
# Faculty of Informatics and Computer Science

**Information System**

**Automated system that detect different face types and select appropriate fitting glasses.**

By: Marihan Mamdouh

**Supervised By**

Professor Ann Nossier

**July 2020**

# Abstract

Recommending the appropriate fitting glasses that perfectly match each face shape. This process is done by recognizing the face via extracting the facial features and determine its type so the user choosing the glasses could have the best choice that matches his or her face in the right way. However, the traditional ways are useless as people try a lot of frames one by one to find the one that satisfies them and matches their face which is time-consuming and may end up wearing a glasses that does not fit their face shape perfectly, hence we are suggesting an automated process of selecting the glasses which is recommending the glasses for each face type in addition to that allow the user to try them on their face virtually in front of a web camera so that it saves a lot of wasted time and also enhance user's experience in selecting his or her glasses. Moreover, users use this recommendation system to try different glasses that is categorized for each face shape and allow the user to choose from one category but with a variety of frames within the specified category for each face type. finally, this done by utilizing deep learning and Convolutional neural networks to extract the facial features and recognize each face then determine if the user face shape is belong to which class of the four classes which are round, square, heart, or oval types then recommend the frame shape based on that determination of the face.

# Attestation &Turnitin Report

I understand the nature of plagiarism, and I am aware of the University's policy on this.

**Signature** **Date**

# Acknowledgements

This project could not have been done without the help and effort of associative professor Ann Nosier and I would like to show appreciation as she stood by my side for the whole year and supported, motivated and encouraged me a lot and provided me also with several research resources and information that helped me doing this project.

Finally, I would like to thank my family for their continuous motivation and support that kept me going through all this journey at the university.

# Table of Contents

# List of Figures

# List of Tables

# 1 Introduction

## 1.1 Overview

Why do glasses look good on display or models but not the same when you try them on? It could be that as the glasses' shape is not suitable for your face type, and it is very overwhelming for anyone who wears glasses to pick out a pair. besides, determining your face type plays an important role in selecting the right pair of glasses. Moreover, different styles look very different on different face shapes. Furthermore, there are fashion rules based on fashion designers' views that set-up the rules of how glasses fit the person's face type and the factors for determining each face type. however, these rules should be put into consideration while selecting the glasses that match different face types. In this project, we propose a recommendation system to find a suitable glasses that fit the face of the user, additionally, this process is done through facial feature extraction using the convolutional neural network which is used to determine the face type by classifying face shapes into four different face shapes which is round, oval, heart and square, as a result, the system will be recommending a glasses which perfectly fits each face type, through capturing the user face as input and supply the user with the type of their face as an output. In addition to that, the shape of glasses they should wear, also a list of glasses that suits them and allow them to search in the same category for their faces. This will facilitate choosing the pair of glasses they should wear, save time, and will be more convenient to the users.

## 1.2 Problem Statement

People who wear glasses find it very hard to pick out a pair of glasses. There are a large number of styles and each style looks very different on different face types, as a result, you have to know which type of face-types do you have to know which style that fits your face perfectly. hence the problem here is how to choose a pair of glasses that match your face and classifying each face type.

## 1.3 Scope and Objectives

**Scope** of the project is creating a recommendation system that is based on CNN algorithm to classify a person's faces and determine whether this face oval, round, square, or heart and reach the best accuracy, after that recommend the best match pair of glasses for each face type.

**The objective** is to help make the process of choosing the glasses easier since it will not be mandatory for them to spend a lot of time trying a lot of glasses one by one, and go to different shops to find the perfect match glasses on their faces.

This recommendation system designed to solve this problem and make people's life easier by providing a system that captures the person's face and determine the face type then recommend the appropriate glasses that perfectly match each face type and also provide the user with another option which is a list that contains a variety of glasses that he/she can choose from. Since each face shape has a specific frame that match it.

## 1.4 Report Organization (Structure)

- This dissertation is about the work through the whole project, this report is divided into sections, each section describes specific thing in the project.

- First section is the Introduction it explains the overview of the whole project, problem statement, scope, and objectives of the project, report organization structure, work methodology which contains all the steps from the beginning until finishing the project and work plan each task with its start date and end date.
- Second section is the related work which contains the background that describes all the terminologies used in the project beginning from the meaning of face recognition until all the details that will be used in the whole project. in addition to that literature survey that holds the techniques in other research papers and a comparison between those techniques.
- Third section is about the proposed solution with the detailed methodology.
- Fourth section includes the implementation of the system and the algorithms used.
- Fifth section is about the evaluation and testing of the work.
- Sixth chapter discusses the results of the experiments
- Seventh, the last section, includes a summarization of the whole project and future work.

## 1.5 Work Methodology

- choose the idea of the project.
- writing the proposal.
- Read about deep learning, image processing, and face recognition
- Study and analyze information about the convolutional neural network and different algorithms
- analyze research papers and related work.
- choosing methodology.
- writing the interim report.
- Study python programming language
- gathering dataset
- study many ways of data preparation
- implement data preparation (pre-processing)
- Implement classification models
- training and testing
- work on accuracy improvement
- working the GUI that allows the user to choose the best match glasses, browse frames, and try them on their face.
- writing thesis.

## 1.6 Work Plan (Gantt Chart)

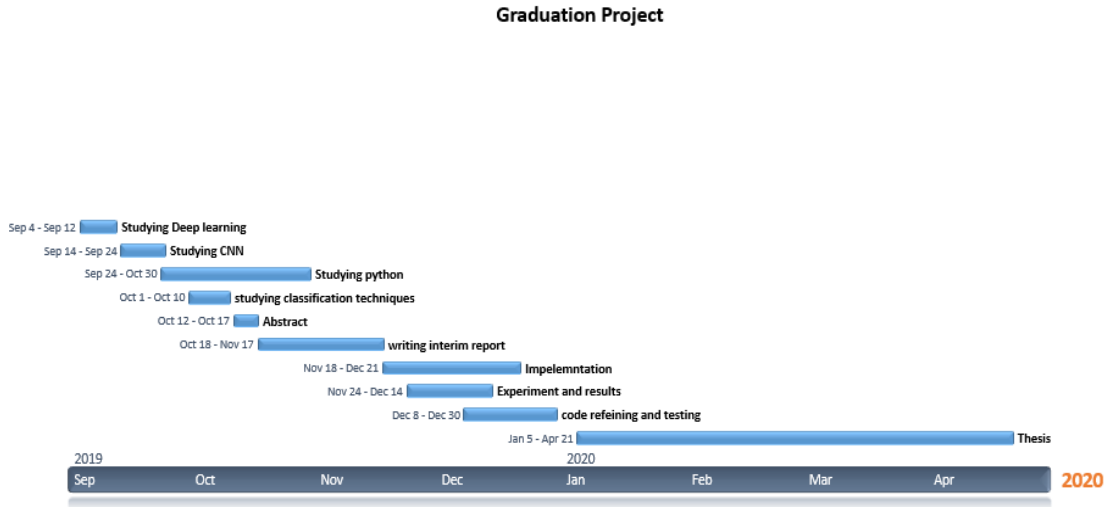



**Figure 1 Gantt Chart**

# 2 Related Work (State-of-The-Art)

## 2.1 Background

### 2.1.1 Deep learning Neural Network

**Deep learning** Deep learning is a subfield of machine learning inspired by the structure of the human brain.

In deep learning, the network learns from the data by itself while training the network, it does not need to extract the features Since the network does this on its own.

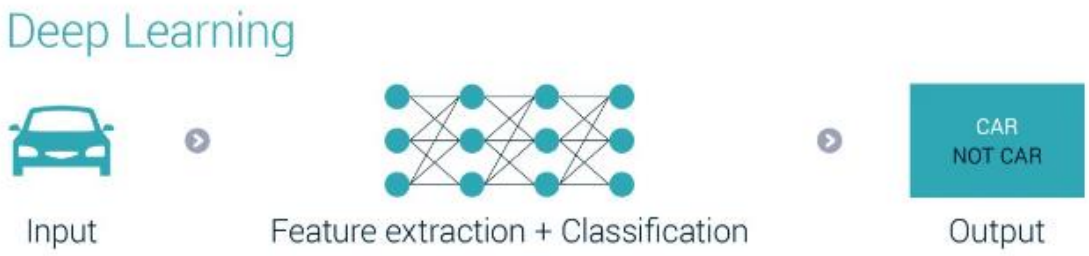



**Figure 2**

**Neural network as a classifier:**

A neural network consists of neurons, organized in layers, which transform input into some output. Every unit receives an input, practices a nonlinear function to it after that transfers the output on to the following layer. Commonly, the networks are assigned to be feed-forward: the output of each layer is the input for the following layer.

**A neural network** consists of cascaded layers the first layer is the input layer the middle layers are the hidden layers and the final layer is the output layer.

**learning happens in two ways**:

- **supervised learning**: when the deep learning model learns from labeled data
- **unsupervised learning**: when the deep learning model learns from unlabelled data

The neural network training phase have two major steps in which they loop in order to reach the optimal value.

- Step1: adjusting random weights and do forward propagation.
- Step2: look at the errors and do backward propagation.

**Forward propagation:** is calculating "input*weights +bias =activate"

**weights(w)**: is a parameter that transforms input data to the next layer.

**x:** is the input.

**bias (b):** shift the curve right and left to get the best fit

**backward propagation**: is a process of calculating errors and update them sequentially

**Convolution**: the convolution concept is applying a filter to an input that is receptive which means results in activation and keep repeating of sliding the filter on a given input until transforming it to a feature map.

**Activation function**: is a non-linear function that is used to learns complex behavior of the neuron as.it represents how the neurons communicate in the brain with each other and decide a certain neuron should be activated or not depends on a certain threshold it reaches. as linear functions are not strong enough to model many kinds of the data activation function is applied.

