time consuming and prone to mistakes as inconsistent among specialist especially when done manually or with older methods (Hossam et al., 2021). Additionally, it can be hard to only rely on predetermined face shape categories because many faces may not cleanly fit into one group and certain traditional classification show structural redundancy (Hossam et al., 2021). Furthermore, more than simply face shape is involved in making correct glasses recommendation like skin tone, cultural norm, personal preferences, current fashion trends and other special facial features that have a big impact on interaction and user satisfaction (Loukil et al., 2023). There is also indicates that to overcome the limitations of basic rules-based mappings is by more objective techniques must be formed like expert feature extraction and possibly considering other personal attributes. Other than that, using deep learning to learn complex relationships even though face and characteristic features stay basic concept (Sagarika & Mokashi, 2024).

2.5 Augmented Reality (AR) in Virtual Eyewear Try-On

The capacity of augmented reality (AR) to enhance user interaction through digital overlays has made it a trend technology in the fashion and retail industries. This is because user can see many glasses could appear on their faces in real-time with AR-enabled eyewear try-on system without having physically go to a store. With a much convenient and engaging shopping experienced that provided by AR which combines facial tracking with 3D rendering. This section will discuss a general introduction of augmented reality (AR), the technology with tools that are being used and how AR is technically used in virtual eyewear applications.

2.5.1 Overview to Augmented Reality

The meaning and idea of augmented reality is incorporating virtual content into 3D landscape. The augmented reality creates a more accurate and engaging visual representation (Luidolt & Zhou, 2023). When it comes to virtual try-on systems, augmented reality applications let users take a picture of a real object and then see it in real time on virtual avatar (Luidolt & Zhou, 2023). Hand free interaction and overlaying virtual objects on the physical world are made possible by augmented reality (Luidolt & Zhou, 2023). Virtual try-on systems that is replicate the experience of wearing items in virtual settings are something like an e-commerce application that is expanding quickly (Hu et al., 2023). There are two primary types of AR:

a) Marker-Based AR

   This method depends on particular visual indicator that sometimes called "AR tags" and it applied to actual things (Luidolt & Zhou, 2023). For instance, an AR application may transfer a pre-processed outfit onto a virtual avatar by using AR tags on clothing items. The portable augmented reality experience then renders this virtual item as a 3D model (Luidolt & Zhou, 2023). Sekhavat suggested an AR software that leverages a user's facial image and anthropometric data to create personalized avatars on smartphones. AR tags are then used to attach clothing (Luidolt & Zhou, 2023).

b) Marker less AR

   This kind of AR depends on real-time environmental awareness and mapping rather than particular physical markers (Luidolt & Zhou, 2023). To position and control virtual objects in the actual environment it usually applies advanced techniques like 6DOF (six degrees of freedom) tracking and layer identification of surfaces (such as the ground surface) (Luidolt & Zhou, 2023). For example, by placing a virtual human model on a detected ground surface it can spawn anywhere as a 3D augmentation of the real world that enabling genuine interaction (Luidolt & Zhou, 2023). With the ability to transmit objects from "any source without any pre-processing of the input image," markerless AR systems frequently provide more flexibility for real-world situations (Luidolt & Zhou, 2023).

2.5.2 AR Tools and Technologies

a) ARkit: Apple AR Development Platform

Apple released ARKit in 2017 as a robust software development kit (SDK) made especially for making augmented reality apps for iOS devices, such as iPhones and iPads (Ferrão et al., 2023). ARKit, one of the first marker-less AR SDKs as it uses visual-inertial odometry (VIO) to determine the exact camera position in the surroundings and offering more complex AR experiences than conventional marker-based solutions. ARKit provides a wide range of fundamental characteristics that are necessary for AR development, such as illumination estimates, motion tracking, and environmental awareness (Gupta et al., 2023). The platform's incredible precision in identifying both horizontal and vertical surfaces enables the realistic placement of virtual items in the actual environment (Lu et al., 2021). ARKit facilitates the building of AR applications by supporting powerful scene processing, device motion tracking, camera scene capture, and display conveniences (Gupta et al., 2023).

Since its first release, ARKit has undergone substantial development. ARKit 2 made it possible to follow 2D photos and identify typical 3D objects like toys, furniture, and sculptures (Magaia et al., 2021). Later iterations included more robust features including facial tracking, motion capture, occlusion, and joint sessions enabling multi-user experiences. Other capabilities that set ARKit apart from other rival systems are 3D object scanning and globe tracking (Borisova et al., 2021). The anchor system that is enables virtual objects to hold their location in the real world irrespective of camera orientation or device movement, is a crucial component of ARKit (Magaia et al., 2021).

Because it only works with iOS devices running iOS 11.0 or later, ARKit's primary drawback is its platform exclusivity (Kamalam et al., 2022). Additionally, only iOS devices with the A12 processor or later models (iPhone Xr and beyond) may use some of ARKit's more advanced functionalities, like motion capture and occlusion (Borisova et al., 2021). AR Foundation offers a unifying framework that enables AR experiences to run with both ARKit and

ARCore, greatly decreasing the development effort for multi-platform projects for developers looking for cross-platform connectivity (Zwoliński et al., 2023). ARKit is frequently integrated with game engines such as Unity3D and Unreal Engine 4 to streamline the development process (J. Li et al., 2024).

b) ARCore: Google AR Development Platform

Google introduced ARCore in 2018 as a framework for software developers to create augmented reality apps, mostly for Android smartphones (Chen et al., 2025). Because it works with major gaming engines and focuses on surface detection, the platform is a viable option for AR development (Vidal-Balea et al., 2021). The three primary functions of ARCore are motion tracking, environmental awareness, and light estimation (Hořejši et al., 2024). Using Simultaneous Localization and Mapping (SLAM) technology, motion tracking enables the platform to comprehend the device's position and orientation in the physical world (Alkady et al., 2024). While light estimation modifies virtual material to meet real-world lighting conditions, environmental understanding allows the recognition of horizontal, vertical, and angular surfaces (Borisova et al., 2021).

Through APIs that facilitate both Android and iOS development, ARCore provides cross-platform capabilities in contrast to ARKit, which is exclusive to iOS devices (Nikolarakis & Koutsabasis, 2024). Because of this, ARCore is a desirable choice for developers who want to expand their audience without repeating their work (Kakoutopoulos et al., 2025). Devices running iOS 11.0 or later and Android 7.0 or later can use ARCore, while some of its more sophisticated functions could need more recent hardware (Borisova et al., 2021). Cloud Anchors is a noteworthy feature of ARCore that facilitates collaborative and multiplayer AR experiences by allowing anchors and feature points to be shared across numerous devices (Cao et al., 2021). This feature makes it easier to create shared augmented reality environments where several people can interact with the same virtual objects at once.

According to performance evaluations, ARCore can map areas faster than ARKit since it tracks more feature points. On the other hand, ARKit has demonstrated improved precision in identifying both vertical and horizontal surfaces (Lu et al., 2021). AR Foundation offers a unifying framework that integrates with both ARCore and ARKit, greatly minimizing development effort for developers looking to construct cross-platform AR applications (Zwoliński et al., 2023). The creation process is streamlined by ARCore's smooth integration with well-known game engines like Unity3D and Unreal Engine 4 (J. Li et al., 2024). The platform is an accessible choice for AR development because it is free for developers.

c) Web-based AR Solutions

As an alternative to device-specific SDKs the web-based augmented reality (WebAR) enables users to see augmented reality material directly in web browsers without install software that needed. As good in accessibility, web-based AR is especially useful for expanding audiences across many operating systems and devices. One of the first and most popular frameworks for web-based augmented reality was AR.js. It was first released in 2017. AR.js, which is built on JavaScript and modified from the 2015 JavaScript version of ARToolKit has allows for marker-based augmented reality experiences in web browsers (Cao et al., 2021). With its remarkable performance that up to 60 frames per second on mobile devices. This lightweight solution is appropriate for a range of uses, including as educational resources (Russo, 2021). There is no requirement for specialist hardware or application installation because the framework simply requires browsers that implement WebGL and WebRTC JavaScript APIs (Cao et al., 2021).

There are numerous tools and methods accessible for developers looking to create web-based augmented reality experiences. To offer several approaches to presenting virtual content, AR.js can be expanded with other JavaScript frameworks like A-Frame and three.js (Cao et al., 2021). With the benefit of not requiring third-party plugins, HTML5 has also become a platform for web-based AR development (Syed et al., 2023). The development of WebXR standards has made it possible for markerless web-based augmented reality solutions to be used in both commercial and scientific purposes (Ferrão et al., 2023). These advancements have allowed web-based augmented reality to do more than just track markers. Furthermore, more complex web-based augmented reality experiences with enhanced object identification and image processing capabilities have been made possible by the integration of OpenCV, an open-source computer vision toolkit with JavaScript (Nikolarakis & Koutsabasis, 2024).

In a comparable manner, Mota et al. developed a web-based application that helps teachers produce educational AR content without the need for programming knowledge by using Scratch-based visual scripting (Blattgerste et al., 2023). Another example is SimpleAR, which was created by Apaza et al. and uses Blockly, a visual scripting tool based on JavaScript, to let users create interactive augmented reality content by fusing AR markers with 3D objects (Blattgerste et al., 2023).

Compared to device-specific SDKs, web-based AR systems have a number of advantages. Because users may experience AR using a conventional web browser on their portable device, they eliminate the need for an Integrated Development Environment (IDE) and suitable devices (Nguyen et al., 2020). Because web browser engines support HTML, CSS, and JavaScript, which can be coded using a basic text editor, this method greatly reduces the entry barrier for both developers and end users (Nguyen et al., 2020). However, compared to native SDKs like ARKit and ARCore, web-based AR solutions might provide fewer advanced capabilities, especially in areas like illumination estimation and environmental awareness.

d) Snap AR

Although ARKit and ARCore are the industry leaders in AR development for iOS and Android, there still many more platforms serve certain use cases and developer requirements. Creators may now create AR effects especially for the Snapchat platform with Snap AR, which was created by Snap Inc. Lens Studio for creating AR lenses and SnapML for incorporating machine learning into AR experiences are two of Snap's primary tools (Minaee et al., 2022). Another product from Snap Inc., Camera Kit, offers cross-platform resources for creating augmented reality experiences that can be accessed via Snapchat's camera.

Table 2.5 Comparison AR tools and Technologies

Criteria	ARKit	ARCore	Web-based AR	Snap AR (Lens studio)
Platform Compatibility	iOS only (iPhone, iPad with A12 or newer)	Android & limited iOS via ARCore SDK	Cross-platform (runs in mobile browsers)	Snapchat app (iOS & Android)
Hardware Requirement	High-end iOS only	Varies by Android device	None (only a browser & camera)	Mid-range smartphones with Snapchat
Cross-platform Development	iOS only	Android & partial iOS via Unity/ARCore	Web-based, all modern browsers	Locked to Snapchat ecosystem
ML integration (Face Shape)	With CoreML & Vision frameworks	With TensorFlow Lite or ML Kit	(via JS libraries like TensorFlow.js)	With SnapML (upload custom models)
Ease of Development	Medium (requires iOS/macOS & Xcode)	Medium (Android Studio + Unity support)	Easy to Moderate (Web dev + 3D tools)	Easy (Visual UI + ML import in Lens Studio)

Use Case Fit for project	Not suitable platform-locked, limited deployment	Not ideal Android focus, deployment complexity	Highly suitable browser-based, flexible ML/AR, no install needed	Limited restricted to Snapchat platform, not ideal for general web try-on

2.5.3 AR Application in Virtual Eyewear Try-On

In virtual eyewear applications, augmented reality (AR) improves the user experience by enabling real-time eyeglass try-on and offering contactless, interactive, and customized shopping or styling options.

a) Virtual Try-On System Work in Technical Approaches

One of the primary technical categories of virtual try-on systems for eyewear is 3D-based techniques. This approach has a unique way of projecting virtual glasses onto a user’s face. Using 3D modelling techniques is the most popular method for virtual eyewear try-on aligns 3D eyeglass models with the user's face. This system typically involves several key steps such as:

i. Face and Head Modelling
   Numerous methods use a single image or video feed to create a 3D model of the user's head (Joshi et al., 2024). This serves as a basis for precisely positioning eyeglasses.

ii. Head Pose Estimation
   To accurately place the virtual eyewear, systems monitor the user's head position and movements in real-time (J. Wang et al., 2023).

iii. Virtual Eyewear Overlaying

The predicted head posture is used to overlay the 3D model of the spectacles on the user's face (An et al., 2021).

iv. Rendering and Compositioning

The final representation is produced by the system rendering the virtual spectacles and combining them onto the user's picture or video feed (Martin-Brualla et al., 2020).

Large collections of eyewear may find it challenging to scale 3D-based methods since they require a distinct 3D model for every style of spectacles but they still can yield incredibly realistic results (J. Wang et al., 2023). Whatever the method, virtual try-on systems usually have three essential elements:

i. User Detection and Localization

To start the try-on process, the system must first detect a user's presence (Marelli et al., 2022).

ii. Feature Extraction

To precisely align the virtual eyewear, systems use methods such as SIFT (Scale-Invariant Feature Transform) to extract facial features (An et al., 2021).

iii. Processing Pipeline

Applications for virtual try-ons arrange their modules in a pipeline, with front-end modules controlling the user interface and back-end modules handling the computational components (Marelli et al., 2022).

To deliver a convincing and realistic virtual try-on experience, these systems need to overcome a number of technical obstacles like precise face tracking, good rendering, and responsive performance on devices with constrained processing power.

b) Virtual Try-On System Work in User Experience Flow

Usually, user will following a predetermined order flow in virtual eyewear try-on systems. This is made to smoothly in-person try-on process:

i. Initial User Capture and Detection

The first step in the process is for the system to locate and identify the user, typically via a camera feed on a smartphone, tablet, or dedicated in-store digital mirror (Marelli et al., 2022). This first phase usually starts by recognizing a face to start the try-on procedure. Then, establishing a baseline for tracking and analysis later on.

ii. Facial Analysis and Attribute Extraction

After identifying the user's face, the system examines a number of facial features that are essential for choosing and positioning eyewear correctly. This is to inform functional matching of eyewear to the user's face.

iii. Product Selection and Recommendation

Based on the facial analysis, the system then assists with product selection by searches through a database of eyewear models, which usually has thousands of possibilities.

iv. Interactive Visualization and Try-On

The interactive visualization that lets users view themselves using the virtual eyewear is the main component of the experience as real-time tracking enables the virtual eyewear to follow the user's head movements, allowing them to see how the glasses look from different angles (Marelli et al., 2022).

v. User Interaction

Users can engage with the system at any point along the process to help them make a buying decision because Users can compare styles by quickly switching between several eyewear models.

This well-organized flow adds the convenience and improved benefits of digital technology to a user experience that is simple and attractive, also simulating a physical try-on.

2.6 Related Work and Comparative Studies

Customers can now see how various frames would seem on their faces without physically trying them on thanks to virtual eyewear try-on technology. It is a major turn in the retail eyeglasses sector. This technology allows customers to try on different eyewear styles and fits via a virtual mirror. This is showing bridging gap between traditional in-store purchasing and internet shopping.

2.6.1 Commercial Applications of AR Eyewear Systems

Several companies have developed virtual eyewear try-on solutions to enhance online shopping experiences:

a) Ditto’s Virtual Try-On
   It offers a 3D application for sunglasses and eyeglasses that prioritizes precise sizing. It also fits the user's estimated facial proportions to a library of glasses. A credit card sized object put on the forehead help in estimating facial size and users must file a brief video while turning their face horizontally (Marelli et al., 2022).

b) Jeeliz
   This system provides a JavaScript widget that can be incorporated into websites or mobile web apps to enable real-time virtual try-ons for glasses. Although it has limits with regard to tracking speed and glass location, the application creates 3D models of glasses in real-time on live camera feeds (Marelli et al., 2022).

c) MemoMi's Memory Mirror

This is intended for usage in actual stores. It is apart from applications that are only available online. In front of a "magic mirror" that captures films of them sporting various accessories users can try on actual spectacles (Marelli et al., 2022).

2.6.2 Comparative Analysis of Face Detection Methods, Face Shape Classification, AR and Recommendation Techniques

Only academic research and openly accessible studies are included in this comparative summary table. Limited technical availability and private techniques that hinder a fair scientific comparison are the reasons why commercial applications are not included.