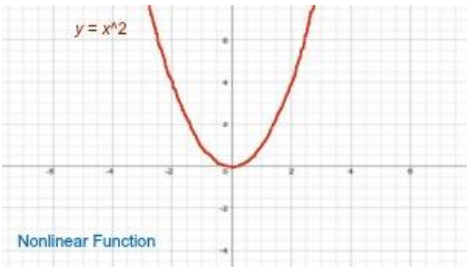



**Figure 3 Non-linear Function**

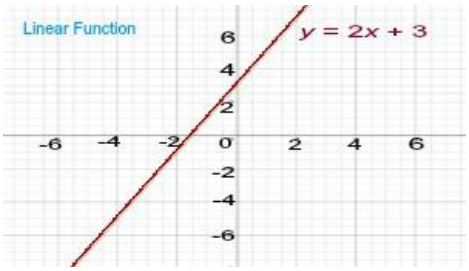



**Figure 4 linear function**

**Activation function types:**

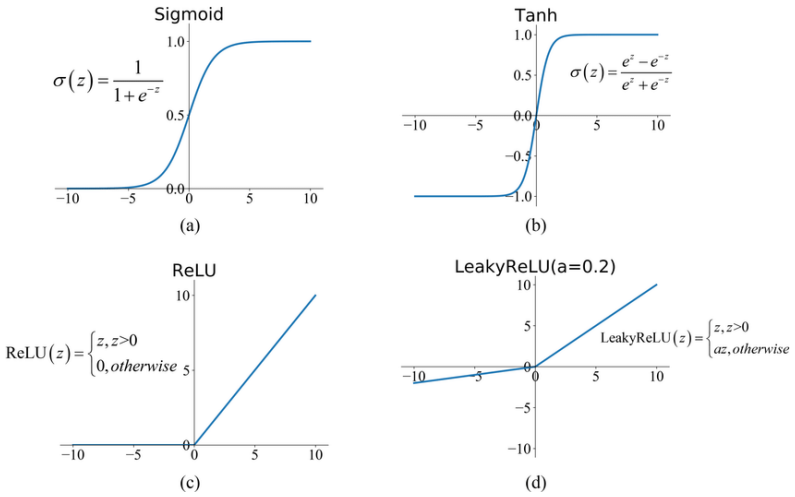



**Figure 5 Activation functions example**

**RELU** is the most commonly used activation function.

**Optimizer:** is an algorithm used for minimizing a given loss function by making it close to zero as much as possible by changing the learning rate and weights to make the model predictions as accurate as possible.

**Gradient decent:** it is a way of findings which way you should move while minimizing the error (in the backward prop)

**Learning rate:** a small number ranging between 0.1 and 0.001 that changes at each iteration to get close to minimum loss function.

Too low learning rate will lead to slowing the progress of the learning process as the update weights in the network are very tine

Too high learning rate it may causes unwanted divergent behavior in your loss function.

**Epoch:** single path of the data and same data keep passes to the network through a number of epochs and during this cycle the model keep learning.

**Batch size:** how many data we want to send to data at once.

**Max pooling:** the process of extracting maximum values from an image which holds the important feature from the input to shrink the size of the image.

**Drop-out:** technique used to avoid overfitting by choosing neurons randomly during training and drop them, which means in forward pass their contribution removed. on the backward pass there is no changes in the neurons weight.

**Normalization:** to prepare the data for training it should be normalized by scaling the numerical data to be between 0 and 1 to make the training process much easier.

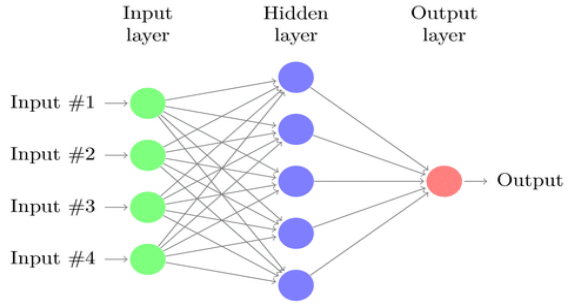



**Figure 6 Simple Neural network Example**

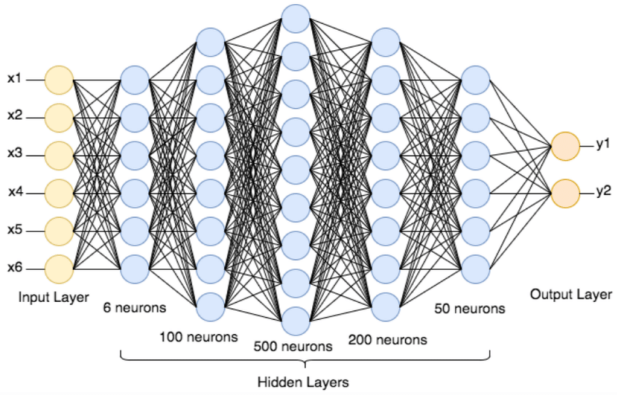



**Figure 7 Deep Neural Network ExampleCNN**

**A Convolutional Neural Network**: is a Deep Learning algorithm that can learn from data, it used for classification problems and image analysis. Convolutional Neural Network is a combination of convolutional concept and neural networks.

**CNN** is a combination of Convolutional layers, Pooling layers, and normalization layers, and fully connected layers Moreover, activation functions it also uses the convolution layers and pooling layers for model activation.

**CNN** composed of neurons each neuron takes inputs and perform some operations like assigning weights and biases to an object in the image to differentiate between one from the other object and produces some output.

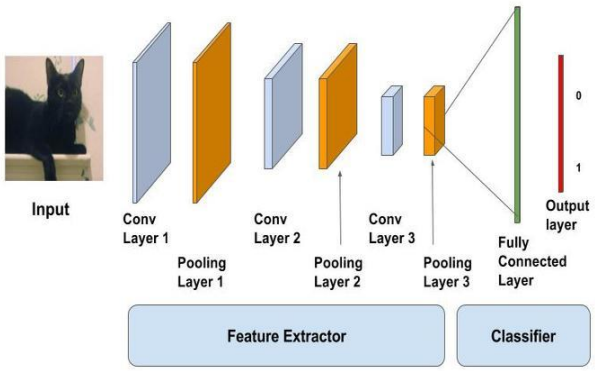



**Figure 8 Image classification example**

### 2.1.2 Machine learning

**Machine Learning**: is a technique that analyses data using algorithms and learns from that data, then decides about new data. furthermore, it instructs the computers to be able to learn and act as people do hence enhance their learning over time by feeding them with information from real-world interactions and observations.

The algorithm has the ability to analyze large amounts of data and learn the features from a given data and start classifying based on what the network learned from the extracted features of the data that the network trained on and later-on the network can classify data that never seen before.

The machine is trained using a large dataset.

The machine learns from the trained data without being told how to do so.

In machine learning, the features must be extracted first then the network start classifying unlike deep learning.


**Figure 9**

### 2.1.2.1 Types of machine learning:

#### *2.1.2.1.1 supervised machine learning*

The majority of practical machine learning uses supervised learning.0

Supervised learning is when you have an input v(x) and an output (Y) and you use an algorithm to learn the mapping function from the input to the output. Y = f(X)

The algorithm is learning from the training dataset.

The algorithm makes predictions on the training data.

Supervised learning can be categorized in classification or regression,

when an acceptable level of performance achieved the learning stops.

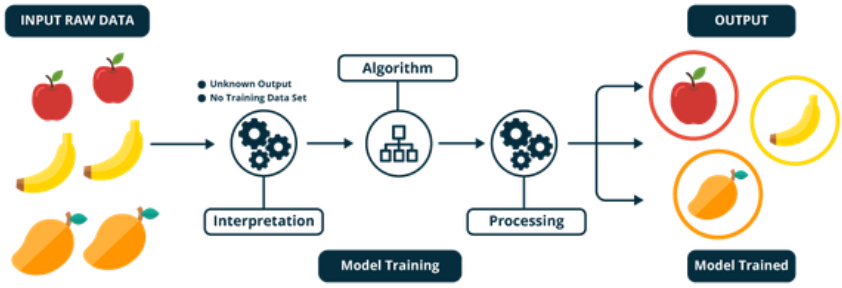



**Figure 10 Supervised learning Example**

#### *2.1.2.1.2 unsupervised machine learning*

you only have input, but you do not have output, unsupervised learning aims to learn more about data.

**unsupervised** learning can perform more complex task than supervised learning

**unsupervised** learning goal is to have the algorithms recognize patterns within the training data sets and classify the input objects based on the patterns that the system itself classifies. The algorithms analyze the datasets by obtaining useful features or information from them.0

#### *2.1.2.1.3 semi supervised Machine Learning*

This type of machine learning between supervised and unsupervised machine learning.

### 2.1.3 Image processing

Image processing is a technique to apply some operations on an image to extract some useful features or to get an enhanced image.


**Figure 11 Example of performing some image processing operation on an image displaying the RGB layers.**

### 2.1.4 Facial recognition

face detection or Face recognition is a method or a way of detecting a human face using algorithms and technology.

A facial recognition system utilizes biometrics to map facial features from an image or video that compares the information of known faces in a database to find if there is matching between them.

**Figure 12 face detection Example**

## 2.2 Literature Survey

In this paper, the first classification method is LDA it utilized python sci-kit- learn library in the classifier testing and training. the linear discriminant analysis combines features and separate classes that can be utilized as a way of dimensionality reduction or a classifying technique while training face shape classifier. Vectors with high dimensional features vector are eliminated into 2 different components before the classification.

The second classifier used is the support vector machine (SVM) which classifies the image through the feature space's points are mapped with the widest gap between classes as much as possible. Two kinds of SVM approaches are being used the first one is (SVM-LIN) and the other one is (SVM-RBF).

The third one is ANN or artificial neural network it is a network that consists of computational units named artificial neurons, they are interconnected. this artificial neuron goal is to approximate functions which are non-linear through a set of the linear transformation in addition to activation functions non-linear. The ANN utilized two layers which are hidden in addition to two neurons in the first layer and five neurons in the second layer, the estimation of the parameters was made through quasi newton method known as LBFGS.

The last classifier is KNN which classifies the membership of the training sample through measuring its distance with the rest of the training samples through using the Euclidean instance equation. The model looks for the closet five neighbours for classification.

in this paper, Pornthep Sarakon and Theekapun Charoenpong used a support vector machine for face shape classification. the algorithm consists of three steps, face plane identification, head segmentation, face shape classification. classify the face shape into four groups. The four type of the face shape are round face type, long face, square face, ellipse face [2]

this paper Hinal N. Shah used 3D face reconstruction from a single 2D image to measure the forehead, face length, and jawline, Using the facial features measured in 3D space, and then evaluate ma different machine learning algorithms.[3]

in this paper, Wisuwat Sunhem and Kitsuchart Pasupa this paper used active appearance model (AAM) classification model and face segmentation technique classify face shapes into five different face shapes which is round, oval, oblong, square and heart and compare them with LDA, ANN and SVM the best accuracy found is support vector machine with Radial Basis function kernel [4]

## 2.3 Analysis of the Related Work

**Table 1 Comparison between different methodologies**

| Paper | Dataset | Technique used | Accuracy |
|---|---|---|---|
| Adonis Emmanuel DC. Tio [1] | ImageNet 2012<br>500 images for celebrity faces | Compares inception v3 that uses CNN with 5 different classifiers. | |
| | | LDA | 61.6% |
| | | SVM_LIN | 55.2% |
| | | SVM_RBF | 50.6% |
| | | ANN | 54.0% |
| | | KNN | 64.6% |
| Pornthep Sarakon, Theekapun Charoenpong [2] | 3d whole human body data | Support vector machine | 73% |
| Wisuwat Sunhem and Kitsuchart Pasupa [4] | 1,000 images of different face shapes of women. | LDA<br>ANN<br>SVM- Linear<br>SVM-RBF | 58%<br>60%<br>64%<br>72% |

# 3 Proposed solution

## 3.1 Solution Methodology

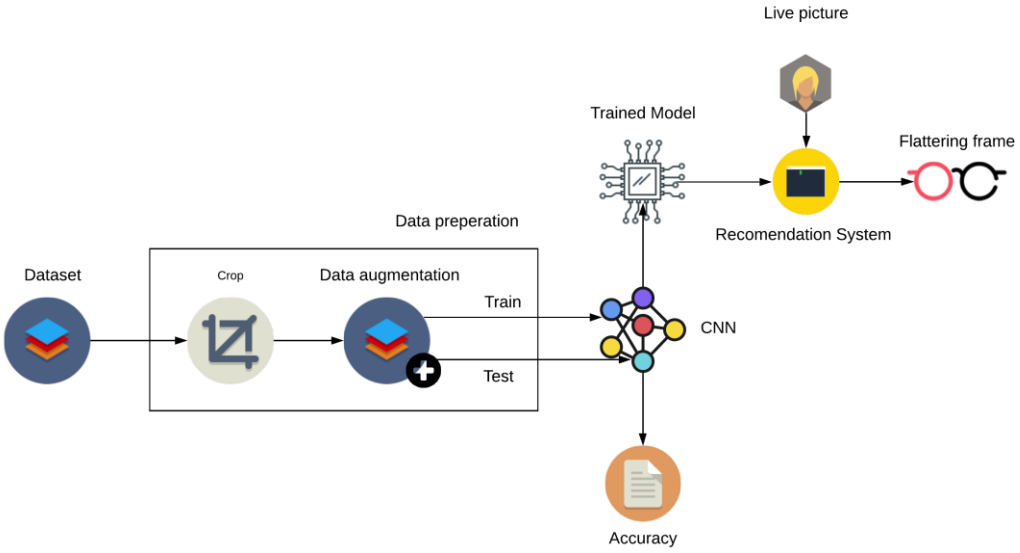



**Figure 13 The Flow of the project's structure**

### 3.1.1 Dataset

Dataset used in this project has been downloaded from Kaggle is a combination of 4000 images of celebrities. All the images are labeled and categorized in 4 face shapes round, heart, oval, square.

### 3.1.2 Pre-processing

We applied crop to the whole dataset by cropping the face only from the whole image to make the learning rate for the training set increase by focusing on the important features and ignore unwanted features of the image.

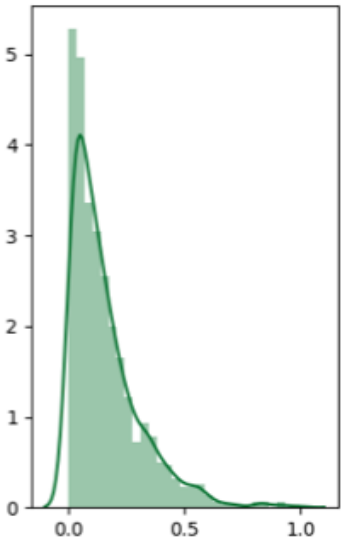
#### 3.1.2.1 Resizing Images

All images are resized into 300x300x3 to be suitable for input to the neural network.

3.1.2.2 Normalization

Is on of pre-processing major steps, it is being done in order to have best practices for training a neural network all numeric values (all pixels) from 0 to 255 are rescale into a 0 to 1 range to speeds up the learning process, and for faster convergence.

.



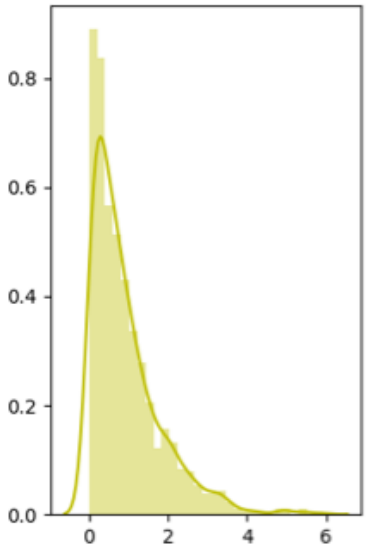




**Figure 9 original data**

**Figure 10 normlized data**

3.1.2.3 One hot encoding

Transform labeled classes to a vector of zeros and ones. The length of the vector depends on the number of classes that the model will be classifying it.

In this project, the model is classifying four different categories, each category of the four categories

will be encoded into a vector each vector will have a length that is equal to the number of the output categories which are four outputs. our categories are round, square, heart, or oval for the first category all the elements will be zeros except the first element will be one. for the second category, the second element in the vector will be one and all the elements will be zeros. for the third category, all elements will be zeros except the third element will be one. Finally, the fourth category all the elements in the vector will be zeros except the fourth element will be one.

**Figure 14 Round [1,0,0,0]**

**Figure 15 Square [0,1,0,0]**

**Figure 16 Heart [0,0,1,0]**

**Figure 17 Oval [0,0,0,1**

### 3.1.2.4 pickle file

A pickle file allows objects structure like dictionary or array to be converted to a byte stream so it can be saved as a file. “pickling “is serializing and “unpickling” is dieselizing the data, which used to save space and be much faster when being transferred over a network or to be used later. Dieselizing (unpickling) is loading the “PKL” file back during runtime into the memory of the program.

### 3.1.2.5 Augmentation

overfitting occurs when the dataset is not enough to train the network.

Data Augmentation is applied as one of the major and very effective ways to prevent Over-fitting to occur.

creating more images from the images that already exist this process is done by applying some operations to the image like changing the size or orientation data augmentation is applied using Keras using the Image Data Generator instance.

**Arguments Used**:

rotation shift range= 30 rotate the image with range 30

width shift range = 0.1 shift the width of the image with range 0.1

height shift =0.1 shift the height of the image with range 0.1

shear the image with range 0.2: Shear the image in a clockwise direction.

horizontal flip: randomly flip the image horizontally

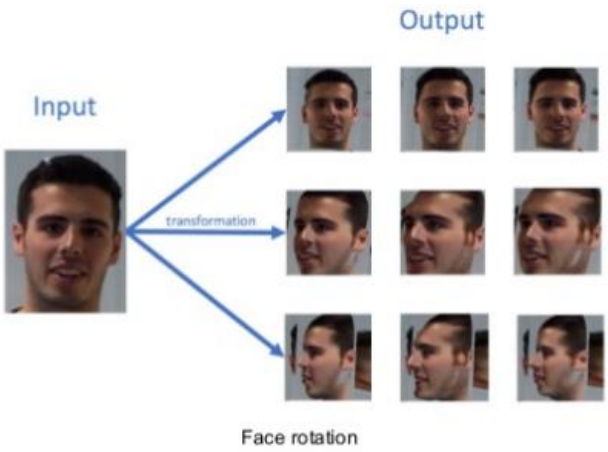



**Figure 18 Data Augmentation Example**

### 3.1.3 Implement Classifiers

#### 3.1.3.1 Fully Connected Neural Network

First The first trial was using facial landmarks it is done by extracting 17 points of the face send them over Neural network. the 17 points (x, y) identifying the position on the image so we collect these (17x2=34) features normalized by width and the height of the image and train a Fully connected neural network model with varying the number of layers and tuning the parameters which the accuracy obtained an accuracy of classification method in the range 52%.

#### 3.1.3.2 Convolutional neural network

to achieve the objective of the project, deep learning needs to be understood to implement the neural network to classify the faces.

A convolutional neural network that consists of eleven layers to classify the images:

one input layer

seven convolutional layers with RELU activation function each one of them followed by max-pooling layer.

two Fully connected layers with a RELU activation function.

one Output layer with the SoftMax activation function. (with output four classes).

Number of epochs uses=200 epochs

Number of classes = 4 classes

batch size =4.

##### *3.1.3.2.1 Input layer*

The first layer is the input layer supplied by the image data represented by a 3-dimensional array. the input layer provides the network with input or features.