Table 2.6 Comparative Academic Research

Author	Face Detection Method	Face Shape Classification Method	Remarks
(De Luna et al., 2024)	Deep Transfer Learning	Convolutional Neural Networks - based classification (5 shapes)	AR try-on: No Strength: High accuracy using transfer learning Limitation: No AR Recommendation technique: Rule-based after classification
(Salim et al., 2023)	Swin Transformer model	Swin Transformer deep model	AR try-on: No Strength: Transformer based for higher accuracy Limitation: No AR and recommendation system Recommendation technique: No

(Suriyalakshmi et al., 2023)	Convolutional Neural Networks	Convolutional Neural Networks + Rule mapping	AR try-on: No Strength: Useful for eyewear suggestions Limitation: No AR and Recommendation technique: Rule-based
(Hu et al., 2023)	Convolutional Neural Networks / Deep Learning	Not classified by face shape	AR try-on: Yes Strength: Functional AR eyewear try-on Limitation: No face shape personalization Recommendation technique: ML-based (style matching)
(An et al., 2021)	AR Toolkit + 3D Modeling	Not applied (footwear focus)	AR try-on: Yes Strength: Real-time AR on mobile Limitation: Not for eyewear or face-related applications Recommendation technique: No
(Marelli et al., 2022)	YOLOv3	Not applied	AR try-on: Yes Strength: Browser-based AR with responsive UI Limitation: No classification or recommendation Recommendation technique: No

(Sagarika & Mokashi, 2024)	MTCNN	Convolutional Neural Networks (face + hairstyle features)	AR try-on: Yes Strength: Supports hairstyle + eyewear suggestion Limitation: Limited realism in AR output Recommendation technique: Rule-based (combined features)
My System0	RetinaFace	Convolutional Neural Networks (5 shapes)	AR try-on: Yes Strength: Integrated pipeline: detection → classification → suggestion→ AR Limitation: Pending evaluation, performance may vary with lighting/camera quality Recommendation technique: Rule-based (face shape– driven)

2.7 Summary

In this chapter, the essential elements of creating a virtual a virtual eyewear try-on system were been discussed. It began with the face landmark extraction and face detection which are crucial for recognizing facial structure. Afterwards, it covers the topic of face shape categorization approaches which are used to customize the recommendations regarding to particular characteristics. There is also highlighted of tools and uses of augmented reality (AR) as a major enabler of virtual try-on. In order to better understand the current idea and pinpoint areas for development, recent commercial solutions were been evaluate. This chapter finding the serve as basis for the system design and technique that are discussed in the following chapters.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Research Methodology Framework

This study uses a phases-based methodology framework that consist of three primary objectives. Each phase requires particular tasks to deliverables that are intended. The first step is to review the literature on face detection, face shape classification and AR try-on systems. Facial image datasets are being retrieve from open source and it is being prepared during the data collection and preprocessing stage. A CNN architecture is used to create, train and assess a face shape classification model. The recommendation is based on rules where the system phase use to established style matching guidelines to map face shapes to appropriate eyewear styles. Then the system integration is existed to process the user interface and recommendation logic are connected to the classification results. For real time visualization, the Web AR development phase use library from JavaScript to overlay eyewear models. Every stage produces different results that support the overall objective of a customized virtual eyewear try-on system.

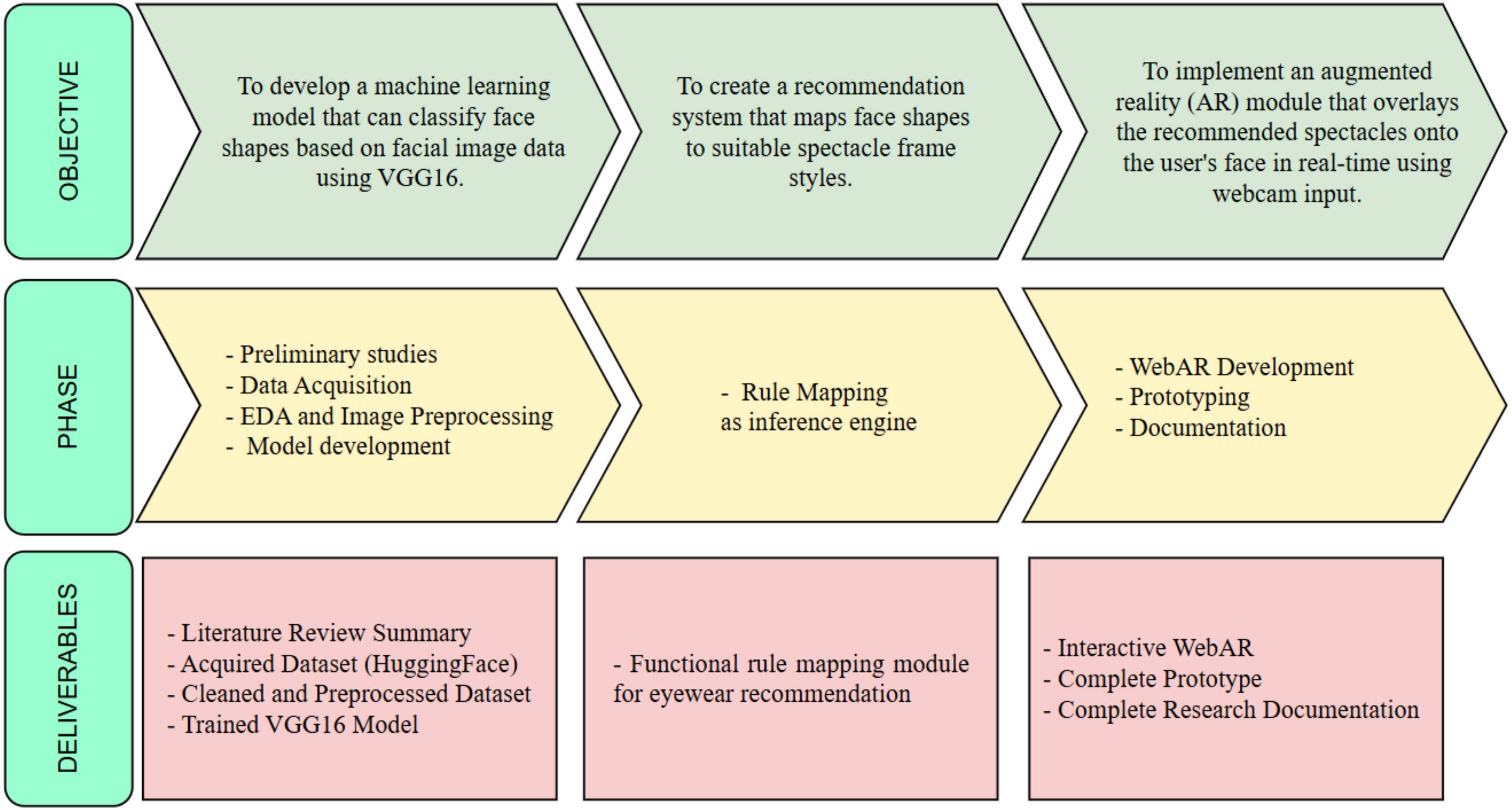


Figure 3.1 Research Framework

Table 3.1 Research Methodology

Objectives	Phases	Activity	Method/Tool	Deliverable
To develop a machine learning model that can classify face shapes based using VGG16	Preliminary Studies	Review the literature on CNN architectures which is VGG16, AR eyewear systems, and facial shape categorization. Determine the gaps and support the model selection.	Literature review, Conceptual Study and interview	Summary of related studies, identified research gaps, and theoretical framework.
	Knowledge and Data Acquisition	Acquire dataset from Hugging Face https://huggingface.co/datasets/bkprocovid19/face_shape . Explore dataset labelling and distribution. Collect technical knowledge on required Python libraries.	Hugging face datasets, TensorFlow, OpenCV, Pandas, NumPy	Verified dataset and configured environment ready for EDA and preprocessing.
	EDA and Image Preprocessing	Analysed dataset structure, class distribution, image dimensions, and aspect ratios. Perform image resizing, normalization, and augmentation.	Python Matplotlib, Seaborn, OpenCV, ImageDataGenerator	Clean and preprocesses dataset ready for training.
	Model Development & Model Evaluation	Implement and train the VGG16 model. Evaluate using accuracy, precision, recall, and F1-score.	TensorFlow / Keras, VGG16 pretrained weights	Trained VGG16 model for face shape classification.

To create a recommendation system that maps face shapes to suitable spectacle frame styles.	Rule Mapping as inference engine	Define logical mapping between face shape categories and suitable eyewear styles. Implement a rule-based recommender script that contains 38 rules.	Python (Flask backend), JSON for rules	Functional rule mapping module for eyewear recommendation.
To implement an augmented reality (AR) module that overlays the recommended spectacles onto the user's face in real-time using webcam input.	WebAR Development & Integration	Develop WebAR interface for real-time eyewear visualization using AR.js and 3D models. Implement face tracking and alignment.	HTML, CSS, JavaScript, AR.js, Three.js, MindAR	Interactive WebAR eyewear try-on module
	Prototyping	Integrate the Flask backend, VGG16 model, and WebAR interface into a unified system, followed by debugging and UI refinement to ensure a smooth user workflow.	Flask API, JSON communication, code Optimization, Browser Developer Tools.	Fully functional integrated system and finalized prototype for demonstration.
	Documentation	- Prepare final report and appendix	Microsoft Word / Google Docs	Final academic report and appendix

3.2 Preliminary Studies

The proposed face shape-based virtual eyewear try-on system underwent a set of preliminary studies prior to the system development as a way of getting familiar with the underlying concepts, feasibility, and technical requirements of the system. This step was to research the available solutions, to name possible obstacles and to select the appropriate tools, frameworks and data sets to use in the project.

The analysis of the existing literature and commercial solutions demonstrated that the majority of virtual try-on apps are based on the use of advanced face recognition, landmark tracking and 3D overlay technologies. Nevertheless, most of them are restricted to mobile applications or require expensive proprietary AR SDKs. This finding prompted the use of web-based augmented reality WebAR as a light and easy to use solution that can be used to visualize eyewear directly in a browser.

Models like VGG16 were discussed in order to find one that would suitably classify face shapes. VGG16 architecture has been selected because of its established success in the feature extraction task and its suitability to the transfer learning methods.

The outcome of this stage provided clarity on system design decisions, including the use of Flask for backend API integration, AR.js for browser-based 3D rendering and rule-based recommender for eyewear suggestion mapping. The findings from these studies guided the next phase which is knowledge and data acquisition where it goes more in-depth technical preparation and dataset collection were carried out.

3.3 Knowledge and Data Acquisition

The significance in this phase was to collect theoretical and empirical data that would be used in the development of the virtual eyewear try-on system that is based on face shapes. The knowledge acquisition entailed broad literature and technical reading to do with computer vision, facial geometry recognition, Convolutional Neural

Networks CNN and Augmented Reality AR visualization methods. Different online academic materials such as IEEE Xplore, ScienceDirect, ACM Digital Library and Hugging Face repositories were researched to learn about the principles of face detection, facial landmark extraction and virtual object overlay. This allowed a good basis on both aesthetic mapping between faces and eyewear designs and the algorithmic chain of work concerning feature extraction and classification.

To collect data, this research uses the Face Shape Dataset that is provided as a free open-source resource on Hugging Face (bkprocovid19/face_shape). The dataset has about 5000 facial images spread evenly in five classes Heart, Oblong, Oval, Round and square, each having 1000 samples. All the images have pre-labels in place, which guarantees consistency in data and lessens the necessity of manual labelling. However, the dataset may restrict its representation across gender as it is limited to female subjects only.

The set was downloaded and saved in an organized directory format which could be used in the TensorFlow image data streamline. To handle the loading, visualization and augmentation of data, several Python libraries: OpenCV, NumPy, TensorFlow, and Matplotlib were used. A controlled environment of Google Drive and Google Collab was used to store the dataset and preprocessing scripts and perform efficient experimentation and version control.

The combination of the literature knowledge and data acquisition using the Hugging Face shape dataset made this step the foundation of the following Exploratory Data Analysis (EDA) and Image Preprocessing stages so that the further analysis and preparation operations could be performed in an effective manner.

3.4 Exploratory Data Analysis and Image Preprocessing

In order to prepare the photos for training the face shape classification model, this step included investigating the dataset properties. Image characteristics like class balance, resolution fluctuation and aspect ratio distribution were revealed during the exploratory data analysis (EDA) stage. In order to guarantee data consistency and robustness prior to model training, the preprocessing stage normalized all photos through methodical scaling, normalization, and augmentation.

3.4.1 Exploratory Data Analysis (EDA)

Exploratory data analysis was performed using Python libraries such as Matplotlib, Seaborn, NumPy, Pandas, and OpenCV to understand the characteristics of the Hugging Face “bkprocovid19/face_shape” dataset. The dataset from Figure 3.2 and 3.3 show it already came with a train-test split that contained 4,000 images for training and 1,000 images for testing. Each equally distributed across five face shape categories such as Oval, Round, Square, Heart, and Oblong for 1,000 per class.

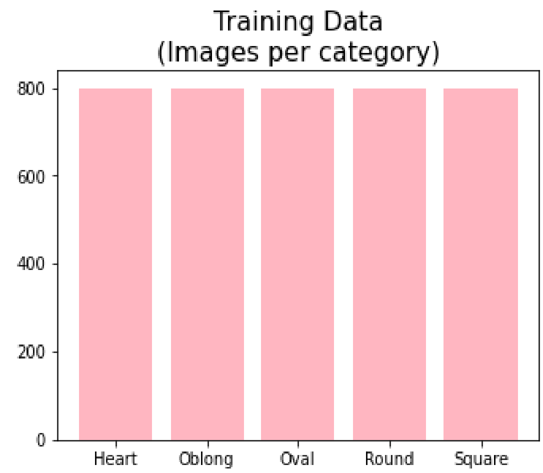


Figure 3.2 Distribution Dataset Training

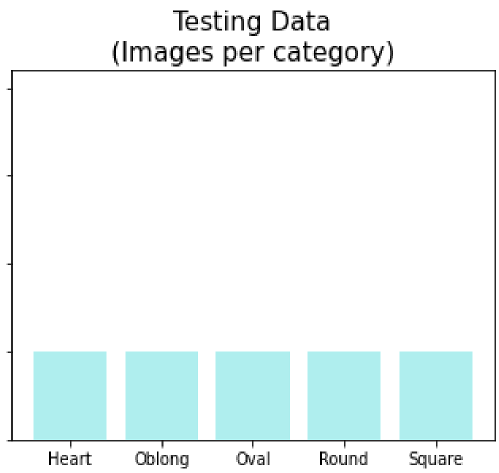


Figure 3.3 Distribution Dataset Testing

To begin the analysis, image dimensions were extracted and stored in an array to observe overall variability. A Pandas Data Frame was then created to summarize and inspect image properties such as height, width, and aspect ratio. Descriptive statistics and histograms were generated to visualize the distribution of image sizes and aspect ratio categories (portrait <1, square =1, landscape >1).

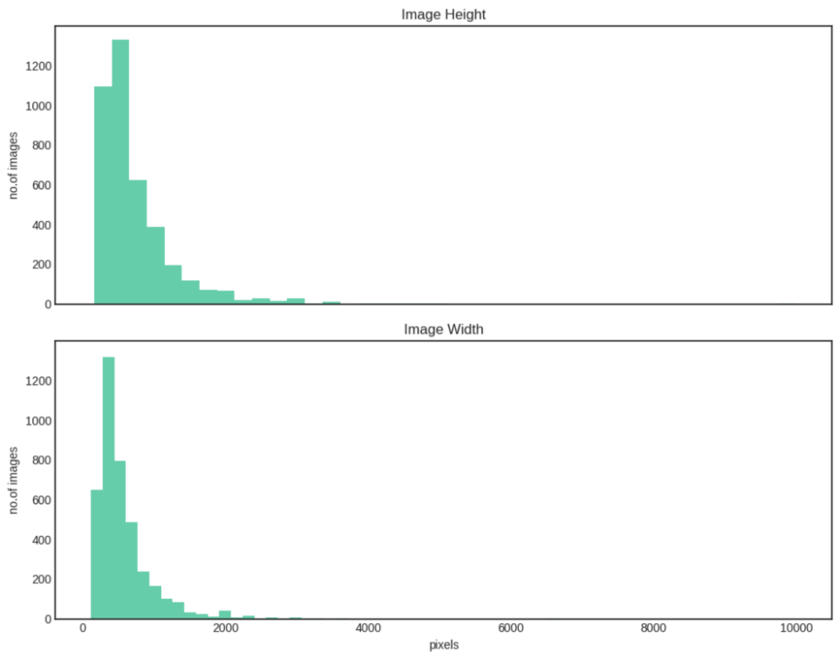


Figure 3.4 Distribution of Image Height and Width

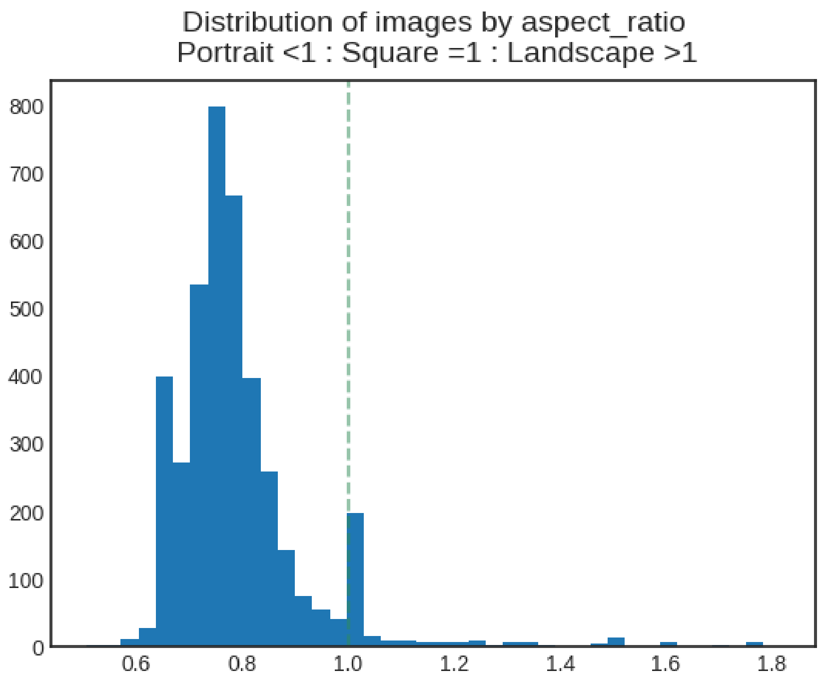


Figure 3.5 Distribution of Image Aspect Ratio

The Figure 3.4 & 3.5 above shows distribution of image heights, widths and aspect ratios in the dataset. The Figure 3.4 show a right skewed distribution with most images having resolutions between 200 to 800 pixels, and a small number of outliers with resolutions significantly exceeding 2000 pixels. There, figure 3.5 shows that most of the images are of portrait type (ratio < 1) and few of the outliers have a very large resolution. This implies that a majority of the pictures will be tall as compared to their width. Since the VGG16 model requires square input size (224 x 224), such an imbalance makes it essential to have a good preprocessing pipeline such as the one that resizes and pads image dimensions to eliminate aspect ratio distortion when training the model.

Overall, due to this non-uniformity it is important to have a robust preprocessing image dimension resizing and padding to avoid aspect ratio distortion during training. Also, the EDA process ensured data integrity, since no corrupted records were found, which made the dataset clean and declared it fit to be processed further. These were the direct implications on the design of the preprocessing steps in a way that would make the models ready.