##### *3.1.3.2.2 hidden layers*

first, we applied seven convolution layers which contain 32 activation map in the first convolution layer and filter of size (3,3) and stride (1,1) which means the filter will move or slid by 1 pixel at a time across each 3*3 block pixels. Each pixel in the image is multiplied element by element with the filter and the result of the multiplication is added to obtain the activation map (the output) which will be passed as an input to the next layer.

The second activation function is applied after each convolution layer the activation function is utilized as a decision-making body at the output of a neuron. we used reflected linear unit activation function (RELU).

**Relu Activation function**

when the input value is positive the more active the neuron will be.

Relu is applied as it transforms the input as either zero or the input itself.

If the input is zero RELU function output result will be zero.

If the input is greater than zero, the RELU activation function output value will be the actual value.

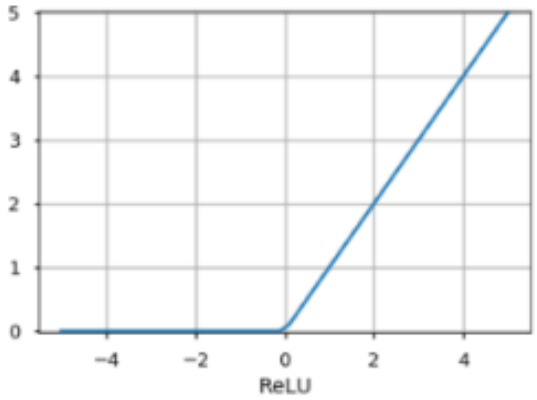



**Figure 19 Relu Activation Function**

after that, we applied max pooling of filter size (2,2,) and apply the maximum operation over each 2*2 block of the image to eliminate the computation and number of parameters to avoid overfitting.

The same previous process is repeated for the seven convolutional layers.

Batch normalization is applied after the last convolution layer.

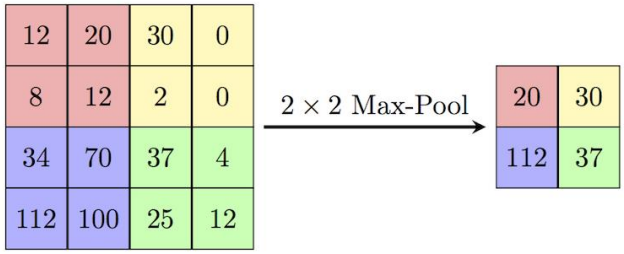



**Figure 20 Max pooling Example**

### *3.1.3.2.3 Fully connected layers (dense layers)*

in image classification, to classify the object in an image some spatial information is not necessarily needed, as a result, the output of the last convolutional layer is flattened into one long vector.

This vector is the input for the Fully Connected layers. The weighted sum of the deep features is being done across all the spatial locations in the image.

In this model, two fully connected layers are applied

Fully connected input layer flattens the output came from previous layers and converts them into a single vector that can be an input for the next layer.

The first fully connected layers predict the correct label by taking the inputs from the feature analysis and applies weights.

The Fully connected output layer predicts the final probabilities for each label.

### *3.1.3.2.4 Output layer*

The model consists of one output layer This is the layer which gives out the predictions. either the face is round, heart, oval or square.

**SoftMax Activation function**

The SoftMax activation function is applied to make all the output values range between zero and one.

The activation function to be used in this layer is the SoftMax activation function.

The model id a Multiclass classification as a result SoftMax is used.

work as class probability all the sum is exactly equal to one.

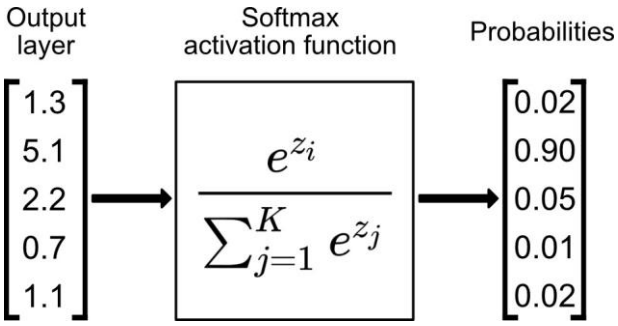



**Figure 21 SoftMax Activation Function Example**

### *3.1.3.2.5 Cost/Loss Function*

Loss function is the difference between the predicted output and the actual output the network produced.

The Loss function is applied to calculate the gradients.

The gradient used to update the weights of the neural network.

we update the weights of the network with multiple trials or iterations with different weights to minimize cost/loss function to ensure few errors as much as possible.

when there is a lot of mistakes the loss is high.

when the is a few mistakes the loss is low.

### *3.1.3.2.6 Optimizer*

Stochastic Gradient Descent (SGD) optimizer is applied using Keras to ensure accurate predictions by minimizing the cost function as cost function predicts if the training direction of increase is moving in the right direction.

weights and cost function are updated every iteration until reach a global minimum (optimal point reachable).

*3.1.3.2.7*

*Table 2 CNN model structure*

| Layer (type) | Output shape | Para # |
|---|---|---|
| input_1 (Input Layer) | (None, 300, 300, 3) | 0 |
| conv2d_1 (Conv2D) | (None, 300, 300,32) | 896 |
| activation_1 (Activation) | (None, 300, 300, 32) | 0 |
| max_pooling2d_1 (MaxPooling2) | (None, 150, 150, 32) | 0 |
| conv2d_2 (Conv2D) | (None, 150, 150, 64) | 18496 |
| activation_2 (Activation) | (None, 150, 150, 64) | 0 |
| max_pooling2d_2 (MaxPooling2) | (None, 75, 75, 64) | 0 |
| conv2d_3 (Conv2D) | (None, 75, 75, 64) | 36928 |
| activation_3 (Activation) | (None, 75, 75, 64) | 0 |
| max_pooling2d_3 | (MaxPooling2 (None, 37, 37, 64) | 0 |
| conv2d_4 (Conv2D) | (None, 37, 37, 128) | 73856 |
| activation_4 (Activation) | (None, 37, 37, 128) | 0 |
| max_pooling2d_4 (MaxPooling2) | (None, 18, 18, 128) | (None, 18, 18, 128) |
| conv2d_5 (Conv2D) | (None, 18, 18, 128) | 147584 |
| activation_5 (Activation) | (None, 18, 18, 128) | 0 |
| max_pooling2d_5 (MaxPooling2) | (None, 9, 9, 128) | 0 |
| conv2d_6 (Conv2D) | (None, 9, 9, 256) | 295168 |
| activation_6 (Activation) | (None, 9, 9, 256) | 0 |
| max_pooling2d_6 (MaxPooling2) | (None, 4, 4, 256) | 0 |
| conv2d_7 (Conv2D) | (None, 4, 4, 256) | 590080 |
| batch_normalization_1 (Batch) | (None, 4, 4, 256) | 1024 |
| activation_7 (Activation) | (None, 4, 4, 256) | 0 |
| max_pooling2d_7 | (None, 2, 2, 256) | 0 |

| | | |
|---|---|---|
| flatten_1 (Flatten) | (None, 1024) | 0 |
| dense_1 (Dense) | (None, 500) | 512500 |
| dense_2 (Dense) | (None, 100) | 50100 |
| dense_3 (Dense) | (None, 4) | 404 |
| Total params: 1,727,036<br>Trainable params: 1,726,524<br>Non-trainable params: 512 | | |

### 3.1.4 Accuracy

The CNN model classifies the images with accuracy 88% for training and 75% for testing.

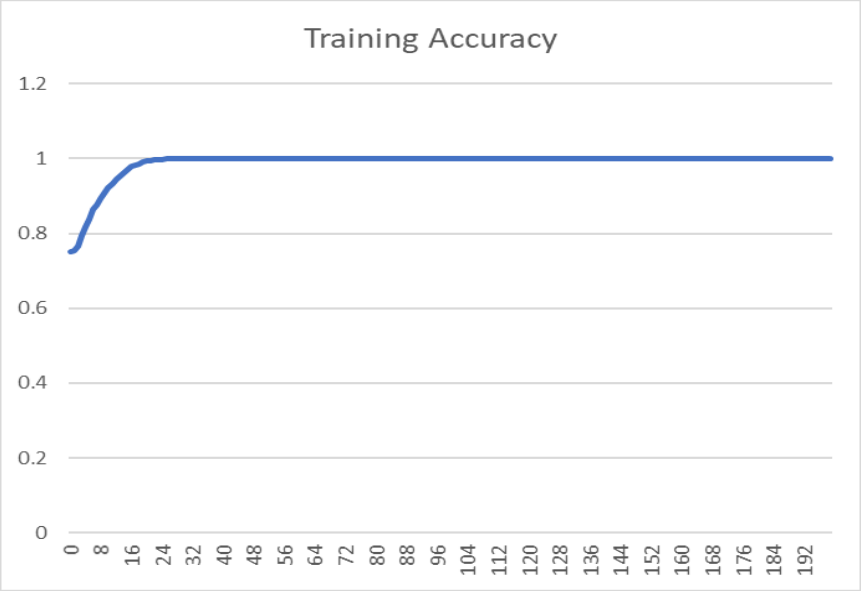



**Figure 22**

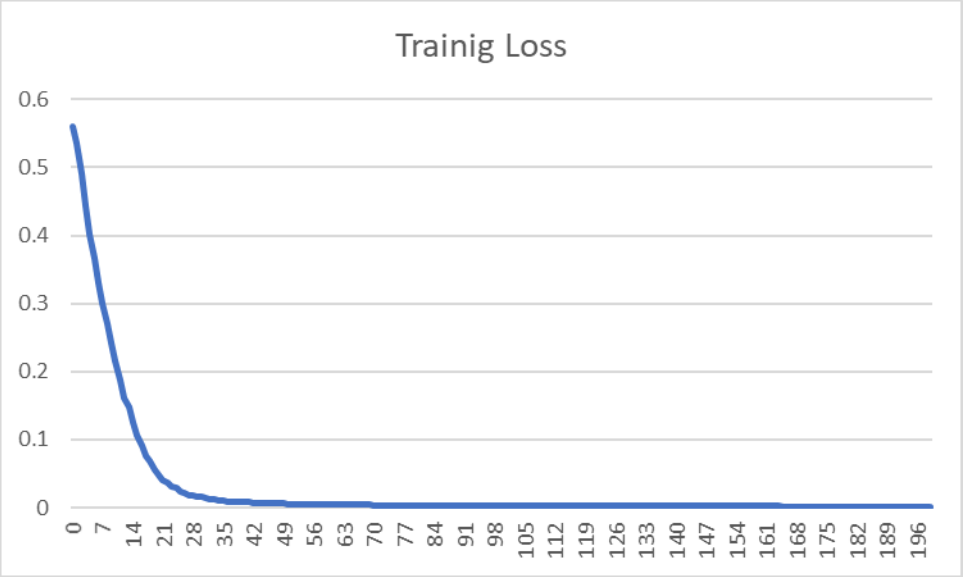



**Figure 23**

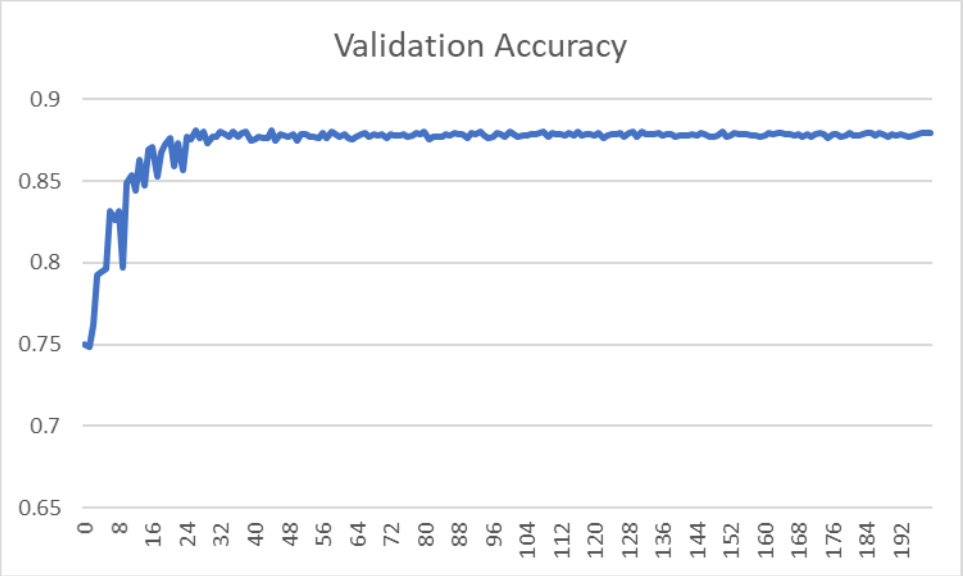



**Figure 24**

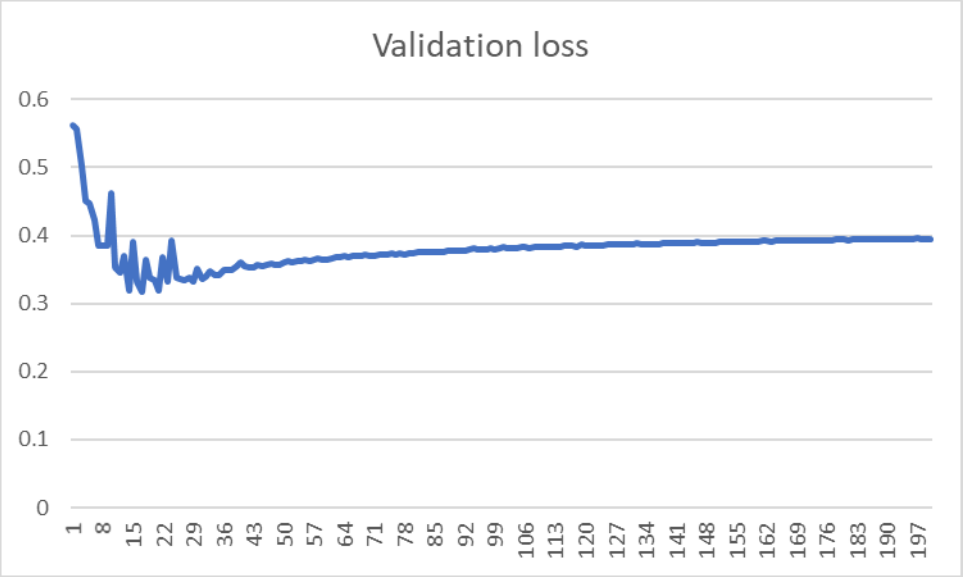



**Figure 25**

## 3.2 Functional/ Non-functional Requirements

### 3.2.1 Functional requirements

Functional requirements describe the behavior of the system, How the system should behave or run. The captured result can be expressed as functions or task the system is required to perform as much success as possible with minimum error it could reach.

**The functional requirements of this recommendation system:**

- ability to classify face type using CNN correctly.
- ability for the system to recommend the appropriate glasses that match the face shape.
- the ability for the user to try the recommended glasses on their face.

### 3.2.2 Non- functional requirements

The non-functional requirements outline the most important attributes of the system. attributes

can be considered to improve the existing system to make it more usable to the users.

The system can still work without one or more of the non-functional requirements.

**Usability**. The appearance of the user interface is user friendly and it is very easy for the users to interact with it. the colors of the screen are purple, and all the buttons are described and very clear for the user to understand.

**Performance**: The performance of the system should be fast and accurate

**Accuracy**: The system should detect the face type accurately.

## 3.3 Simulation set up

**Used libraries**

The classification of the faces was implemented in python and the GUI.

**NumPy**

Python programming language library performs mathematical operations on arrays

**Keras**

It is a Python open-source library used for creating deep learning models. It is a user-friendly framework for implementing deep neural networks.

**Sci-Kit learn**

library in python that includes tools for machine learning

**Pandas**

python open-source library that saves that data files as python objects columns and rows.

**OS**

python library that creates and removing directories.

**random**

python library that produces random numbers.

**cv2**

open-source computer vision library it is used to solve computer vision problems.

**T-kinter**

library in python, used to create GUI

**PIL**

python imaging library is an open-source library in python that performs operations on different image file formats.

**CSV**

comma-separated values library is used for saving data in tabular format

**Matplotlib**

**plotting library** in python used for plotting figures.

## 3.4 Design

This diagram describes the functionalities of the system and how the user interacts with it.

This describes sequences of actions that the system could perform by interacting with different users or actors.

there are two types of actors primary and secondary actors:

The primary actor is the actor who needs someone to assist them to use the system.

. secondary actors: external actors who provide the system with assistance.

A use case diagram captures many functionalities that are complete, which means that it shows the interaction between the system and the users from the beginning to the end.

**The user can do the following:**

Take a picture of his face and the system define their face type.

The user can try-on the recommend frame for their face type.

The user also can select a frame from a group of frames that recommended by the system and matches the user face type.

The user can also choose to browse frames from an online store.