3.4.2 Image Preprocessing

A structured preprocessing pipeline was created after the exploratory analysis to normalize the image data for best compatibility with the VGG16 model that would be utilized for training. Preparing the training and testing sets while maintaining significant facial traits was the main goal of the procedures.

a) Face Detection with Bounding Box

To exclude unnecessary background information, the region of interest (face) was retrieved using OpenCV's face detection feature. For testing, processed datasets were produced from a version of RGB photos with faces detected used for training.

b) Cropping and Resizing

To avoid distortion, each image was cropped and reduced to 224 by 224 pixels as it the required input dimension for VGG16, while maintaining its original aspect ratio. This made sure that every sample had the same scale and visual quality.

->

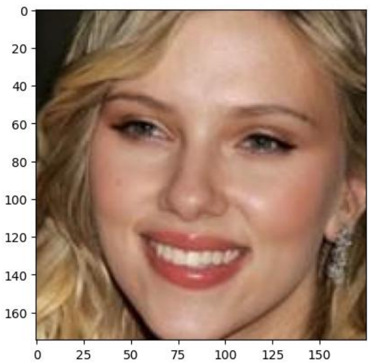


Figure 3.7 Image After Crop

Figure 3.6 Image Before Crop

A comparison of preprocesses output figure 3.7 and the original input image figure 3.6. To fulfil VGG16 input standard while preventing aspect ratio distortion, the image is cropped to centre the face and shrunk to 224 x 224 pixels.

c) Data Augmentation

Only the training dataset was subjected to augmentation techniques such horizontal flipping and rotation modifications in order to improve generalization and decrease overfitting. This procedure increased the variety of visuals and strengthened the model resilience to changes in the real world.

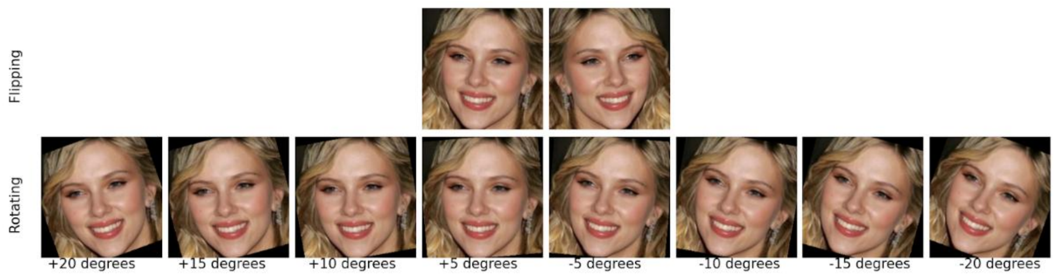


Figure 3.8 Augmentation Image

To mimic natural head movements and strengthen the model resistance to unobserved real-world, figure 3.8 show the training images were subjected to horizontal flipping and controlled rotation in range of -20 to + 20 degrees.

There was no need for further manual separation because the Hugging Face dataset was already segmented of ratio 80/20. Before developing the model, the consistency of the training and testing sets was confirmed.

3.5 Model Development

The face shape classification module was developed using a transfer learning approach of VGG16 convolutional neural network (CNN) as the base model. Transfer learning was chosen to leverage the pretrained weights from the VGGFace dataset, which contains large-scale facial image representations that capture essential facial features and structure. This allowed faster convergence and reduced the need for extensive training data.

3.5.1 Dataset Preparation

To assess model generalization, the preprocesses dataset was split into training and testing sets of 80/20. To comply with VGG16 input specifications, all input photos were reduced to 224 x 224 x 3 pixels. Python pickle package was used to load and save image data in NumPy arrays. An ImageDataGenerator with rotation +20°/-20° and horizontal flipping augmentation was used to improve the resilience of the model. In this way, overfitting can be avoided and facial differences were simulated.

3.5.2 Dataset Preparation

The pretrained VGG16 model was used as the feature extractor, where all convolutional layers were frozen to retain learned facial features. The fully connected layers which are top layers were replaced with custom classification head that consisting of:

- A flatten layer for dimensionality reduction.
- A dense layer with 64 neurons and ReLU activation.
- A dropout layer with rate 0.5 to reduce overfitting
- A final dense layer with 5 neurons and SoftMax activation to predict five face shape classes which are Heart, Oblong, Oval, Round and square.

3.5.3 Model Configuration

For multi-class classification problems, the model was put together using the Adam optimizer with a categorical cross-entropy loss function. The following guidelines were used during the training process:

- Batch Size: 32
- Epochs: 50
- Optimizer: Adam
- Evaluation metric: Accuracy and Loss

During training, both training and validation accuracy/loss were monitored per epoch using Matplotlib visualization. The model achieved consistent convergence, indicating effective feature extraction and classification capability.

3.6 Model Evaluation

The test set was evaluated by performance from a confusion matrix which is used to display a visualization of the distribution of the classification among all the classes of the face shapes. Measures like accuracy, precision, recall and F1-score were obtained in order to evaluate the general performance. The model was found to have good discriminative capability especially when distinguishing between geometrically different classes such as round and square faces.

3.7 Rule Mapping as Inference Engine

The rule mapping technique in this project serves as the core decision making structure that relates the face shape classification result with the appropriate eyewear recommendation. After the model predicts the user's face shape category, the system applied a rule-based approach to determine which eyewear styles best complement the facial geometry based on aesthetic design principles between face shapes and eyewear structure. This method was selected because of its interpretability, simplicity and transparency, which provide deterministic and explicable results without depending on extensive user preference data.

3.7.1 Design of Rule Mapping

The principles had been developed depending on the combination of eyeglasses frame geometry and facial features. As per literature on optical styling principles and fashion-based suggestions, contrasting shapes would tend to create facial balance. Square faces are also less harsh and rounded which is ideal with the square faces, round faces are also ideal with angular faces. Table 3.2 and Figure 3.9 is the summary of the association between each face shape and the eyewear recommendations, which are used in this study.

Table 3.2 Summary of Eyewear with Face Shape

Face Shape	Recommended Eyewear Style	Rationale
Heart	Cat Eye, Rectangle, Wayfare, Square	Balances a wide forehead and narrow chin by adding width to the lower face.
Oblong	Round, Cat Eye, Wayfare, Aviators, Geometric	Adds volume to narrow and long faces, creating a balanced proportion.
Oval	Round, Cat Eye, Rectangle, Wayfare, Square, Aviators, Geometric, Browline, Oval	Considered balanced and versatile; suitable for most eyewear styles.
Round	Rectangle, Wayfare, Square, Geometric, Browline	Angular shapes help elongate and define soft facial contours.
Square	Round, Cat Eye, Wayfare, Aviators, Geometric, Oval	Softer and curved frames smooth out sharp jawlines and strong angles.

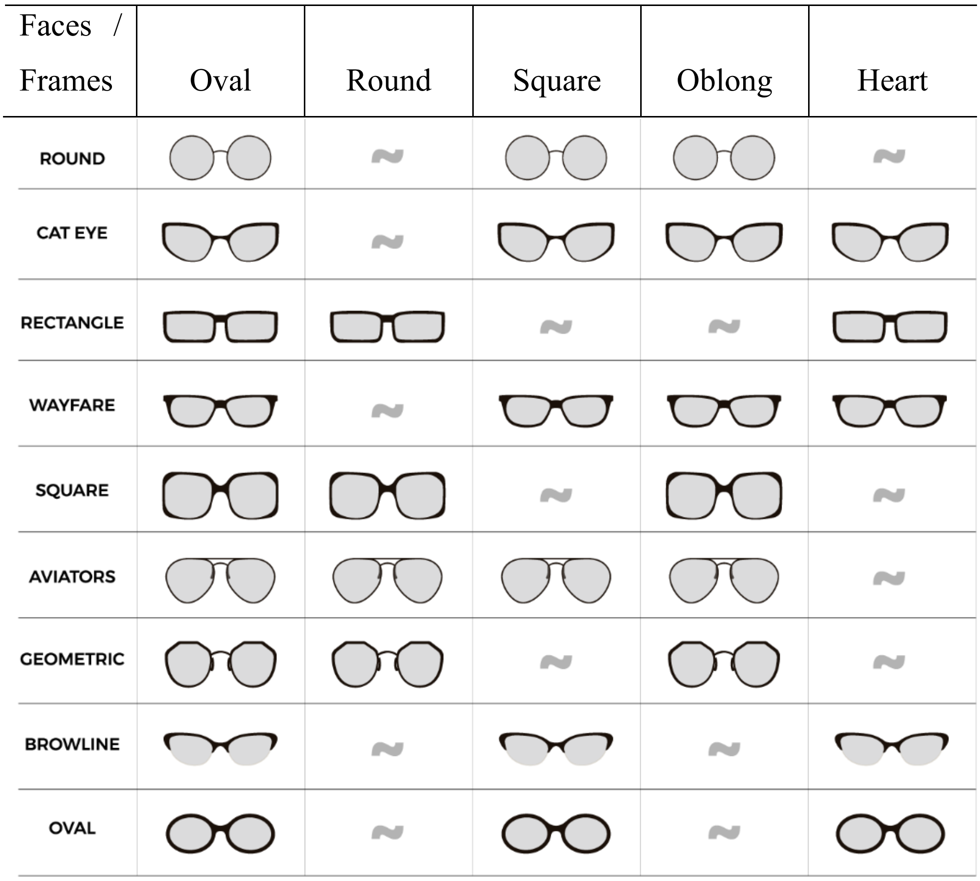


Figure 3.9 Eyewear with Face Shape

3.7.2 Implementation

The backend module (backend/recommender.py) implements the rule mapping as a Python dictionary. A list of suggested eyewear styles is the value of each key in the dictionary, which reflects a face shape label predicted by the CNN classifier. The system uses the get_recommendations () function, which takes the anticipated face shape as input and returns the matching list of eyewear kinds, to retrieve the recommended eyewear. In order to preserve consistent system behaviour, the function makes sure that a default list ("classic", "wayfarer") is supplied if no face shape is detected. The structure is demonstrated by the code sample that follows:

```python
# backend/recommender.py
RECOMMENDATIONS = {
    "heart": ["Cat Eye","Rectangle","Wayfare","Square","Browline"],
    "oblong": ["Round","Cat Eye","Wayfare","Aviators","Geometric"],
    "oval": ["Round","Cat Eye","Rectangle","Wayfare","Square","Aviators","Geometric","Browline","Oval"],
    "round": [ "Rectangle","Wayfare","Square","Geometric","Browline"],
    "square": ["Round","Cat Eye","Wayfare","Aviators","Geometric","Oval"]
}

def get_recommendations(face_shape):
    if not face_shape:
        return []
    key = face_shape.lower()
    recs = RECOMMENDATIONS.get(key, ["classic", "wayfarer"])
    return recs
```

Figure 3.10 Snippet Illustrates the Structure

This modular implementation makes scalability and maintenance simple. Without changing the design of the system, the lexicon can be expanded to incorporate other face shapes or eyewear types.

3.8 WebAR Development

The WebAR (Web-based Augmented Reality) module allows users to virtually wear eyewear with the use of a web browser and does not require any third-party software to be installed. This section will incorporate real time facial tracking with 3D eyewear models, to give the consumer a realistic and interactive try-on experience. The frameworks used in the implementation are JavaScript frameworks with support of 3D rendering, camera-based tracking, and model alignment, like Three.js and AR.js. The solution will be platform neutral and highly accessible by the virtue of the fact that WebAR is greatly accessible.

3.8.1 3D Eyewear Model Preparation

To guarantee effective rendering and WebAR compatibility, the 3D eyewear models were created in the .glb and gltf formats. In order to better fit the intended eyewear designs, the models were acquired from free internet 3D repositories that offer open-licensed components. The eyewear recommendations produced by the rule mapping technique are represented by each model. By lowering the number of polygons and superfluous texture complexity, the models were optimized to strike a compromise between performance and visual quality.

Using straightforward web-based tools and online model editors, each 3D model was modified for appropriate scale, rotation, and alignment with respect to the user's face landmarks to guarantee good fitting and visualization during AR rendering. The /models/ directory contains all completed eyewear files, which are dynamically imported into the interface based on the expected face shape and the recommendation that goes with them.

3.8.2 Face Tracking and Model Alignment

This is based on the fact that the system used real-time face tracking to position the eyewear model on the facial part of the user accurately. Here the facial landmarks are identified through AR.js face tracking with MediaPipe, FaceMesh to recognize facial landmarks like the ears, nose bridge and the eyes. It is these landmarks that determine the proper scale, rotation and translation of the 3D eyewear model. The alignment algorithm makes the eyewear track the head motions keeping the position in a realistic way as the user tilts or turns their head in front of the camera.

MindAR was incorporated in this project to provide the effective and precise real-time tracking of facial landmarks in the browser. It identifies several significant facial points and a 3D mesh is created, which represents the face structure of the user. The nose bridge landmark is where the eyeglasses models are located so that it can remain in place regardless of the orientation of the face. By this mechanism, the transformation matrix of the 3D model is dynamically updated so that the virtual glasses remain properly aligned even as the user moves forward, shifts their head or changes expression to create a natural and immersive AR visualization.

3.8.3 Rendering and Visualization

A-Frame is rendered and visualized with the MindAR face-tracking framework using this module. MindAR provides useful real-time facial landmark recognition, which stabilizes the 3D eyewear models on the face of the user. A-frame provides a declarative interface to creating 3D and AR experience on the web directly using HTML syntax. This mixture will remove the requirement of native mobile applications and allow a lightweight and completely web-based AR try-on experience.

The system loads each eyewear model which is stored in the /glasses/ directory in the .glb or .gltf format. Then, pre-loaded by using the a-asset-item tag. The virtual model is then placed in the correspondence with the identified facial anchor point which surrounds the nose bridge area through a number of a-entity elements each of which has mind-ar-face-target.

```
<a-scene mindar-face="autoStart: false" embedded>
  <a-assets>
    <a-asset-item id="wayfareModel" src="glasses/wayfare_glasses.glb"></a-asset-item>
  </a-assets>

  <a-camera active="false" position="0 0 0"></a-camera>

  <a-entity mindar-face-target="anchorIndex:168">
    <a-gltf-model
      src="#wayfareModel"
      position="0 -0.1 0.08"
      scale="0.004 0.004 0.004"
      rotation="0 0 0"
      visible="false">
    </a-gltf-model>
  </a-entity>
</a-scene>
```

Figure 3.11 Snippet for AR rendering scene

3.9 Prototyping

The last step of the development process is the system prototyping stage, during which the separately developed components, the VGG16 classification model, Flask backend and WebAR interface will be implemented into one architecture. This part describes how the system will be organized in its structure, how the modules will interact with each other as they operate, and the functional verification of the system that will be carried out to ensure that the prototype is efficient and ready for demonstration.

3.9.1 System Workflow

After satisfactory training results, the finalized models were saved in keras format and later converted into TensorFlow Lite (TFLite) for web development. This lightweight version was integrated into the backend for real-time inference in the WebAR environment, enabling prediction of a user face shape before eyewear recommendation and AR visualization.

The backend, model inference, rule-based recommendation, and frontend augmented reality display are all connected by the integrated system's sequential working process. In order to provide real-time interactivity and output consistency, this kind of workflow is required.

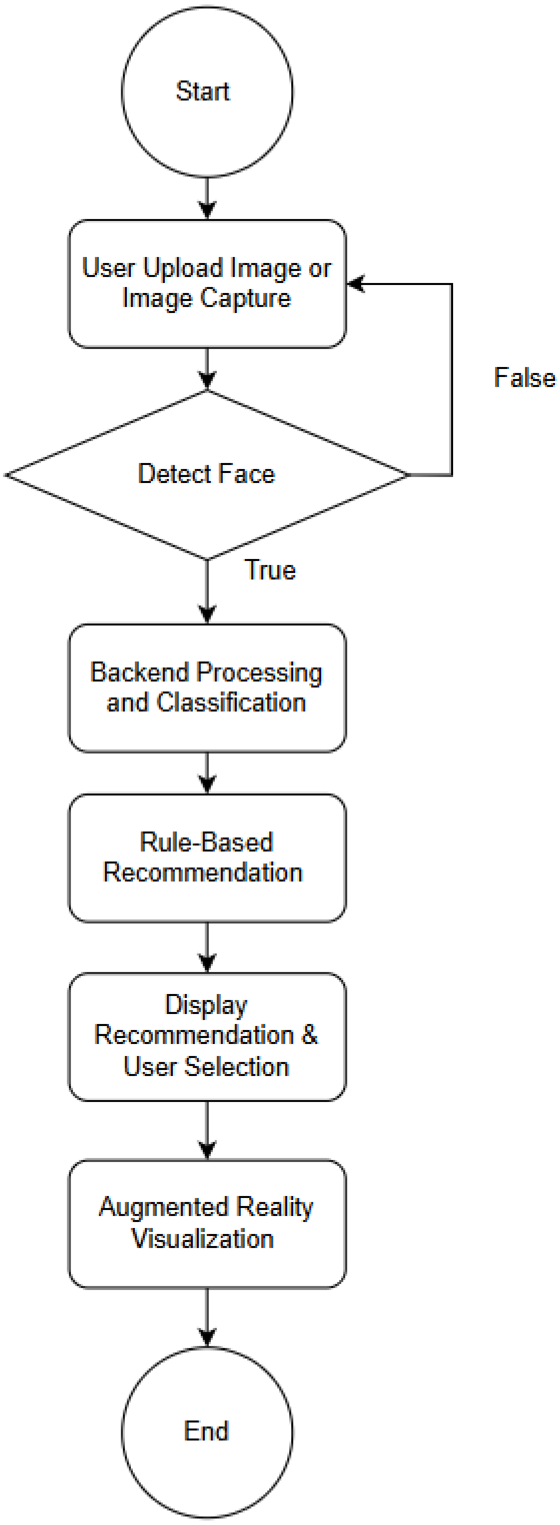


Figure 3.12 Prototype Workflow

In summary, this workflow demonstrates how the classification, rule-based logic, and web-based AR technologies are aligned to provide an experience of a seamless and interactive process of eyewear recommendation.