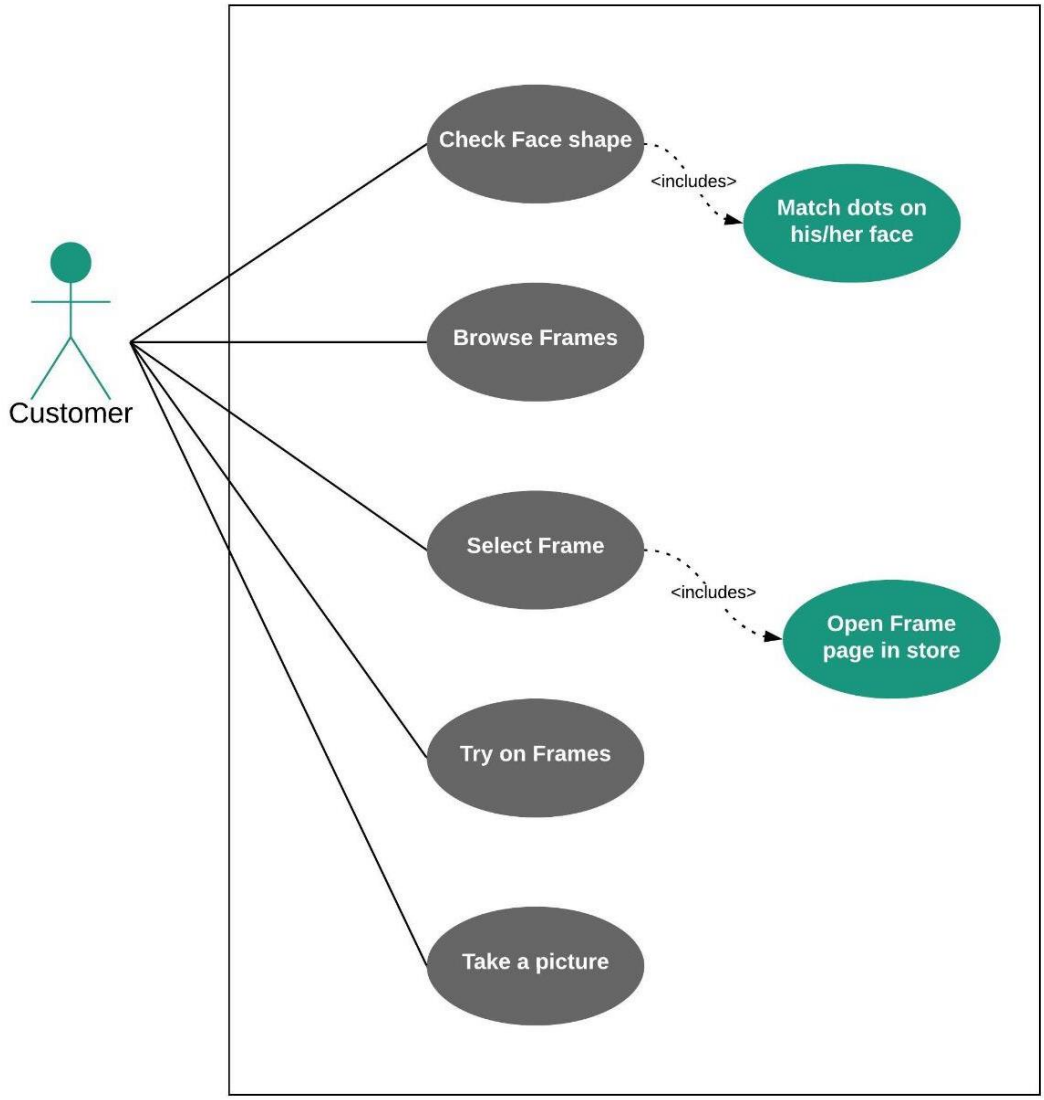



**Figure 26 Use case diagram**

# 4 Implementation

Description of the code flow

Firstly, the datasets are pre-processed and prepared, Read Images In this step, we store the path of the dataset directory into a variable then the function is created to load folders holds images into arrays using NumPy library. Cropping the images in this step, code for cropping is used to crop the face only from the image for a better learning. resizing all the images using cv2 library, normalize the images by dividing all the images over 255 to be in the same range between 0 and 1, besides, one-hot encoding to know the image belongs to which class of the four classes as all the classes will have a vector of 4 elements, one of the elements will be one and all the remain elements will be 0. 1 means this image belongs to which class of the four classes either round, square, oval or heart. the datasets are dumped and stored in a pickle file. Secondly, the CNN model is created using Keras library. It was created with 11 layers the first layer is the input layer that holds the input. then become the convolutional layers which are 7 layers each one followed by RELU activation function and max-pooling and the last layer of the 7 layers is following by batch normalization, after that 2 Dense layers is implemented to flatten the data and the last one is the output layer followed by a sigmoid activation function.

**The first layer, input layer:**

```
training = False  # True for training and False for testing
if training:

    ## load training data
    data_fid = open('data_preparation\\training_set.pkl', 'rb')
    [trainX, trainY] = pickle.load(data_fid)
    data_fid.close()

    images = Input(shape=inputShape)
```

**Figure 27**

**Convolutional layers:**

```
x = Conv2D(32, (3, 3), strides=(1, 1), kernel_initializer=init, padding='same')(images)
x = Activation('relu')(x)
x = MaxPooling2D(pool_size=(2, 2))(x)

x = Conv2D(64, (3, 3), strides=(1, 1), kernel_initializer=init, padding='same')(x)
x = Activation('relu')(x)
x = MaxPooling2D(pool_size=(2, 2))(x)

x = Conv2D(64, (3, 3), strides=(1, 1), kernel_initializer=init, padding='same')(x)

x = Activation('relu')(x)
x = MaxPooling2D(pool_size=(2, 2))(x)

x = Conv2D(128, (3, 3), strides=(1, 1), kernel_initializer=init, padding='same')(x)
x = Activation('relu')(x)
x = MaxPooling2D(pool_size=(2, 2))(x)

x = Conv2D(128, (3, 3), strides=(1, 1), kernel_initializer=init, padding='same')(x)
x = Activation('relu')(x)
x = MaxPooling2D(pool_size=(2, 2))(x)

x = Conv2D(256, (3, 3), strides=(1, 1), kernel_initializer=init, padding='same')(x)
x = Activation('relu')(x)
x = MaxPooling2D(pool_size=(2, 2))(x)

x = Conv2D(256, (3, 3), strides=(1, 1), kernel_initializer=init, padding='same')(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
x = MaxPooling2D(pool_size=(2, 2))(x)
```

**Figure 28**

**The two dense layers and the output layer:**

```
x = Flatten()(x)
x = Dense(500, activation='relu')(x)
x = Dense(100, activation='relu')(x)
outputs = Dense(num_classes, activation='sigmoid')(x)
```

**Figure 29**

after applying the CNN classifier, on a dataset of size equal to 500 images the accuracy was 40% but after increasing the dataset size to 4000 the accuracy reached 65% Thirdly after applying batch crop to extract the face only the accuracy reached 88% for training and 75% for testing. Thirdly, the performance and accuracy of the model were evaluated using the confusion matrix and Precision (positive predictive value), the Recall (measures how many times the model predicts correctly in the positive classes), and F1-score (F-measure) the results are stored in CSV file using pandas library.

```
#report that contains the results of the confusion metrix
predictions = model.predict(testX)
report = classification_report(testY.argmax(axis=1), predictions.argmax(axis=1))
print(report)
classification_report_csv(report, 'test/' + str(INIT_LR))
```

**Figure 30**

after training and evaluating the model, the model is saved to JSON file to be able to reuse. Fourthly the GUI code is used to recommend the appropriate glasses. The code of the CNN classifier is merged with the GUI code this is done by load the JSON file that contains the training model into the GUI using Keras.

```
def getFaceShapeCNN():
    json_file = open('model.json', 'r')
    loaded_model_json = json_file.read()
    json_file.close()
    model = model_from_json(loaded_model_json)
    # load weights into new model
    model.load_weights('model-045-1.000000-0.881117.h5') #training model
    # print("Loaded model from disk")

    #  Crop face automatically as trained cnn model
    extractor = FaceRegionExtractor()
    img_cropped = extractor.cutout_face_region(capimg)
    image = cv2.resize(img_cropped, (300, 300)) #resize
    dataX = np.array(image, dtype="float") / 255.0 #normilization  mn 0 l 1

    prediction = model.predict(dataX.reshape(1,300,300,3))
    classindx = np.argmax(prediction)
    if classindx == 0:
        data.faceShape = "heart"
    elif classindx == 1:
        data.faceShape = "oval"
    elif classindx == 2:
        data.faceShape = "round"
    elif classindx == 3:
        data.faceShape = "square"
```

**Figure 31**

# 5 Testing and evaluation

## 5.1 Face shape classification evaluation metricises

**Evaluation Metrics:**

The metrices used to test any classifier is confusion matrix, recall function, precision function and f1 function.

**Confusion matrix:** is a table used to measure the efficiency of a classification model.

**Confusion matrix has 4 results:**

1. false positive: if the predicted value is positive and wrong.
2. false negative if the predicted value is negative and wrong.
3. true positive if the predicted value is true and positive.
4. true negative if the predicted value is true and negative.

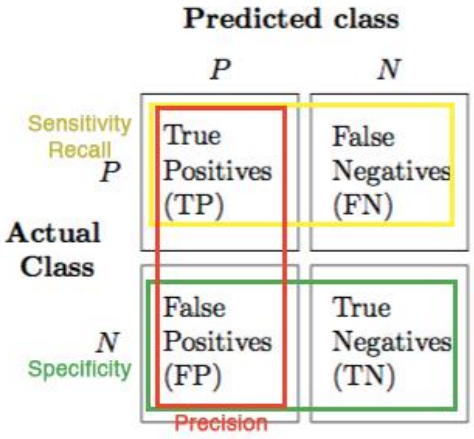



**Figure 32 Confusion Metrix**

**Precision Function:**

Calculates true positive predictions to the actual class value. measures how many times the model predicts true in all classes.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

**Figure 33 Precision formula**

**Recall function:**

calculates true positive predictions to all the predicted results, measures how many times the model predicts correctly in the positive classes.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

**Figure 34 Recall function formula**

**F1 measure function:**

is a combination between recall function and precision function

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

**Figure 35 F-measure Formula**

**Accuracy**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

**Figure 36 Accuracy formula**

## 5.2 Evaluation of Frames Recommendation

Recommend The right frame type for each Face shape. According to fashion designers' researches, there is a frame type for each face shape [7][8]0.

**The round face:** usually has full cheeks rounded chin, larger forehead, and have weaker lines and angles. So, frames with strong angles and lines will ad defining to the face and make the cheeks log fuller. Then we can conclude that the best frame type for round faces is rectangular.


**Figure 37**


**Figure 38**

**The oval face**: considered as a perfect shape because it is balanced with a small forehead and chin with high cheekbones. So, the best shape for oval face is square frames as they look good with high cheekbones.

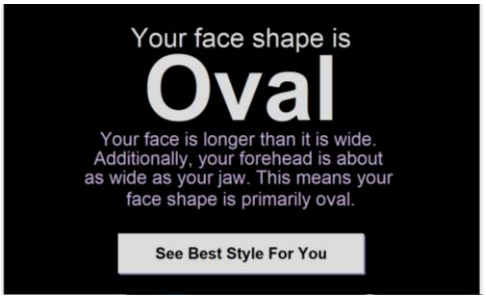



**Figure 39**

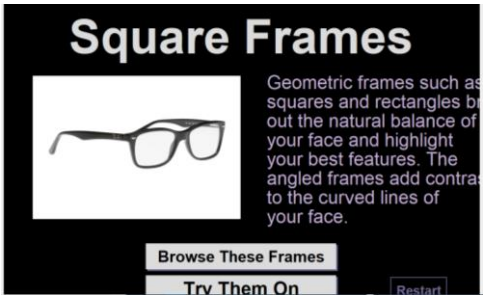



**Figure 40**

**The heart-shaped face:** this shape has two features that distinguish it from other face shapes, high cheekbones, and jawline that tapers to a defined chin. So, the choice is thin frames as they minimize the face top part and make the illusion of a wider chin.