3.9.2 System Directory Structure

The backend, frontend, model and augmented reality components are all combined into a coherent and fully functional architecture during the system integration phase and it shown in table 3.3. The src directory, which contains the frontend and backend subsystems, is where the complete project is arranged. The integration guarantees smooth communication between the WebAR interface for real-time eyeglasses viewing, the face shape classification model, and the Flask-based backend. The backend is located in the backend folder and serves as the central controller for processing requests and managing communication between modules. It includes three main scripts such as face_shape_classifier.py, recommender.py and app.py.

The frontend uses a POST request to deliver the data to the Flask backend when a user uploads or takes a picture of their face through the browser. The image is processed by the backend, which then uses the CNN model to classify the face shape, fetches the relevant recommendations and reacts to the frontend interface. The projected face shape and suggested eyeglasses are then dynamically shown by the frontend, which is housed in the frontend directory. The frontend is built using HTML, CSS, and JavaScript, with integration of AR.js for real-time 3D model rendering. It includes files such as index.html, tryon.html, script.js, style.css.

The models folder stores the TensorFlow version of the VGG16 model face_shape_model.keras used during inference, while the glasses folder contains 3D eyewear assets that are rendered in the AR environment.

Table 3.3 System Directory in src folder

Folder	Description
backend/	Flask backend containing classifier, recommender, and API server scripts
frontend/	Web interface built using HTML, CSS, JS, and AR.js for AR visualization
models/	Stores pretrained VGG16 model in Keras format

glasses/	Contains 3D eyewear models used for AR try-on rendering

3.10 Documentation

Documentation shown in table 3.4 is the final phase of this project involves compiling comprehensive documentation. This includes the preparation of system architecture Figures, code documentation and testing reports. Additionally, a final report summarizing the development process, methodologies used, and project outcomes was produced to meet academic requirements. All documentation materials were structured to facilitate maintenance, reproducibility, and potential future enhancement.

Table 3.4 Details of Documentation Phase

Phases	Activity	Method/Tool	Deliverable
Documentation	Produce final report for research	Microsoft Word, Mendeley Reference Manager	Final Year Project Report

3.11 Summary

This chapter detailed the overall methodology used to develop the face shape–based virtual eyewear try-on system. It began with preliminary studies and knowledge acquisition to establish technical feasibility, followed by image preprocessing and CNN-based model development for accurate face shape classification. A rule mapping technique was then applied to connect classification outputs with appropriate eyewear recommendations. The WebAR module enabled real-time visualization, while system integration ensured smooth interaction between components. Finally, the system prototyping phase verified the functional integration of all components to ensure a stable and operational prototype and all documentation processes were completed to ensure maintainability and reproducibility of the project.

CHAPTER FOUR

RESULT AND ANALYSIS

4.1 Face Shape Classification Experiments

This part presents the experimental investigation as shown in table 4.1 for assess the VGG16 model face shape classification performance. An 80:20 split ratio was used to divide the dataset so that 80% of the images were used for model training and the remaining 20% were set aside for validation in order to guarantee a reliable evaluation. The aim was to track the model learning behaviour, paying attention to accuracy gains and loss convergence over time. Additionally, hyperparameters such as the optimizer, number of epochs and batch size were configured to ensure training stability.

Table 4.1 Experimental Setup

Phase	Objective	Experiment No.	Optimizer	Epoch	Batch Size
Baseline	Observe learning convergence and overfitting tendency	1	Adam	50	32
Optimizer	Identify the most suitable optimizer	2	SGD	50	32
		3	RMSprop	50	32
Epoch	Determine optimal number of epochs	4	RMSprop	70	32
		5	RMSprop	85	32
		6	RMSprop	100	32
Batch Size	Analyze effect of batch size on stability	7	RMSprop	50	16
		8	RMSprop	50	64
		9	RMSprop	50	128

a) Experiment No. 1

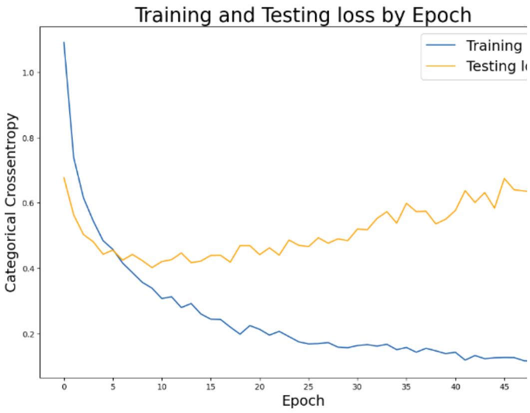


Figure 4.1 Training & Testing Loss Experiment 1

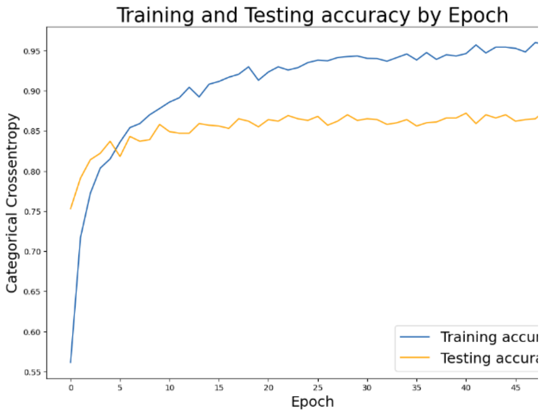


Figure 4.2 Training & Testing Accuracy Experiment 1

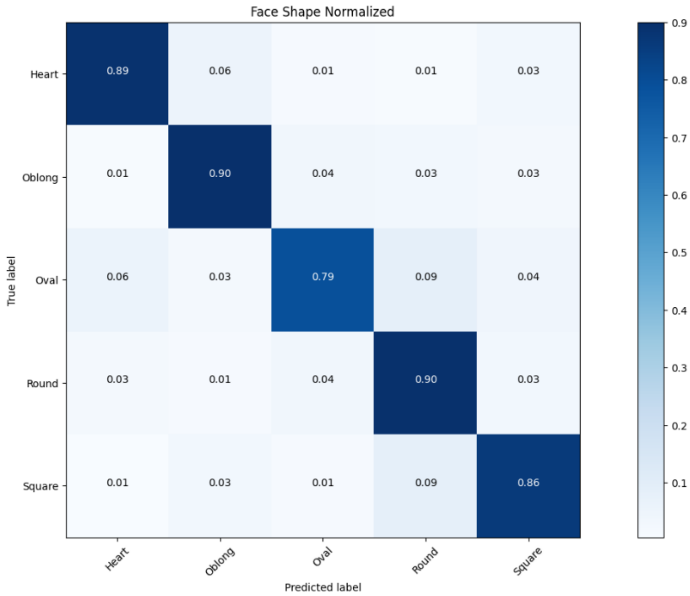


Figure 4.3 Confusion Matrix Experiment 1

From figure 4.1 The loss curve reveals that the training loss (blue curve) is dropping extremely fast, meaning that the model learns fast and the testing loss (yellow curve) again starts rising after about Epoch 10-15, which is an early indicator of overfitting where the model starts to memorize the training data and its performance on unseen data starts to decline. From figure 4.2 the training accuracy (blue) in the accuracy graph is nearly 95 percent (0.95) which is much higher than the testing accuracy (yellow) which is approximately 85-87 percent and the high difference between the two curves further proves the occurrence of overfitting. According to the confusion matrix from figure 4.3, the model is weakest when dealing with a shape of the Oval face (0.79) as it is often confused with a shape of a Round (0.09) or Square (0.04), whereas the Oblong face shape is detected the most well with an accuracy of 0.90.

b) Experiment No. 2

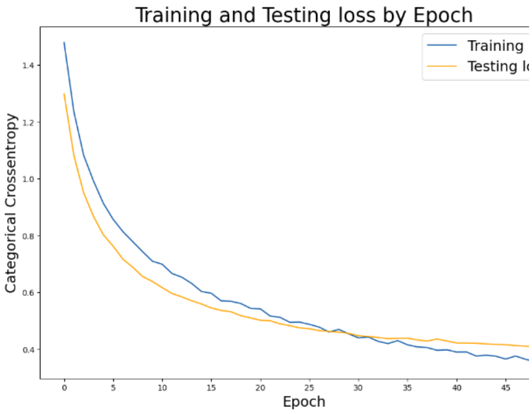


Figure 4.4 Training & Testing Loss Experiment 2

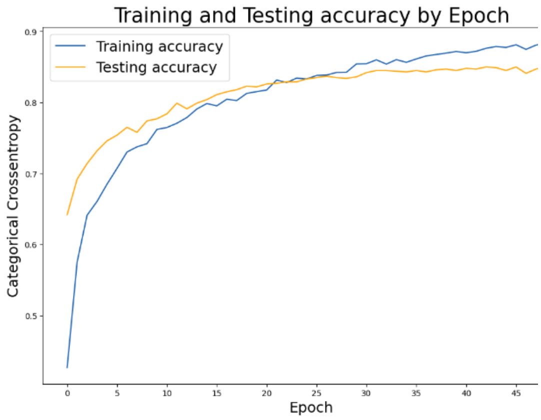


Figure 4.5 Training & Testing Accuracy Experiment 2

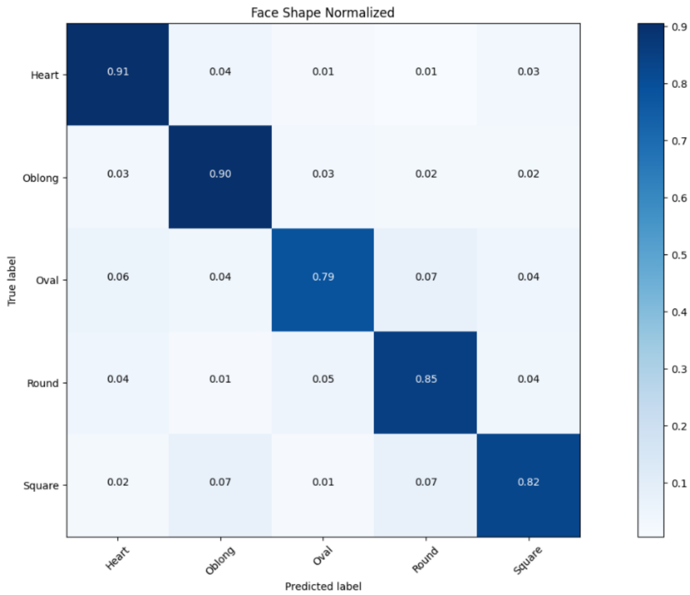
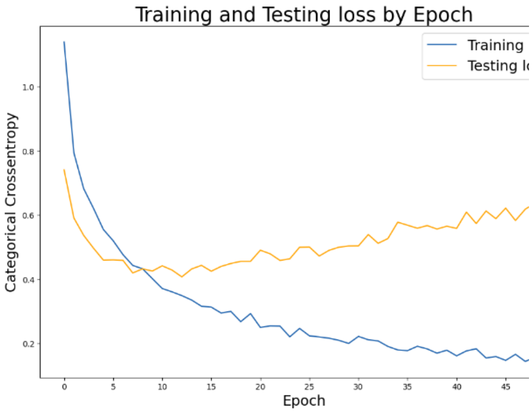


Figure 4.6 Confusion Matrix Experiment 2

The loss curve from figure 4.4 exhibits a lower and more variable rate of change than Adam, and the loss value still is quite large (approximately 0.5-0.6) at Epoch 50, meaning that SGD takes longer to converge. In the accuracy graph from figure 4.5, the accuracy is very slow in the initial stages of training and the ultimate accuracy is not as high as in Experiment 1 which was about 0.80-0.85 when it comes to training and testing accuracy. Based on the confusion matrix from figure 4.6, the performance of the Square face shape decreases to 0.82 (it has been 0.86 in Experiment 1), and the performance of the Heart face shape increases a bit to 0.91.

c) Experiment No. 3



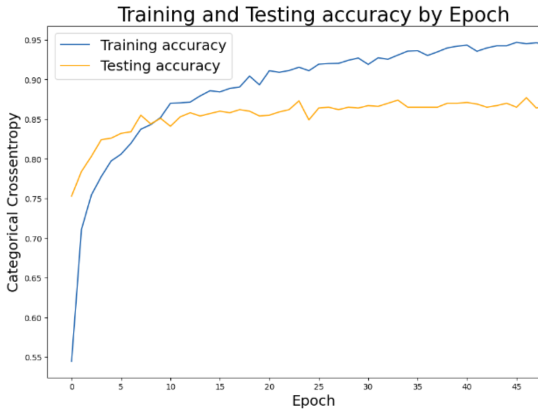


Figure 4.7 Training & Testing Loss Experiment 3

Figure 4.8 Training & Testing Accuracy Experiment 3

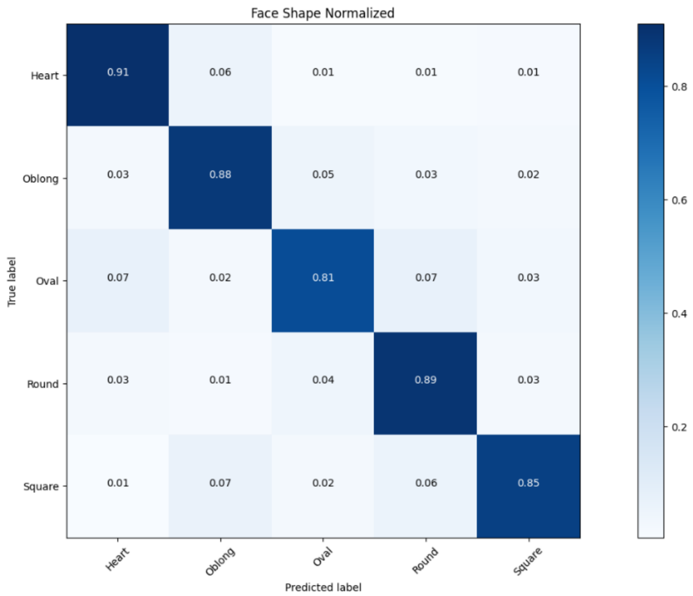


Figure 4.9 Confusion Matrix Experiment 3

The loss plot from figure 4.7 demonstrates a gradual decline but the testing loss (yellow line) has observable variations and this is a typical behavior of the more aggressive update nature of RMSprop. In the accuracy graph from figure 4.8, the model attains high accuracy within a shorter time than SGD, and the difference between the training and test accuracy is more regulated than Adam. According to the confusion matrix from figure 4.9, the performance of the Oval face shape increases to 0.81, and the Round (0.89) and Heart (0.91) face shape are also characterized by the high accuracy.

d) Experiment No. 4

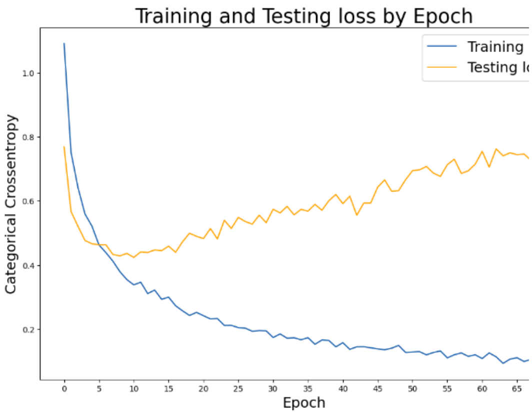


Figure 4.10 Training & Testing Loss Experiment 4

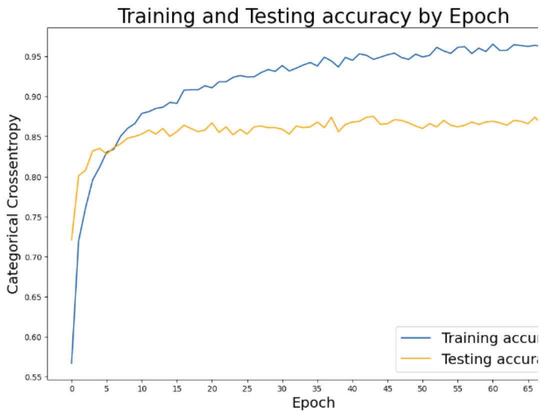


Figure 4.11 Training & Testing Accurracy Experiment 4

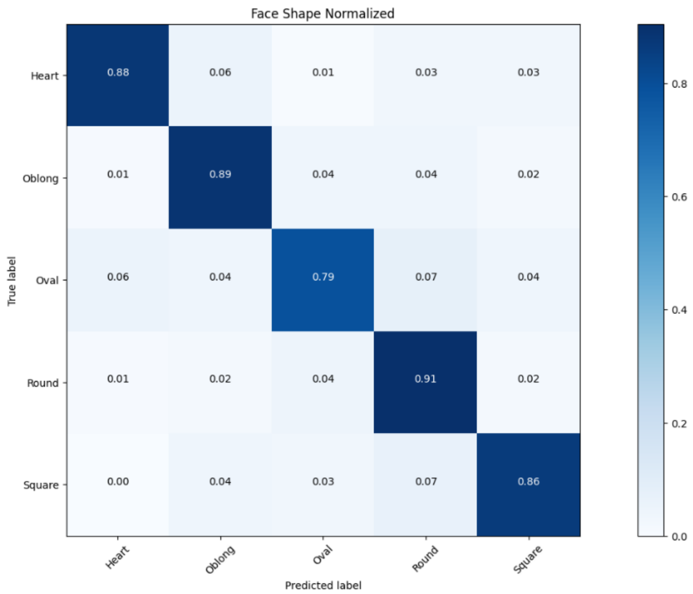


Figure 4.12 Confusion Matrix Experiment 4

According to the loss graph from figure 4.10, loss testing (yellow line) begins to rise at Epoch 30-40, meaning that the model becomes overfitted once the number of epochs is excessive. From figure 4.11 the training accuracy (blue) keeps increasing in the accuracy graph to almost 1.0 (perfect) and the testing accuracy (yellow) levels off, increasing the gap between the two further. In accordance with the confusion matrix from figure 4.12, there is no major improvement over the 50-epoch setting, and the performance of Oval face shape returns to 0.79.

e) Experiment No. 5

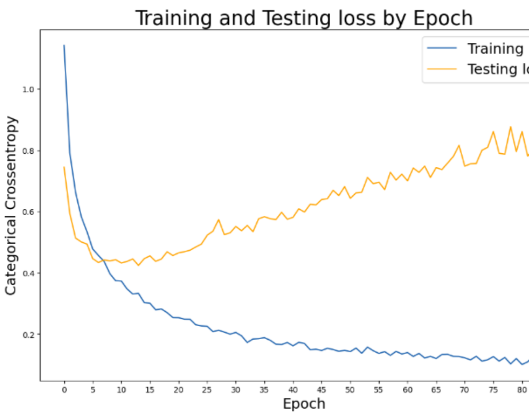


Figure 4.13 Training & Testing Loss Experiment 5

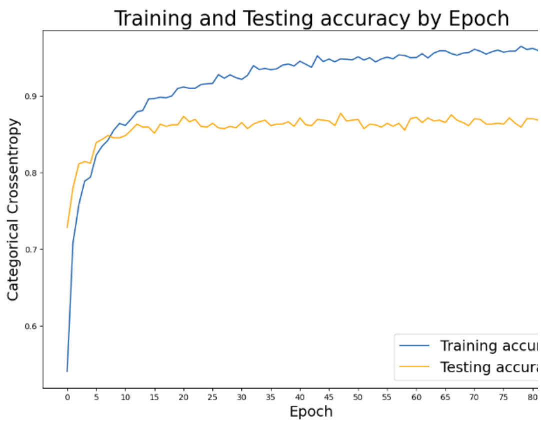


Figure 4.14 Training & Testing Accuracy Experiment 5

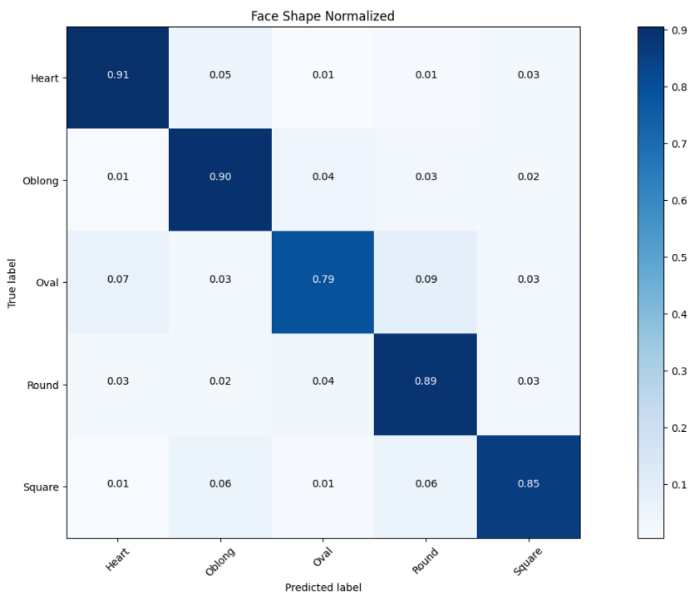


Figure 4.15 Confusion Matrix Experiment 5

From figure 4.13 and 4.14 there is more severe overfitting in the trend of the losses and accuracy graphs because despite the longer period of training (up to 85 epochs) the accuracy in testing does not increase and the loss in testing becomes even greater and is growing with time. Based on the confusion matrix from figure 4.15, the accuracy of Square face shape reduces slightly to 0.85 and the model has high confidence in identifying the Heart (0.91) and Oblong face shape (0.90).

f) Experiment No. 6

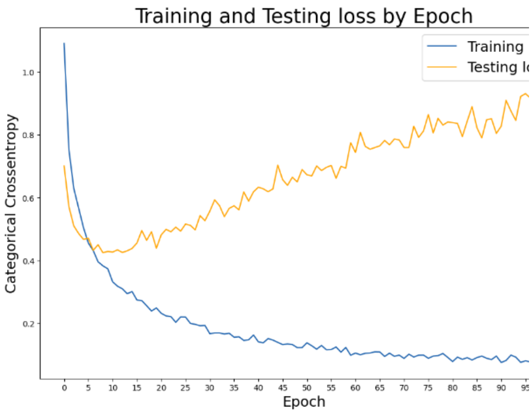


Figure 4.16 Training & Testing Loss Experiment 6

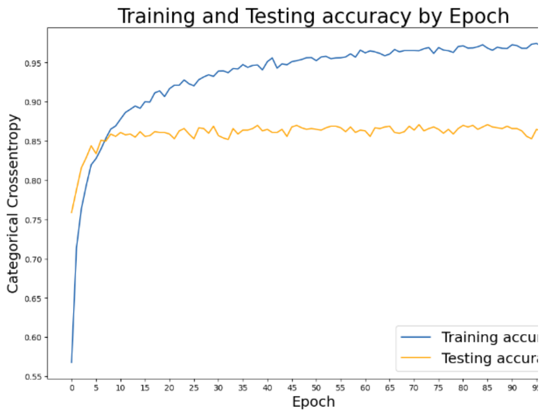


Figure 4.17 Training & Testing Accuracy Experiment 6

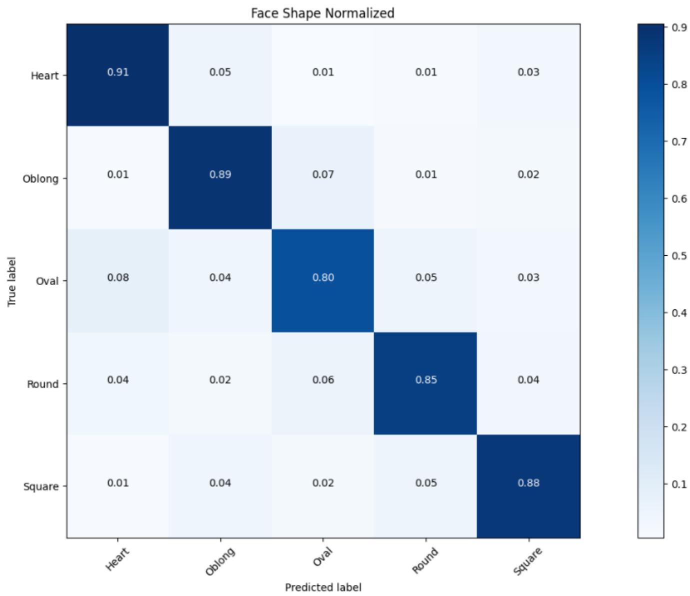


Figure 4.18 Confusion Matrix Experiment 6

The loss graph from figure 4.16 indicates a very wide difference between the training (blue) and testing (yellow) curves, which clearly demonstrates that training the model until it reaches the 100th epoch with this dataset is not efficient and it negatively affects the ability of the model to generalize. The training accuracy in the accuracy graph from figure 4.17 is about 0.98 and the testing accuracy is about 0.87 so it seemed that the model is memorizing the training data as opposed to learning patterns that will be applicable elsewhere. The confusion matrix from figure 4.18 shows that although there is a problem of overfitting, the accuracy of the classification of the Square face shape is improved a second time to 0.88, whereas Oval face shape is difficult to classify with 0.80.

g) Experiment No. 7

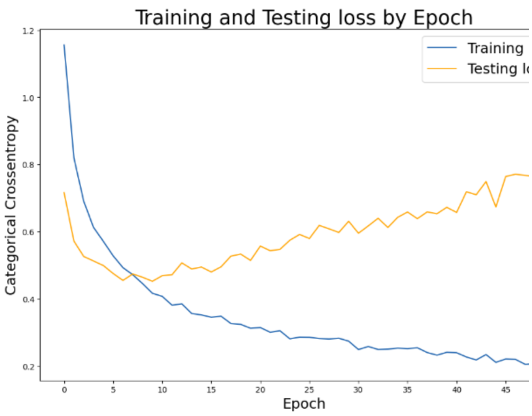


Figure 4.19 Training & Testing Loss Experiment 7

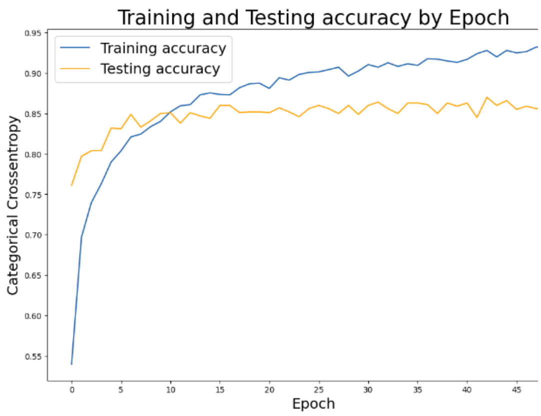


Figure 4.20 Training & Testing Accuracy Experiment 7

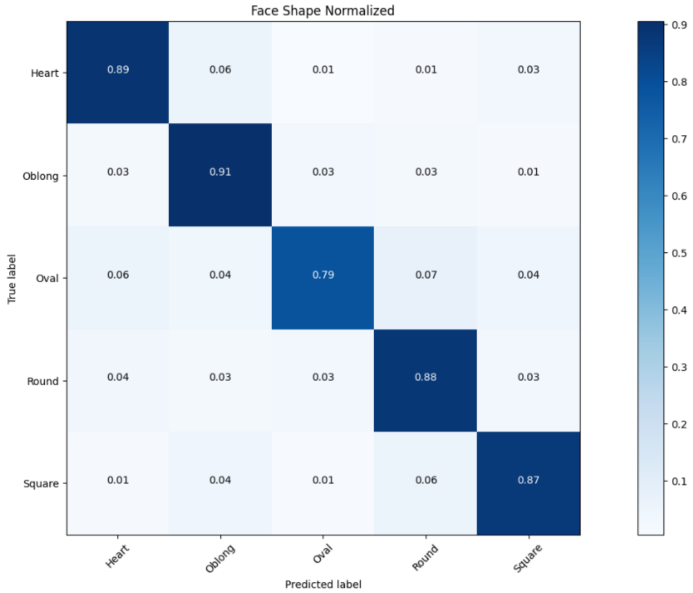


Figure 4.21 Confusion Matrix Experiment 7

The loss and accuracy plotting from figure 4.19 and 4.20 seem highly jagged, as it should be with a small batch size of 16, which results in unstable and noisy weight updates per epoch. Although the training curves are unstable, the ultimate performance remains quite decent where it is indicated in the confusion matrix from figure 4.21 that the Oblong face shape is the most accurate with an accuracy of 0.91.

h) Experiment No. 8

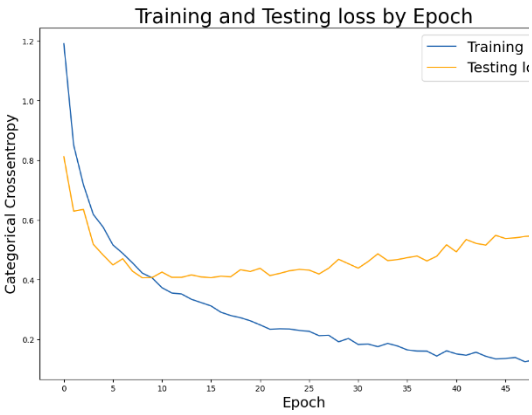


Figure 4.22 Training & Testing Loss Experiment 8

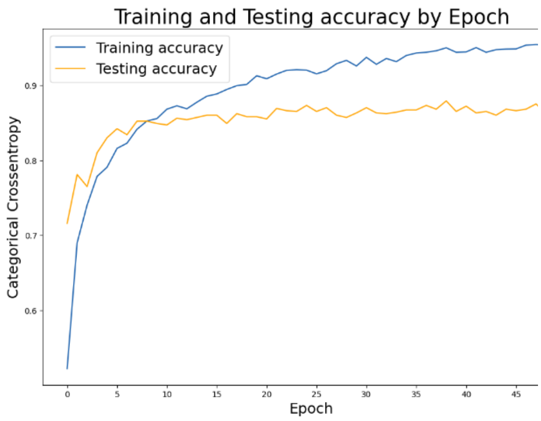


Figure 4.23 Training & Testing Accuracy Experiment 8

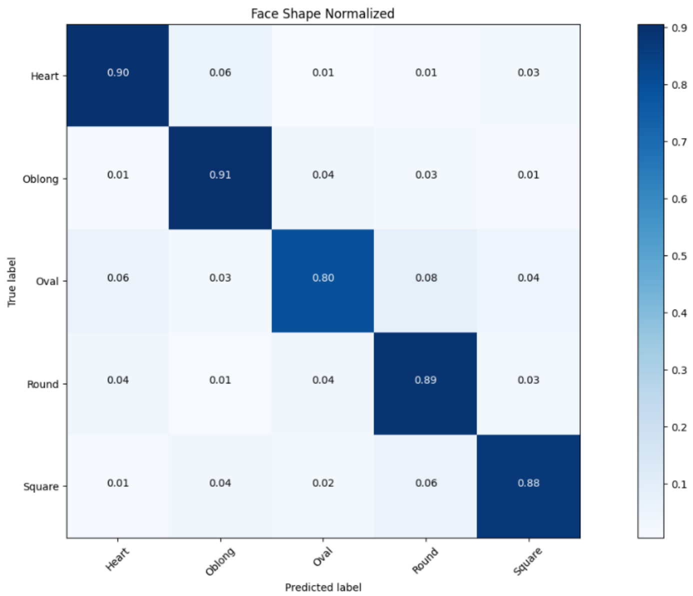


Figure 4.24 Confusion Matrix Experiment 8

Loss and accuracy curves from figure 4.22 and 4.23 become more smooth than Batch 16, which means that there is greater stability during training. The confusion matrix from figure 4.24 shows that the model has balanced performance in all classes and strong performance in detecting the Heart (0.90), Oblong (0.91), and Round (0.89) face shapes.

i) Experiment No. 9

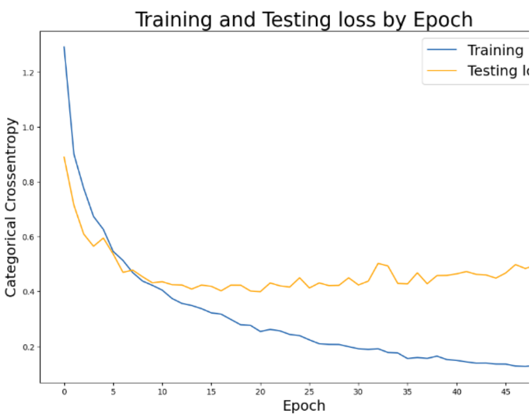


Figure 4.25 Training & Testing Loss Experiment 9

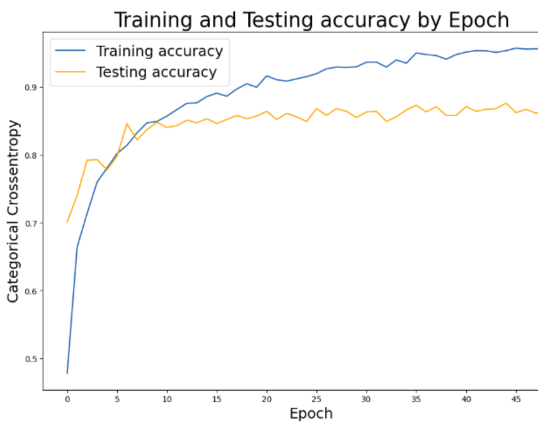


Figure 4.26 Training & Testing Accuracy Experiment 9

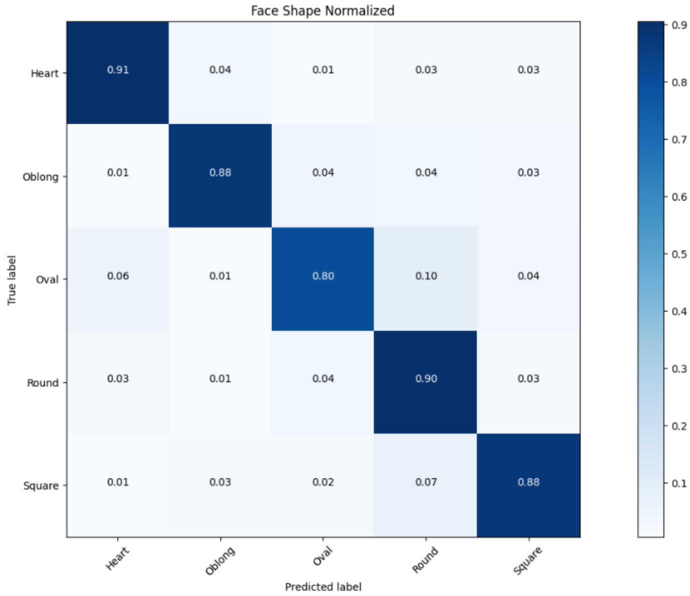


Figure 4.27 Confusion Matrix Experiment 9

Loss and accuracy curves from figure 4.25 and 4.26 are the most consistent of all the batch sizes; but larger batches may occasionally result in the model achieving its peak accuracy more gradually at the start of training (a phenomenon called generalization gap), but the training is otherwise regular. According to the confusion matrix from figure 4.27, the endpoint results are sound as the Heart (0.91) and Round (0.90) face shapes are mostly accurate, and the Oval face shape (0.80) remains predominantly a difficult case to be classified as either Round (0.10)