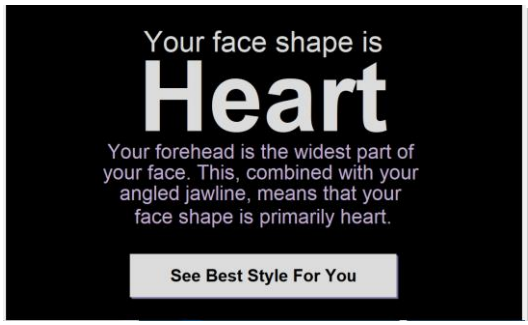



**Figure 41**

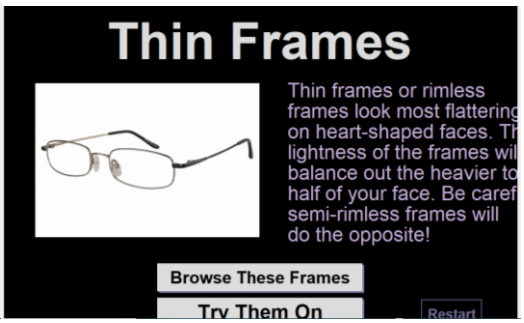



**Figure 42**

**The square face:** this face jawline is strong and angular features the best fit for it is thin frames with lots of curves to soften the face angles. For this face shape, the system will recommend round-shaped frames.


**Figure 43**

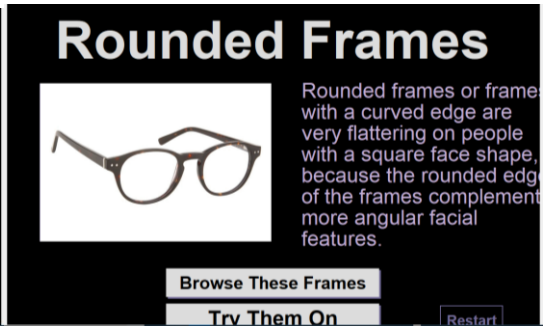



**Figure 44**

# 6 Results and Discussions

the following results after testing the CNN model that was trained using dataset of 4000 images.

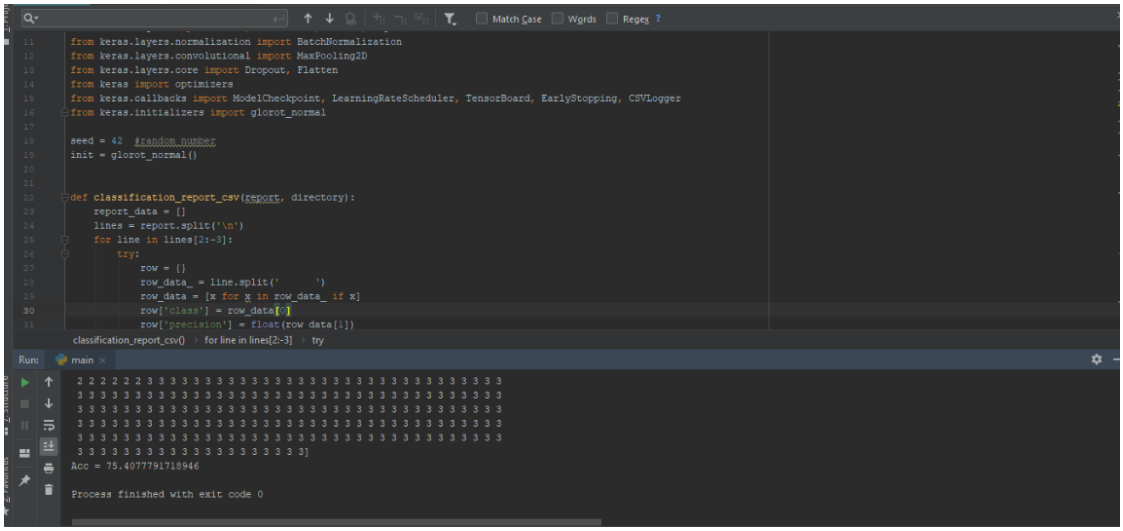



**Figure 45 Testing Accuracy**

In Figure 46 we illustrate the results of the evaluation metrics in each one of the face shapes classes.

T25

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | class | precision | recall | f1_score | support | |
| 2 | 0 | 0.75 | 0.7 | 0.73 | 200 | |
| 3 | 1 | 0.81 | 0.86 | 0.83 | 200 | |
| 4 | 2 | 0.74 | 0.79 | 0.76 | 198 | |
| 5 | 3 | 0.71 | 0.66 | 0.69 | 199 | |
| 6 | | | | | | |
| 7 | | | | | | |
| 8 | | | | | | |

**Figure 46 Evaluation of the CNN model**

**Table 3 previous work Vs. our approach**

| Paper | Dataset | Technique used | Accuracy |
|---|---|---|---|
| Adonis Emmanuel DC. Tio [1] | ImageNet 2012<br>500 images for celebrity faces | Compares inception v3 that uses CNN with 5 different classifiers. | |
| | | LDA | 61.6% |
| | | SVM_LIN | 55.2% |
| | | SVM_RBF | 50.6% |
| | | ANN | 54.0% |
| | | KNN | 64.6% |
| Pornthep Sarakon, Theekapun Charoenpong[2] | 3d whole human body data | Support vector machine | 73% |
| Wisuwat Sunhem and Kitsuchart Pasupa[4] | 1,000 images of different face shapes of women. | LDA<br>ANN<br>SVM- Linear<br>SVM-RBF | 58%<br>60%<br>64%<br>72% |
| Our approach | 4000 images that is labelled and divided into 4 classes | CNN | 75% |

The following figure shows the accuracy of our approach in comparison with previous work. According to it, our approach has reached the highest accuracy.

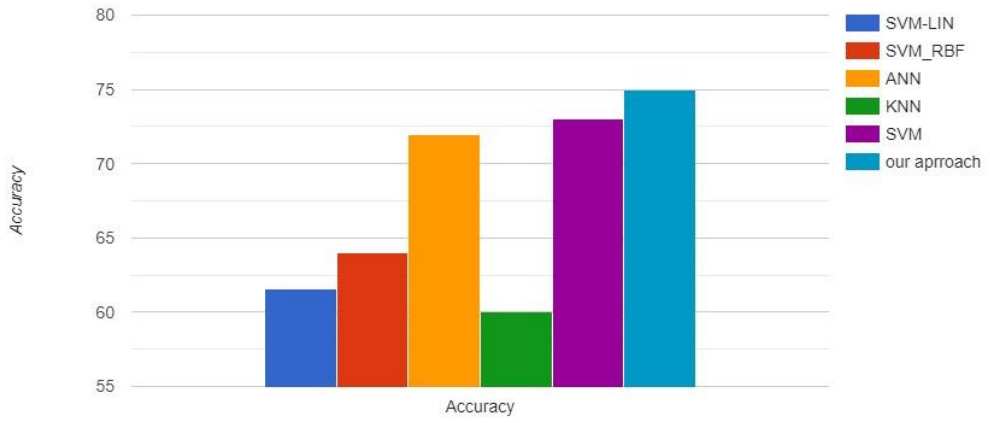



**Figure 47**

# 7 Conclusions and Future Work

## 7.1 Summary

In this paper, we propose a new recommendation system that is consists of two stages first one is classifying the face. Face shape is classified into four groups round, oval, square and heart face. The convolutional neural network method is used as a classifier.to classify the faces. Dataset of people faces that consists of 4000 images which is labelled and categorized in four classes are used. The accuracy rate is 88% for training and 75% for testing .Second stage is recommending the appropriate fitting glasses that match the user's face based on the classified face shape using GUI that is implemented in order to make the user choose best fitting glasses that match their face. Moreover, our system provides the user with two option first one is trying the glasses on their face using overlapping between two images within the boundary box of the eyes. Second option is choosing to browse frames within a specified category that the system recommends fitting the face shape.

## 7.2 Future Work

Our plan in the future is divided into parts first one will be focusing on increasing the accuracy rate.

second part improving the CNN model or trying other machine learning and deep learning models and trying to unsupervised machine learning techniques.

We also want to improve the virtual try-on glasses process by using Augmented reality to fit the face perfectly and also to be used in online shopping to make the users try the glasses before purchase it, as e-commerce is growing nowadays due to its accessibility but one of the disadvantages of the online shopping that customers can not try the products, so by working on providing them with virtual try-on technology will improve and facilitate the online shopping experience.[5]

# References


[1] Tio, A. E. (2019). Face shape classification using inception v3. arXiv preprint arXiv:1911.07916. https://arxiv.org/abs/1911.07916

[2] Sarakon, P., Charoenpong, T., & Charoensiriwath, S. (2014, November). Face shape classification from 3D human data by using SVM. In The 7th 2014 Biomedical Engineering International Conference (pp. 1-5). IEEE. https://www.researchgate.net/publication/282382114_Face_shape_classification_from_3D_human_data_by_using_SVM

[3] Shah, H. (2020). 3D-Guided Face Shape Classification. http://broncoscholar.library.cpp.edu/handle/10211.3/215338

[4] Sunhem, W., & Pasupa, K. (2016, February). An approach to face shape classification for hairstyle recommendation. In 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI) (pp. 390-394). IEEE. https://ieeexplore.ieee.org/document/7449857

[5] P. Azevedo, T. O. Dos Santos and E. De Aguiar, "An Augmented Reality Virtual Glasses Try-On System," *2016 XVIII Symposium on Virtual and Augmented Reality (SVR)*, Gramado, 2016, pp. 1-9, doi: 10.1109/SVR.2016.12. https://ieeexplore.ieee.org/document/7517246

[6] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET) (pp. 1-6). IEEE. https://www.researchgate.net/publication/319253577_Understanding_of_a_Convolutional_Neural_Network

[7] Frames Direct How to Choose the Best Glasses for Your Face0 https://www.framesdirect.com/knowledge-center/face-shape-guide

[8] D. ABELMAN (August 2018). What Is My Face Shape? https://www.allure.com/story/3-steps-to-finding-your-face-s

[9] Sanjeev (November 2010). How to Determine Your Face Shape https://makeupandbeauty.com/face-shape/

[10] Jason Brownlee (March 2016) Supervised and Unsupervised Machine Learning Algorithms
https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

[11] ISHA SALIAN (August 2018) What's the Difference Between Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning?
https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/

# Appendix 2 – User guide

The System has a very simple design in the main window once the system is opened, the user is provided with the first option which is to take a photo to make the system define their face type which round, oval, heart, or square.