Table 4.2 Metric Table Presentation

Experiment No.	Parameter Experiment			Accuracy	Precision	Recall	F1-Score	Confusion Matrix [Heart, Oblong, Oval, Round, Square]
	Optimizer	Epoch	Batch Size					
1	Adam	50	32	0.868	0.865	0.868	0.865	[0.89, 0.90, 0.79, 0.90, 0.86]
2	SGD	50	32	0.854	0.858	0.854	0.855	[0.91, 0.90, 0.79, 0.85, 0.82]
3	RMSprop	50	32	0.868	0.866	0.868	0.866	[0.91, 0.88, 0.81, 0.89, 0.85]
4	RMSprop	70	32	0.866	0.866	0.866	0.865	[0.88, 0.89, 0.79, 0.91, 0.86]
5	RMSprop	85	32	0.868	0.866	0.868	0.866	[0.91, 0.90, 0.79, 0.89, 0.85]
6	RMSprop	100	32	0.866	0.862	0.866	0.864	[0.91, 0.89, 0.80, 0.85, 0.88]
7	RMSprop	50	16	0.868	0.868	0.868	0.867	[0.89, 0.91, 0.79, 0.88, 0.87]
8	RMSprop	50	64	0.876	0.870	0.876	0.872	[0.90, 0.91, 0.80, 0.89, 0.88]
9	RMSprop	50	128	0.874	0.868	0.874	0.870	[0.91, 0.88, 0.80, 0.90, 0.88]

In table 4.2 it shows the experiment result for green row which is the baseline model, yellow for optimizer experiment , orange for epoch experiment and blue for batch size expriment. Overall, experiment 8 has the best result.

a. Baseline Model Parameter Analysis

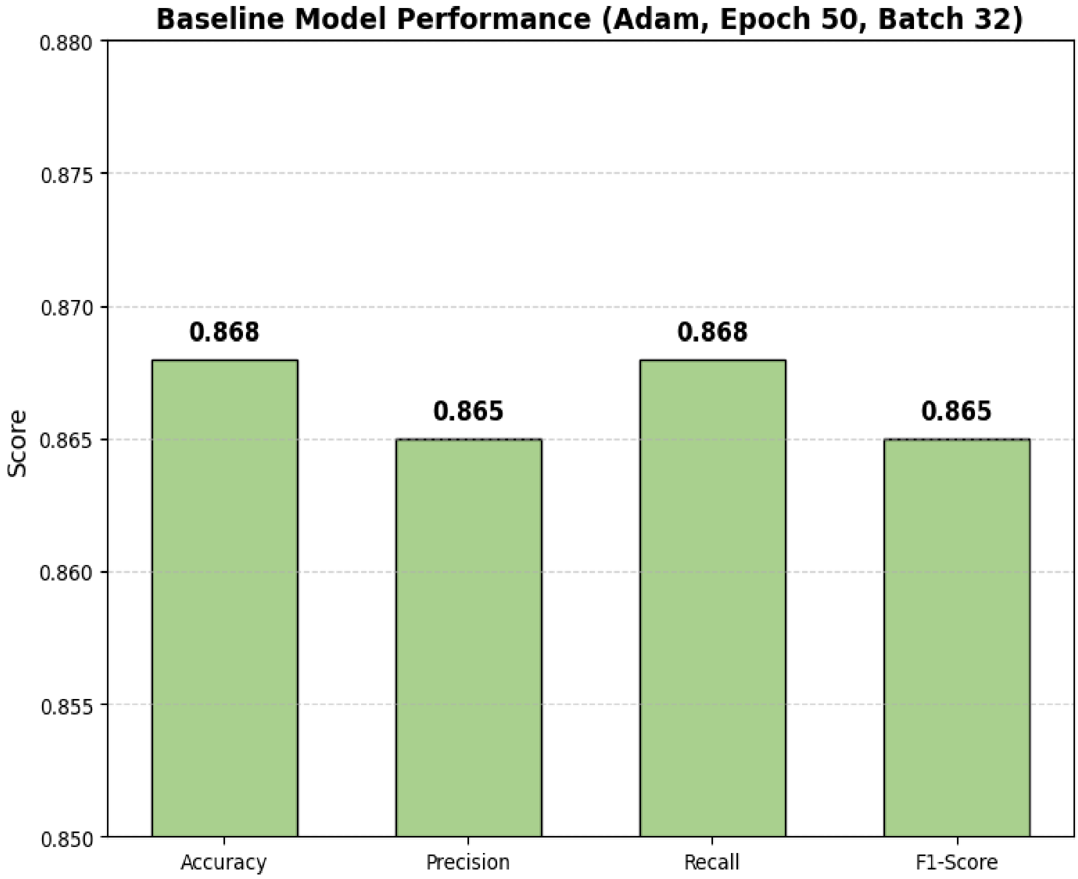


Figure 4.28 Accuracy, Precision, Recall, F1-Score Baseline Model

The baseline experiment was conducted by applying Adam as optimizer, with 50 epochs a batch size of 32 to make as a reference performance for the face shape classification model. The evaluation was performed using standard classification metrics consisting of accuracy, precision, recall, F1- score and confusion matrix. Overall, from figure 4.28 the model achieved an accuracy of 86.8% that showing the baseline configuration was able to correctly classify the majority of face images. The precision and recall values both recorded at 86.5% and 86% respectively mean for a balanced performance between correctly predicted classes and the model ability to identify face shapes. This balance is further supported by an F1-score of 86.5% indicating consistent classification performance.

This model confusion matrix reveals that the model performing best is recognizing Oblong (0.90) and Round (0.90) face shapes. It was followed closely by

Heart (0.89) and Square (0.86). These results suggest that the facial features associated with these shapes are well captured by the model. On the other hand, the oval face shape achieved the lowest score (0.79) and indicated a higher degree of misclassification. In summary, the baseline experiment parameter provides a strong foundation with stable and balanced performance across most face shape categories.

b. Optimizer Analysis

Figure 4.29 Optimizer Effect

The purpose of this experiment was to determine how the different optimizers would affect the performance of the face shape classification model. Three optimizers were compared, which are Adam from baseline model, SGD and RMSprop with other training parameters like the number of epochs 50 and batch size 32 being held constant to make a fair comparison. According to the experimental findings and figure 4.29, both Adam (Experiment 1) and RMSprop (Experiment 3) had the same total accuracy of 86.8% meaning that both optimizers are efficient in optimization of the learning process. In comparison, SGD (Experiment 2) achieved a smaller accuracy of 85.4 percent, and it took longer to converge and learn, implying it was not as stable as adaptive optimizers.

RMSprop showed a bit better on precision, recall and F1-score and had 0.866, 0.868 and 0.866 respectively, which are slightly better than Adam. This implies that RMSprop is more balanced when it comes to a proper balance of identifying face shape classes in the right way and reducing classification errors. The analysis based on the confusion matrix by classification indicates further that RMSprop scores better on classification of the oval face shape as it scores 0.81 on its classification, relative to the scoring 0.79 by Adam and SGD. This is a significant improvement because the oval class can be easily confused with other types of face, specifically Round and Oblong, which makes it the more difficult type to classify. Further, RMSprop is stable in other classes, such as Heart (0.91), Round (0.89), and Square (0.85) and this implies good generalization power.

In conclusion, even though Adam and RMSprop had the same level of accuracy, RMSprop was chosen as the final optimizer because it is more stable in terms of evaluation metrics and it tends to minimize misclassification in more difficult classes, in particular the oval face shape. RMSprop was therefore considered to be the best optimizer that would be used in future experiments to improve the robustness and reliability of the model.

c. Epoch Analysis

Figure 4.30 Epoch Effect

Experiments 4, 5 and 6 were carried out in order to test the effects of increment in the number of training epochs on the performance of the face shape classification model. The epochs varied which are 70,85 and 100 while other parameters being held constant. Figure 4.30 shows the findings suggest that the error reduction in terms of the number of epochs is not significant. The values of accuracy were within a low range at 86.6 to 86.8 which indicates that the model had acquired adequate discriminative aspects at a time of training which was earlier.

Though the performance indicators, including precision, recall, and F1-score, were not significantly changed, F1-score at 100 epochs (0.864) decreased a bit and the first indicators of overfitting appeared. Although some classes increased marginally at the higher epochs, this was compensated by a relative decline in other classes, and no significant net improvement was achieved. Such an imbalance would indicate that, over time, there is no always-improvement of class-wise generalization with greater training.

This observation is further supported by the training and testing loss graphs on which the testing loss rose after the number of epochs surpassed 50. This shows lesser performance of generalization even after extended training. Thus, the final epoch configuration was chosen to be 50 epochs of the baseline model, which offers a decent trade-off between fixed performance, good generalization and computational efficiency.

d. Batch Size Analysis

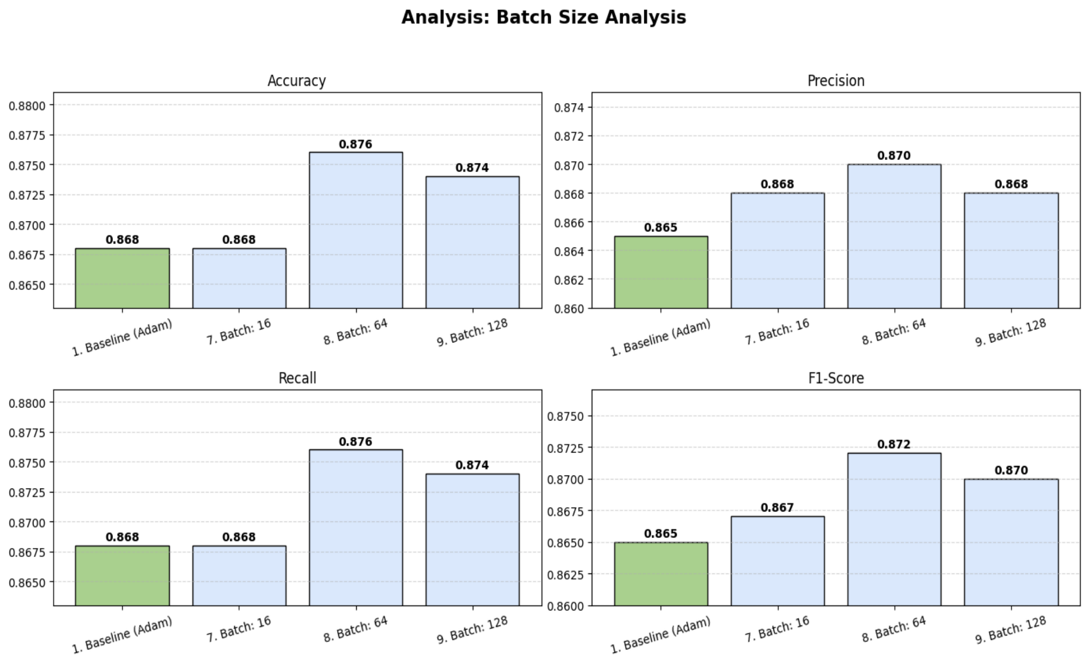


Figure 4.31 Batch Size Effect

The Experiment 7-9 was performed to examine how the face shape classification model is affected by the batch size. The batch sizes of 16, 64 and 128 were tested with the constant optimizer and epochs. The findings from figure 4.31 indicate that the batch size has a significant influence on the performance of the model and model training stability.

Experiment 8 with batch size of 64 got the highest overall performance with the highest accuracy of 87.6 and F1-score of 0.872. The batch size of 64 offers a more favourable trade-off between gradient stability and generalization capacity than the batch size of 16 and 128. This observation can be further supported by class-wise analysis wherein the batch size of 64 obtained high performance in all the face shape categories with Square (0.88) and Oval (0.80) having high performance and they are usually more difficult to classify.

Further, based on analysis of the training and testing loss curves, the larger the batch size the more stable the testing loss over the epochs is particularly at batch sizes of 64 and 128. Nevertheless, the behaviour of batch size 128 with respect to training can still be described as stable, but it does not outperform the total results of the batch size 64. Thus, a batch size of 64 was chosen because it provides the best performance of classification and generalization remains steady and consistent.

4.2 Face Shape Prediction and Analysis

In this section, the proposed model is able to predict the face shape on real word images randomly obtained from the internet, which did not belong to the training or testing sets. This was evaluated by the most successful model configuration as determined in Experiment 8 and this evaluation ensures that the predictions can be considered as the generalization of the model on some unseen data.

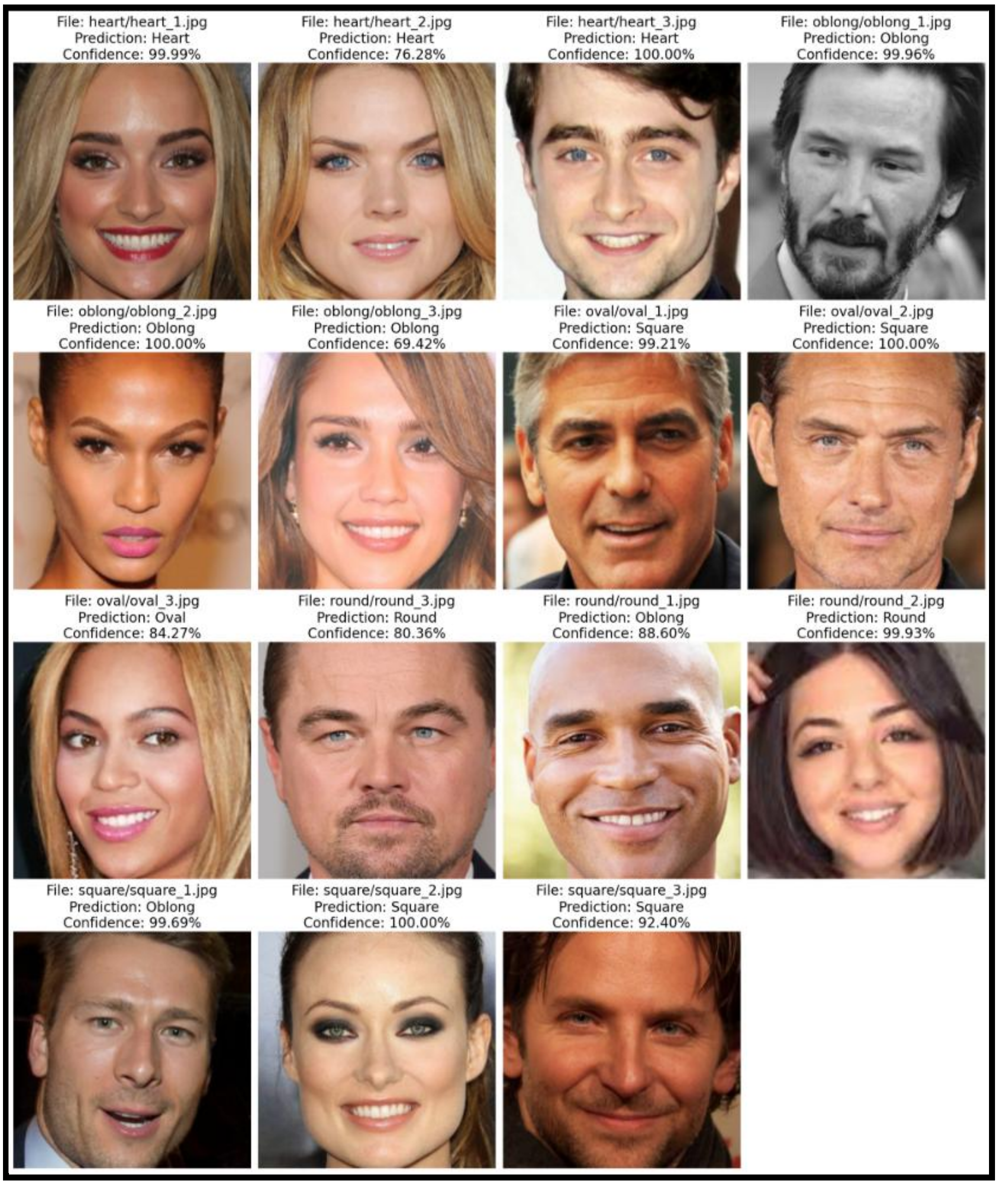


Figure 4.32 Random Image Test

The results of the prediction indicate that the model is very confident when dealing with images that have explicit and distinct faces. An example given below like heart3.jpg and square2.jpg were identified with 100 percent accuracy meaning that the model is very dependable in the instance that the facial structure, jawline and face proportions are clearly uncovered. These findings indicate that the acquired facial representations are adequate in representing discriminatory geometric features of various face shapes.

Furthermore, some cases of misclassification were also noticed. One of them is oval_1.jpg that was wrongly classified as Square with a confidence of 99.21. Other external factors that could account for this mistake may be the camera angle, position of the face, lighting condition or shadows around the jawline which could change the perceived geometry on the face. Also, some face shapes, especially Oval, Square, and Oblong, may have some overlapping features that may add more ambiguity to the classification.

Nevertheless, the model exhibits the ability to generalize well even in unconstrained and extensive samples. Given that the general accuracy is around 86 and it is reasonable to expect the occasional misclassification of real-world pictures. The difference in the background, hairstyle, facial expression and quality of the image did not have much effect on the performance. This shows that the fine-tuning of the VGG16 architecture was indeed effective at enhancing model robustness. All in all, the suggested system can be considered reliable enough to be applied to the real-world face shape prediction.

4.3 Rule-Based Eyewear Recommendation Results

This part provides the findings of the rule-based eyewear recommendation system that uses aesthetic rules that are pre-determined to suggest appropriate eyewear frames that best match and balance the face shape chosen by the user. This is in contrast to the face shape prediction module where the face shape is automatically selected by the user where the prediction page essentially remains an autonomous part of the system. The user is free to select the face shape they want to be recommended to which is a separate part since the user is in charge of the process.

Regarding the user interface, the system will provide some categories of face shapes that one can pick at the page, which are Heart, Oblong, Oval, Round and Square. Users have to manually decide the face shape which best reflects their faces or they can use the result they got from the face shape prediction earlier. After a choice has been made, the system restrictively and dynamically filters and shows the appropriate eyewear frames in the left sidebar based on the chosen category. The design is very simplistic, clear and easy to use especially to users who would rather choose the sample manually than through automated detection.

The logic of the rule-based mapping is applied on the basis of the guidelines of the facial aesthetics. As an example, angular frames like Rectangle or Wayfarer are suggested to people with a round face shape to create contrast and define faces better. Conversely, customers who have a square face shape are recommended Round or Oval frames to flatten off their teeth and create a visual balance. The system allows a uniform and understandable set of recommendations through the application of explicit and explainable rules, so it is a feasible element of assisting eyewear choice before virtual try-on.

Figure 4.33 Result Rule-Based

4.4 WebAR Try-On Results

In this section, the findings of the WebAR virtual try-on system that was created on the basis of the Mind.AR library and technology allowing tracking the face in real-time and supporting the visualization of an augmented reality directly in a web browser will be provided. With WebAR, no extra mobile apps would be required and the try-on feature would be accessible to the users conveniently across the devices.

The system allows having a large range of eyewear models of different 3D quality to choose among which they are obtained from the free license source. These models have realistic geometry and proportions, allowing them to be properly visible of how the various frames fit on the face of the user. The system uses licensed 3D assets because it provides visual appeal with results without handbased 3D modeling, hence simplifying the development side and minimizing time.

The combination of Mind.AR enables 3D eyewear models to be real-time overlapped on the face as shown in figure 4.34 of the user in terms of AR performance. The virtual glasses are also kept at a stable position along the eye region irrespective of the movement of the head of the user or change of face orientation. This shows that the facial landmark tracking and stable pose estimation is effective and is vital to an immersive AR experience.

Moreover, the system allows real-time communication and reactivity with adjustments in head position, facial expression and viewing angle being displayed instantly in the AR with very low latency. This reaction makes users more engaged and realistic, as they can even intuitively interact with the virtual glasses as real glasses. In sum, the WebAR try-on module is an interactive, smooth and realistic virtual fitting process that supplements the parts of the system.

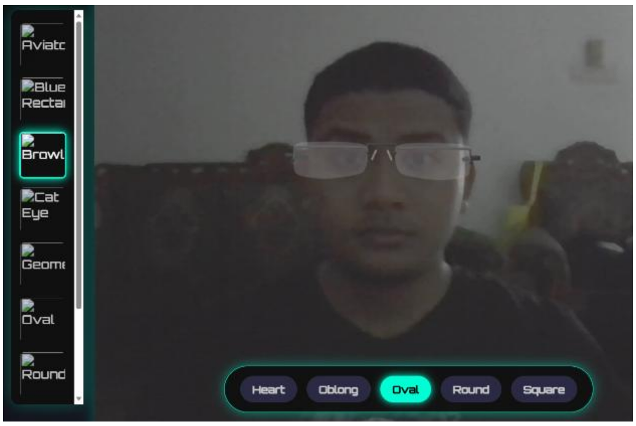


Figure 4.34 Result Rule-Based on AR

4.5 System Integration Results

This section is presenting the results of the systems integration that combines all major modules of the proposed system into a complete end to end workflow. It starts from user interaction and face shape analysis to eyewear recommendation and WebAR visualization. The integration ensures smooth communication between the frontend interface, backend processing and augmented reality components.

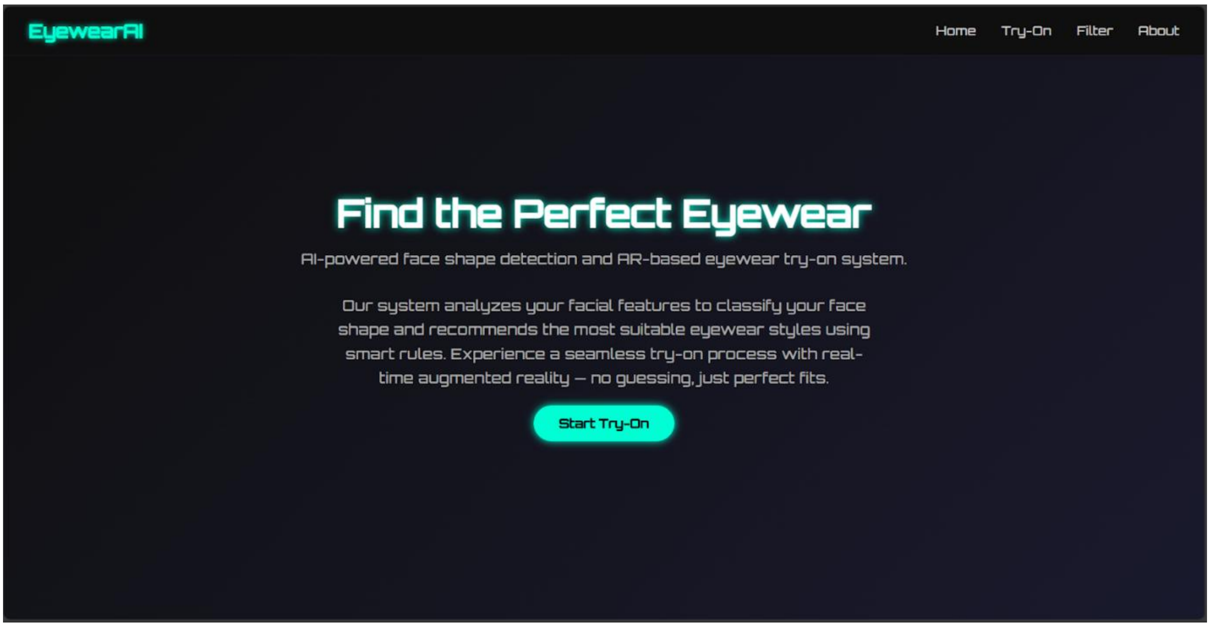


Figure 4.35 Prototype Index Interface

The system interaction starts at the landing page that illustrated in figure 4.35 which serves as an introduction interface providing a brief overview of the application and its purpose. It consists of navigation at the top where user can click to go to the try-on, filter, about page. There is also a button in the middle where it will directly bring the user to the try-on page.

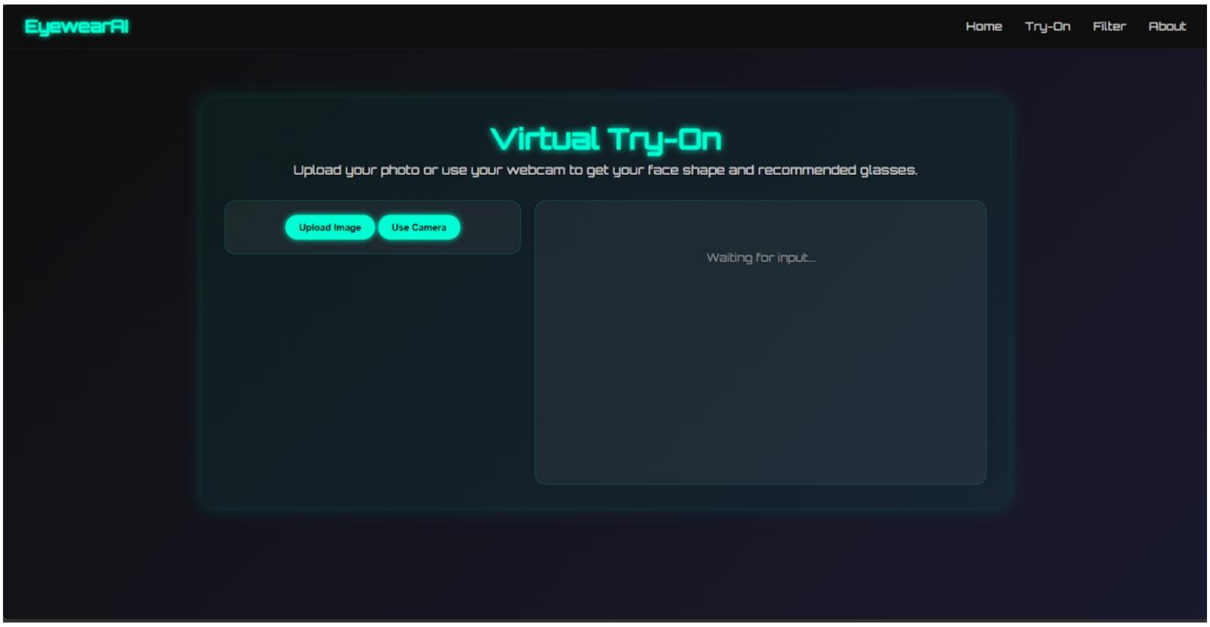


Figure 4.36 Try-On Interface

Users then can proceed to the Try-On page that illustrated in figure 4.36 where they are given the option to either upload an image or use a webcam for the face shape analysis. Once an image is captured or uploaded, the system performs face detection and classification in the backend

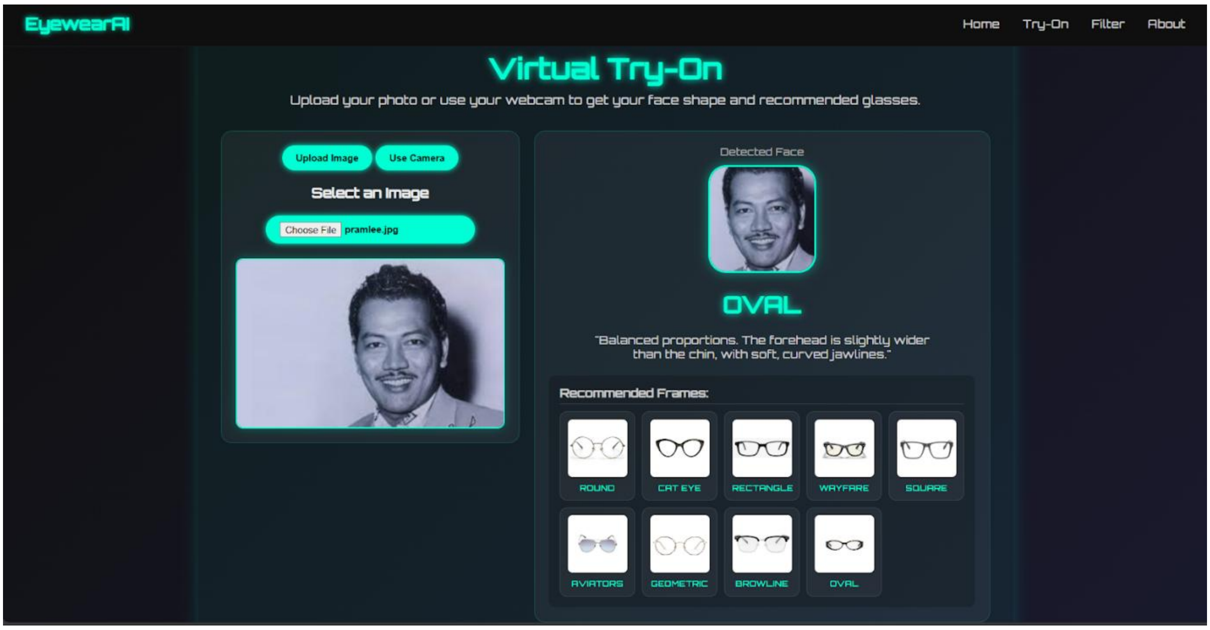


Figure 4.37 Result Interface When Face Exist

If a face is successfully detected that illustrated in figure 4.37, the system will display the detected face region along with the predicted face shape and a list of relevant facial specifications. This confirms that the classification module is functioning purposely and provides users with immediate feedback.

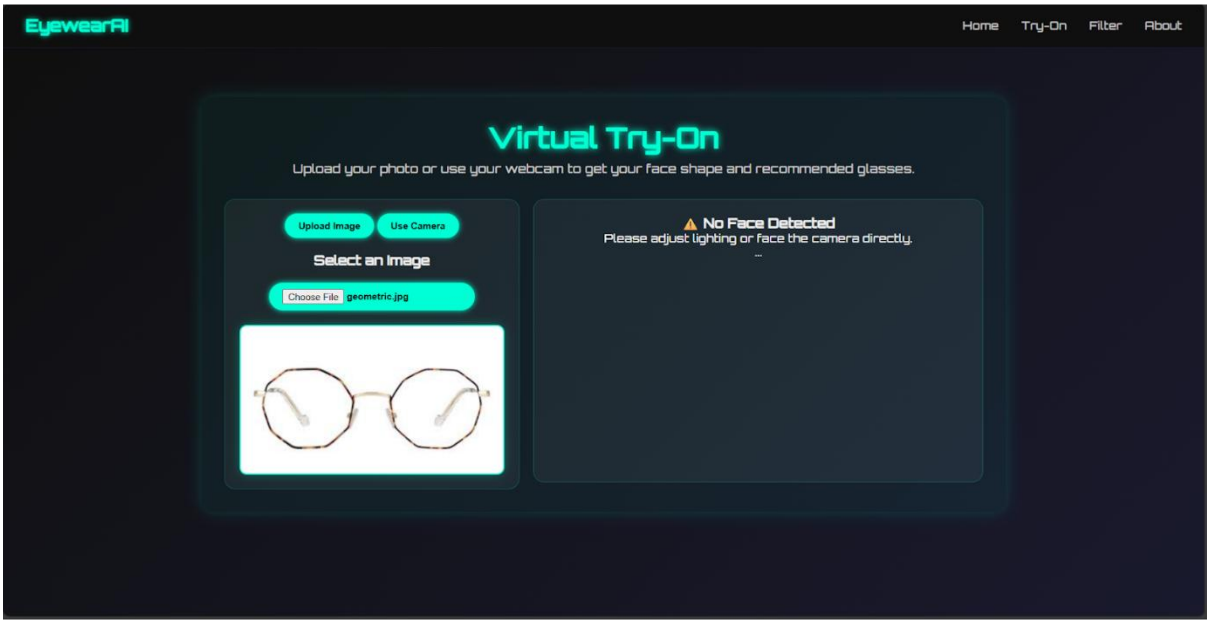


Figure 4.38 Result Interface When Face Non-Exist

In contrast, if there is no face detection that illustrated in figure 4.38 an error message is displayed informing the user that no face was found. There also will be instructions to upload a clearer image or adjust camera position and lighting conditions. This validation mechanism improves usability and prevents incorrect predictions

Figure 4.39 AR Interface

After obtaining the face shape result, users may proceed to the filter page that illustrated in figure 4.39 where they can explore suitable eyewear options and activate the WebAR try-on module. This allows users to visualize selected eyewear frames on their face in real time.

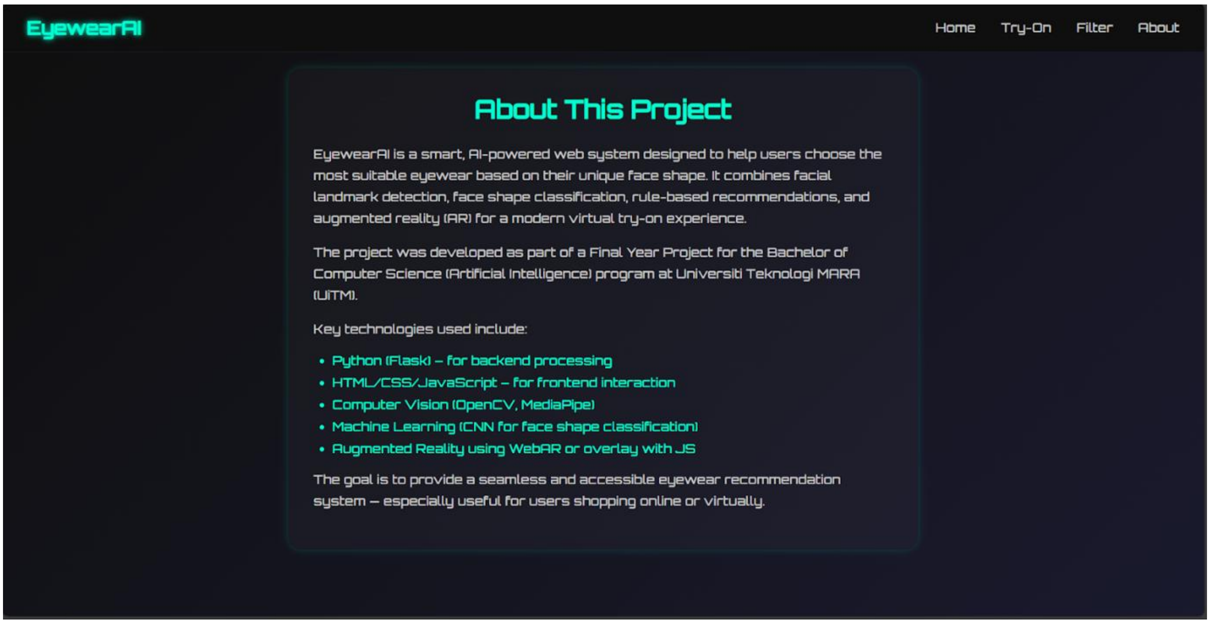


Figure 4.40 Prototype Description

Finally, the last page is about page that illustrated in figure 4.40 that provides a description of information about the project that includes system objectives, features, underlying technologies and serves as a reference section for users. Overall, the successful integration of these interfaces demonstrates the effectiveness of the proposed system in delivering a seamless interactive user experience. The structured flow from image input to AR visualization confirms that the system operates reliably as a unified smart eyewear recommendation system.

4.6 Summary

In this chapter, it shows the evaluation of the proposed intelligent virtual eyewear try-on system. The CNN model for identifying facial shapes was VGG16. The tuned model showed enhanced accuracy and generalization performance following hyperparameter adjustment. The confusion matrix reveals that as with some misclassification among related classes. The system precisely identified the majority of face shapes including oval, round, square, oblong and heart. In order to provide tailored and contextually relevant recommendations the face shape was assigned with a rule-based mapping to suitable eyewear frames. Even with a slight head movement the WebAR try-on was able to overlay virtual glasses in real-time while maintaining accurate nodes and responsiveness. At the end, system integration was completed through feed of an image with the face shape classification, eyewear recommendation and augmented reality visualization. This also demonstrated the system's full functionality. This met the project requirements and objectives as it also provides a practical and engaging user experience.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

This chapter concludes up the study by reviewing the results, constraints and possible future enhancements of the system that was developed. It indicates possible ways for future improvement, discusses the level to which the study objectives have been met and identifies implementation related difficulties. This chapter considers the system's overall performance and contribution to achieving its objectives.