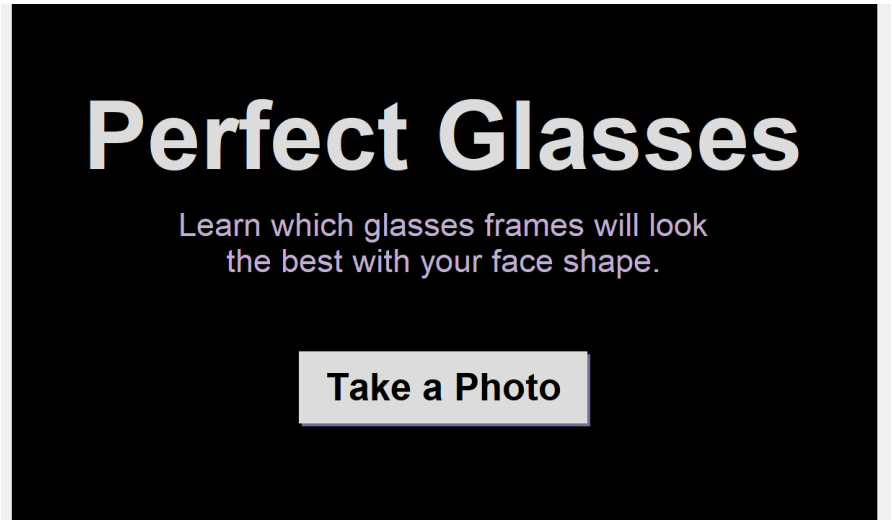



**Figure 48**

Secondly, after the user capturing their face they should drag the dots on their face like the picture to make the position of the glasses fits the face. Then the user clicks on the Done button to go to the next step.

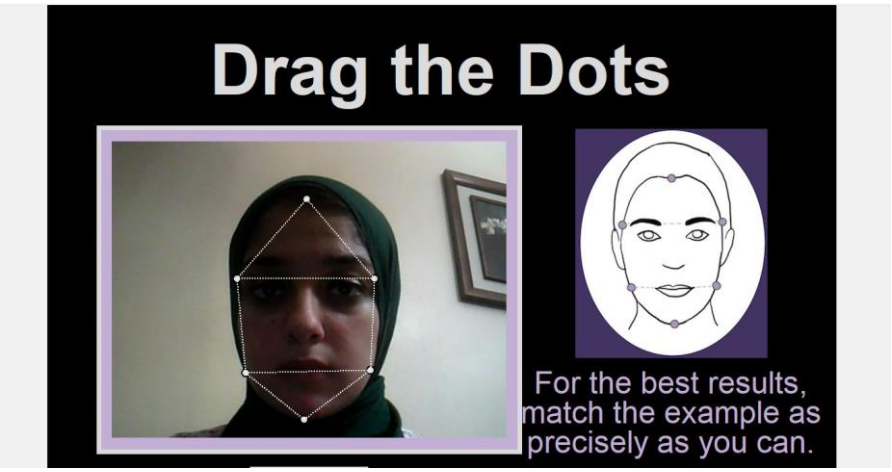



**Figure 49**

Thirdly, the system defines the user face shape. Then the user should click on “see the best style for your button” to see the recommended frames.

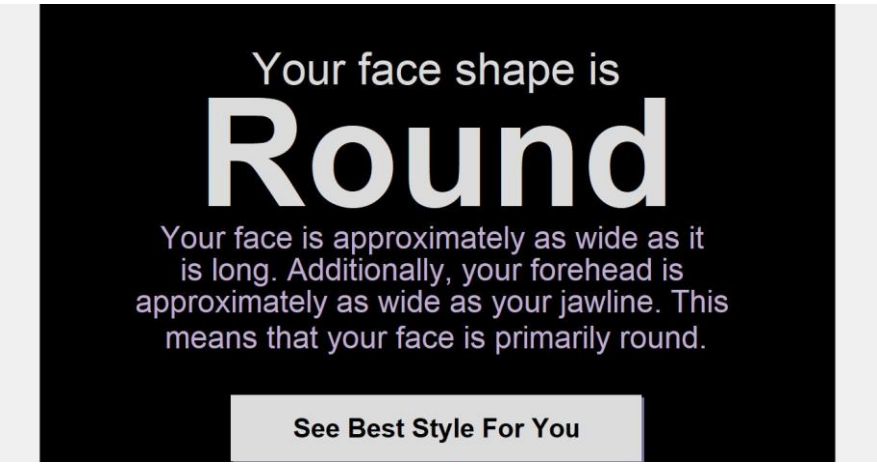



**Figure 50**

fourth, in this step, the user will be provided with two option either browse these frames button or "try them on” button which allows the user to try the glasses virtually on their face


**Figure 51**

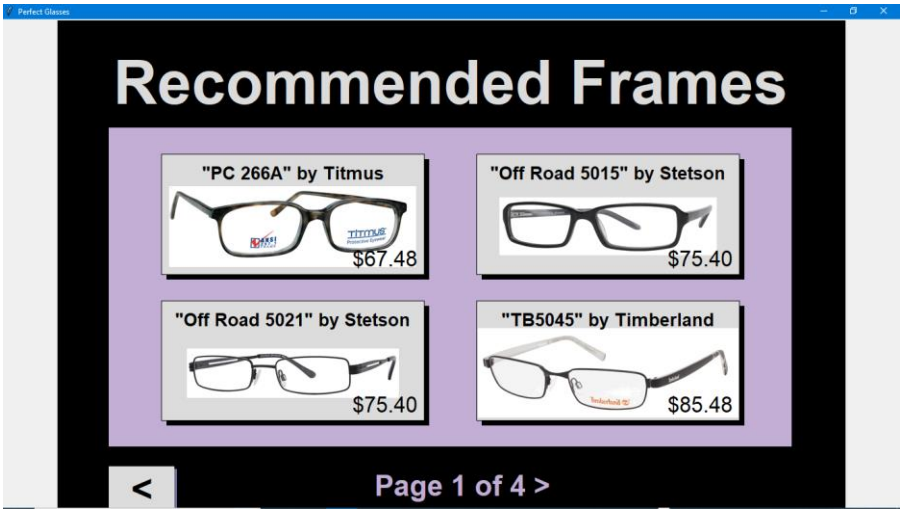



**Figure 52 example browse frames**

if the user clicks on an image of the above images, a link is opened with the details of each frame and the user can buy it through the following website.

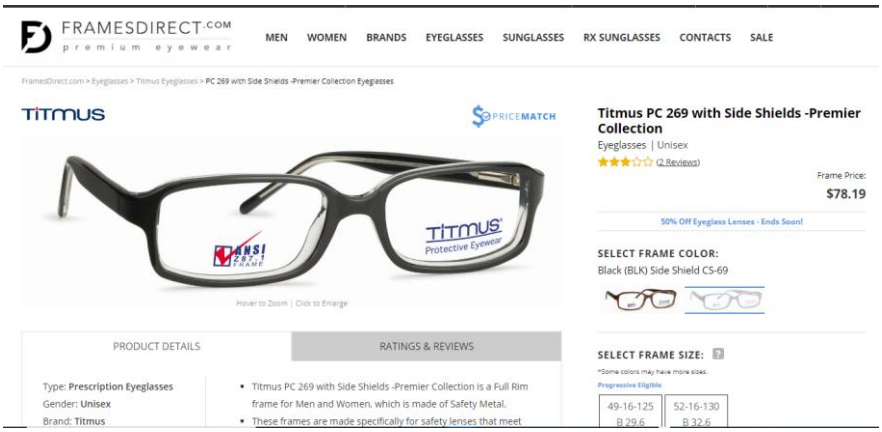



**Figure 53**

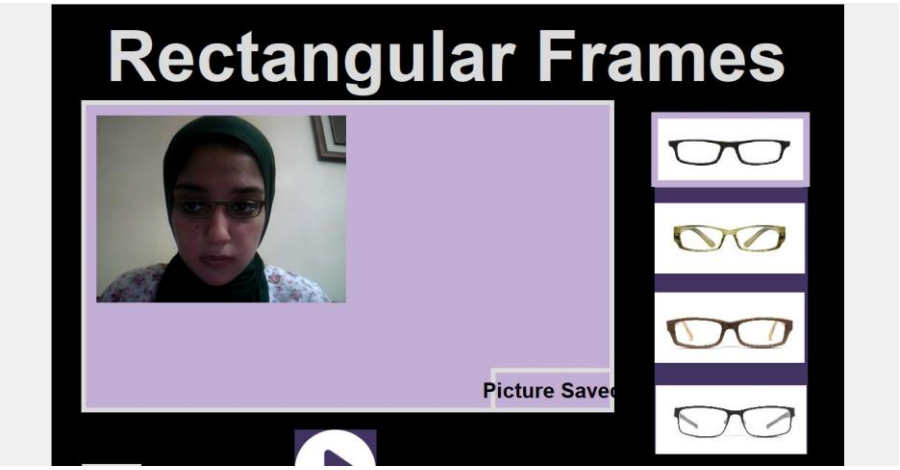



**Figure 54 Example of try-on**

# Appendix 3 – Installation guide

**Project environment**

Python version 3.7 should be installed.

TensorFlow also is needed to be installed to use Keras library.

**Anaconda**: is used to launch the environment.

**Spyder text editor** is used in this project, but any other text editor can be used.

Libraries that the user should install to build Deep learning model:

- Keras
- pandas
- NumPy
- SK-learn

The libraries that the user should install for the GUI:

- Web-browser
- MTCNN
- PIL
- T-kinter

**Command used to install libraries**

pip install “library name”