5.1 Research Objective Achievement

This project aimed to develop an intelligent eyewear recommendation system that integrates face shape classification, rule-based recommendation and WebAR visualization. The achievement in this project objective is discussed as follows.

5.1.1 Face Shape Classification Using VGG16

The initial objective was to create a machine learning model, which can classify face shape based on face images through a Convolutional Neural Network (CNN). This was attained successfully. A CNN classification was developed using VGG16 architecture which was fine-tuned on a dataset consisting of five shape categories which are Heart, Oblong, Oval, Round and Square.

Several experiments were performed to optimize the model performance including optimizer selection, epoch variation and batch size tuning. The final model configuration that has been selected is RMSprop as optimizer, 50 of epochs and batch size of 64. These are the most satisfactory parameters that have been chosen such that the overall performance has been the best with the accuracy of about 86 percent and consistent values of recall and F1 scores. There was also good generalization in the model when the real-world images that were not included in the training or testing datasets were tested.

Overall, the CNN model was able to consistently identify face shapes with acceptable accuracy and robustness and this is fulfilling the first research objective.

5.1.2 Rule-Based Eyewear Recommendation System

The second objective was to come up with a recommendation system that would be used to match face shapes with appropriate eyewear frame styles. This was successfully attained by the use of rule-based recommendation methodology in which already established aesthetic rules were implemented using facial geometry principles.

The rule-based interface is implemented in both pages of Try-On and Filter interfaces to support flexible user interaction. After face shape detection on the Try-on page, the lists of suitable eyewear specifications are automatically displayed while users may also directly access the Filter page and manually select a face shape if desired. In both cases, consistent rule-based logic is applied.

This approach ensures the recommendation consistency, transparency and ease of use that turn it to successfully achieving the second research objective.

5.1.3 WebAR Virtual Try-On Implementation

The third objective was to implement an augmented reality module that allows users to virtually try on recommended eyewear in real time. This objective was successfully accomplished through the integration of WebAR technology using MindAR that helps enable facial tracking and 3D eyewear overlay directly within a web browser.

The WebAR module is effective in tracking facial landmarks and ensures that the 3D eyewear models align well even when the user moves the head. The system facilitates real-time communication and responsiveness in real-time with a low latency and smooth rendering. Free licensed and decent models of 3D are also used without overcomplicating the development. This successful integration of an AR visualization confirms the achievement of the third research objective.

5.2 Project Limitation

Despite getting all project objectives, several limitations were identified. One major limitation is dataset diversity and class ambiguity. The training dataset consisted of approximately 5,000 images that were equally distributed into five face shape categories. While it is sufficient for model training, the dataset lacks diversity in terms of gender, facial hair, facial expressions and face accessories.

The other constraint is associated with the WebAR performance and environment-dependency. These differences are caused by variations in highlighting conditions, camera quality and distance between the user and the camera that may influence face tracking accuracy and 3D eyewear alignment. Moreover, the working of the AR module in a browser setup can be slow based on the ability of the device and network availability. It might possibly lead to slight delays or frame jitter.

Lastly, the rule-based recommendation system relies on predefined mappings and does not account for personal preferences, fashion trends or individual facial proportions beyond face shape. This limits personalization and adaptability in eyewear suggestion.

5.3 Future Work

Things that can be improved in the future are generalization of the models and inclusive process using more diverse datasets that include more categories of face shapes and equal gender representation. More complex methods like synthetic data creation and bigger public ones can also enhance robustness.

In the context of AR visualization, further research is possible on depth estimation or 3D facial mesh reconstruction to enhance realism in head rotation and movement. Latency could also be minimized by optimizing WebAR performance with model compression. Moreover, including standardized eyewear profiles including the frame width, lens height, bridge length and temple length may also increase the precision in fitting and the visual consistency.

In terms of recommendation, a hybrid or learning based recommendation system could be introduced by integrating user preference, feedback and style trends. This would allow the system to provide more personalized and adaptive eyewear recommendations beyond static rule-based logic.

5.4 Summary

In conclusion, this project successfully developed a face shape-based eyewear recommendation system that integrates CNN based face shape classification, rule-based recommendation and WebAR virtual try-on. The system achieved a good classification accuracy of approximately 86% and it demonstrated well generalization when evaluated on real-world images. It also provided an interactive and responsive augmented reality experience. Although limitations remain in terms of recommendation flexibility, the system proposed establishes a good foundation for intelligent virtual eyewear applications. This project has strong potential for real-world deployment and future commercial use through further improvements in robustness, personalization and visualization realism.

REFERENCES

Algaraawi, N., Morris, T., & Cootes, T. F. (2024). Facial feature point detection under large range of face deformations. *Journal of Visual Communication and Image Representation*, *103*. https://doi.org/10.1016/j.jvcir.2024.104264

Alkady, Y., Rizk, R., Alsekait, D. M., Alluhaidan, A. S., & Abdelminaam, D. S. (2024). SINS_AR: An Efficient Smart Indoor Navigation System Based on Augmented Reality. *IEEE Access*, *12*, 109171–109183. https://doi.org/10.1109/ACCESS.2024.3439357

An, S., Che, G., Guo, J., Zhu, H., Ye, J., Zhou, F., Zhu, Z., Wei, D., Liu, A., & Zhang, W. (2021). ARShoe: Real-Time Augmented Reality Shoe Try-on System on Smartphones. *MM 2021 - Proceedings of the 29th ACM International Conference on Multimedia*, 1111–1119. https://doi.org/10.1145/3474085.3481537

Bai, X., Huerta, O., Unver, E., Allen, J., & Clayton, J. E. (2021). A parametric product design framework for the development of mass customized head/face (Eyewear) products. *Applied Sciences (Switzerland)*, *11*(12). https://doi.org/10.3390/app11125382

Blattgerste, J., Behrends, J., & Pfeiffer, T. (2023). TrainAR: An Open-Source Visual Scripting-Based Authoring Tool for Procedural Mobile Augmented Reality Trainings. *Information (Switzerland)*, *14*(4). https://doi.org/10.3390/info14040219

Borisova, T., Stoykova, V., Kazlacheva, Z., & Videnov, K. (2021). Developing an augmented reality textbook for Bachelor and Master programmes "design, technology and management of the fashion industry. *IOP Conference Series: Materials Science and Engineering*, *1031*(1). https://doi.org/10.1088/1757-899X/1031/1/012123

Cao, J., Lam, K.-Y., Lee, L.-H., Liu, X., Hui, P., & Su, X. (2021). *Mobile Augmented Reality: User Interfaces, Frameworks, and Intelligence*. https://doi.org/10.1145/3557999

Chen, B., Macdonald, S., Attallah, M., Chapman, P., & Ghannam, R. (2025). *A Review of Prototyping in XR: Linking Extended Reality to Digital Fabrication*. http://arxiv.org/abs/2504.02998

Cheng, S., Ma, C., & Pan, Y. (2024). StylizedFacePoint: Facial Landmark Detection for Stylized Characters. *MM 2024 - Proceedings of the 32nd ACM International Conference on Multimedia*, 8072–8080. https://doi.org/10.1145/3664647.3680984

De Luna, R. G., Mendoza, D. M., Isada, M. R., Navarro, C. F. C., Tubola, O. D., MacAdag-Um, J. M. P., & Tuason, L. (2024). VIScial: Visual Classification of Facial Shape Using Deep Transfer Learning. *IST 2024 - IEEE International Conference on Imaging Systems and Techniques, Proceedings*. https://doi.org/10.1109/IST63414.2024.10759179

Ferrão, J., Dias, P., Santos, B. S., & Oliveira, M. (2023). Environment-Aware Rendering and Interaction in Web-Based Augmented Reality. *Journal of Imaging*, *9*(3). https://doi.org/10.3390/jimaging9030063

Grd, P., Tomičić, I., & Barčić, E. (2024). Transfer Learning with EfficientNetV2S for Automatic Face Shape Classification. *Journal of Universal Computer Science*, *30*(2), 153–178. https://doi.org/10.3897/jucs.104490

Gupta, Y. P., Mukul, & Gupta, N. (2023). Deep learning model-based multimedia retrieval and its optimization in augmented reality applications. *Multimedia Tools and Applications*, *82*(6), 8447–8466. https://doi.org/10.1007/s11042-022-13555-y

Hořejši, P., Machac, T., & Šimon, M. (2024). Reliability and Accuracy of Indoor Warehouse Navigation Using Augmented Reality. *IEEE Access*, *12*, 94506–94519. https://doi.org/10.1109/ACCESS.2024.3420732

Hossam, M., Afify, A. A., Rady, M., Nabil, M., Moussa, K., Yousri, R., & Darweesh, M. S. (2021, July 3). A comparative study of different face shape classification techniques. *ICEEM 2021 - 2nd IEEE International Conference on Electronic Engineering*. https://doi.org/10.1109/ICEEM52022.2021.9480638

Hu, J., Wu, W., Ding, M., Huang, X., Deng, Z. J., & Li, X. (2023). A virtual try-on system based on deep learning. *2023 3rd International Symposium on Computer Technology and Information Science, ISCTIS 2023*, 103–107. https://doi.org/10.1109/ISCTIS58954.2023.10213129

Huang, B., Li, Q., Li, Y., Chen, X., Liang, G., Ke, M., Xie, D., Huang, G., Zhong, Q., & Chen, H. (2024). Research of Facial Landmark Detection Algorithm based on Deep Learning. *ACM International Conference Proceeding Series*, 561–569. https://doi.org/10.1145/3671151.3671251

Ishikawa, S., & Ikenaga, T. (2022). Image-based virtual try-on system with clothing extraction module that adapts to any posture. *Computers and Graphics (Pergamon)*, *106*, 161–173. https://doi.org/10.1016/j.cag.2022.06.007

Joshi, T. P., Yadav, A. K., Chhetri, A., Agrahari, S., & Ghimire, U. K. (2024). *Virtual Trial Room with Computer Vision and Machine Learning*. http://arxiv.org/abs/2412.10710

Kakoutopoulos, K., Drakakis, E., Papadopoulou, A., & Goumopoulos, C. (2025). Feasibility of Augmented Reality-Based Cognitive Training for Older Adults: The MarketMind AR Approach. *Sensors*, *25*(7). https://doi.org/10.3390/s25072081

Kamalam, G. K., Joshi, S., Maheshwari, M., Selvan, K. S., Jamal, S. S., Vairaprakash, S., & Alhassan, M. (2022). Augmented Reality-Centered Position Navigation for Wearable Devices with Machine Learning Techniques. *Journal of Healthcare Engineering*, *2022*. https://doi.org/10.1155/2022/1083978

Li, J., Liu, Y., Xiao, L., Cai, J., He, H., & Zhang, X. (2024). Augmented Reality Creation Platform Based on Graphical Programming. *IEEE Access*, *12*, 14455–14465. https://doi.org/10.1109/ACCESS.2024.3357549

Li, X., Wu, K., & Zhang, S. (2024). Landmark-in-facial-component: Towards occlusion-robust facial landmark localization. *Image and Vision Computing*, *151*. https://doi.org/10.1016/j.imavis.2024.105289

Li, Y., Wang, C., Xie, L., Jin, Q., Fan, L., Ning, J., & Lu, S. (2024). Facial Landmark Detection Based on High Precision Spatial Sampling via Millimeter-wave Radar. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *8*(4). https://doi.org/10.1145/3699739

Loukil, M., Ghozzi, Y., & Ktari, R. (2023). Face Shape Classification based on Deep Face. *2023 1st IEEE Afro-Mediterranean Conference on Artificial Intelligence, AMCAI 2023 - Proceedings*. https://doi.org/10.1109/AMCAI59331.2023.10431496

Lu, F., Zhou, H., Guo, L., Chen, J., & Pei, L. (2021). An arcore-based augmented reality campus navigation system. *Applied Sciences (Switzerland)*, *11*(16). https://doi.org/10.3390/app11167515

Luidolt, L. R., & Zhou, K. (2023). Hands-Free Virtual Try-On using Untethered AR Glasses for Everyday Shopping Experiences. *Proceedings - 2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct, ISMAR-Adjunct 2023*, 507–512. https://doi.org/10.1109/ISMAR-Adjunct60411.2023.00110

Magaia, N., Gomes, P., Silva, L., Sousa, B., Mavromoustakis, C. X., & Mastorakis, G. (2021). Development of Mobile IoT Solutions: Approaches, Architectures, and Methodologies. *IEEE Internet of Things Journal*, *8*(22), 16452–16472. https://doi.org/10.1109/JIOT.2020.3046441

Marelli, D., Bianco, S., & Ciocca, G. (2022). Designing an AI-Based Virtual Try-On Web Application. *Sensors*, *22*(10). https://doi.org/10.3390/s22103832

Martin-Brualla, R., Pandey, R., Bouaziz, S., Brown, M., & Goldman, D. B. (2020). *GeLaTO: Generative Latent Textured Objects*. http://arxiv.org/abs/2008.04852

Minaee, S., Liang, X., & Yan, S. (2022). *Modern Augmented Reality: Applications, Trends, and Future Directions*. http://arxiv.org/abs/2202.09450

Nguyen, V. T., Jung, K., & Dang, T. (2020). BlocklyAR: A visual programming interface for creating augmented reality experiences. *Electronics (Switzerland)*, *9*(8), 1–20. https://doi.org/10.3390/electronics9081205

Nikolarakis, A., & Koutsabasis, P. (2024). Mobile AR Interaction Design Patterns for Storytelling in Cultural Heritage: A Systematic Review. In *Multimodal Technologies and Interaction* (Vol. 8, Issue 6). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/mti8060052

Rifat, R. H., Siddique, S., Das, L. R., & Haque, M. A. (2023). Facial Shape-Based Eyeglass Recommendation Using Convolutional Neural Networks. *2023 IEEE Symposium Series on Computational Intelligence, SSCI 2023*, 867–872. https://doi.org/10.1109/SSCI52147.2023.10371836

Russo, M. (2021). AR in the architecture domain: State of the art. In *Applied Sciences (Switzerland)* (Vol. 11, Issue 15). MDPI AG. https://doi.org/10.3390/app11156800

Sagarika, S., & Mokashi, B. (2024). A Deep Learning Approach To Face Shape Classification For Hairstyle, Beauty And Eyewear Recommendations. *Proceedings of the International Conference on Soft Computing and Machine Intelligence, ISCMI*, *2024*, 301–307. https://doi.org/10.1109/ISCMI63661.2024.10851601

Salim, B. V., Chyntia, Indrawan, J. O., Hidayat, J., Matthew, S., Mangkang, T. A. E., Hasana, S., & Permonangan, I. H. (2023). Face Shape Classification Using Swin Transformer Model. *Procedia Computer Science*, *227*, 557–562. https://doi.org/10.1016/j.procs.2023.10.558

Sanjar, K., Bang, S., Ryue, S., & Jung, H. (2024). Real-Time Object Detection and Face Recognition Application for the Visually Impaired. *Computers, Materials and Continua*, *79*(3), 3569–3583. https://doi.org/10.32604/cmc.2024.048312

Suriyalakshmi, V. C., Krishnamurthy, M., Sabarna, P., & Swetha, M. (2023). Spectacles Recommendation System Based on the Face Shape Using CNN Model. *2023 International Conference on Network, Multimedia and Information Technology, NMITCON 2023*. https://doi.org/10.1109/NMITCON58196.2023.10276139

Syed, T. A., Siddiqui, M. S., Abdullah, H. B., Jan, S., Namoun, A., Alzahrani, A., Nadeem, A., & Alkhodre, A. B. (2023). In-Depth Review of Augmented Reality: Tracking Technologies, Development Tools, AR Displays, Collaborative AR, and Security Concerns. In *Sensors* (Vol. 23, Issue 1). MDPI. https://doi.org/10.3390/s23010146

Tian, Y., & Ball, R. (2023a). Parametric design for custom-fit eyewear frames. *Heliyon*, *9*(9). https://doi.org/10.1016/j.heliyon.2023.e19946

Vidal-Balea, A., Blanco-Novoa, Ó., Fraga-Lamas, P., & Fernández-Caramés, T. M. (2021). Developing the next generation of augmented reality games for pediatric healthcare: An open-source collaborative framework based on arcore for implementing teaching, training and monitoring applications. *Sensors*, *21*(5), 1–24. https://doi.org/10.3390/s21051865

Wang, J., Liu, P., Liu, J., & Xu, W. (2023). *Text-guided Eyeglasses Manipulation with Spatial Constraints*. http://arxiv.org/abs/2304.12539

Wang, R., Huang, J., Zhang, J., Liu, X., Zhang, X., Liu, Z., Zhao, P., Chen, S., & Sun, X. (2024). FacialPulse: An Efficient RNN-based Depression Detection via Temporal Facial Landmarks. *MM 2024 - Proceedings of the 32nd ACM*

International Conference on Multimedia, 311–320. https://doi.org/10.1145/3664647.3681546

Zou, Z., Jia, D., & Tang, W. (2025). Towards unsupervised learning of joint facial landmark detection and head pose estimation. *Pattern Recognition*, *162*. https://doi.org/10.1016/j.patcog.2025.111393

Zwoliński, G., Kamińska, D., Haamer, R. E., Pinto-Coelho, L., & Anbarjafari, G. (2023). ENHANCING EMPATHY THROUGH VIRTUAL REALITY: DEVELOPING A UNIVERSAL DESIGN TRAINING APPLICATION FOR STUDENTS. In *Medycyna Pracy* (Vol. 74, Issue 3, pp. 199–210). Nofer Institute of Occupational Medicine. https://doi.org/10.13075/mp.5893.01407