Facial Features Analysis Using Machine Learning

B.E. (CIS) PROJECT REPORT

Department of Computer and Information Systems Engineering

NED University of Engineering & Technology,
Karachi-75270

Facial Feature Analysis Using Machine Learning

B.E. (CIS) PROJECT REPORT

Project Group:

MAHEEN M.ALI	**CS-016**
SABA TANVEER	**CS-029**
SAMEER KAZI	**CS-050**

BATCH: 2016-17

Project Advisor:
Dr. Maria Waqas (Internal Advisor)

04 September 2020

Department of Computer and Information Systems Engineering

NED University of Engineering & Technology,
Karachi-75270

ABSTRACT

The most important features of aging are wrinkles, the anti-aging market as a treatment for facial wrinkles is growing rapidly. A precise image-based analysis of those different categories could play a key role in relevant aging applications. In the market, now a days, cosmetic firms produce numerous anti-wrinkle creams, and dermatological firms invest in wrinkle filler injections. While the wrinkles might easily distinguishable by the human eyes, it was a challenging task for machine learning to detect those wrinkles automatically, as there's no specific work and application in this scenario, particularly in our country. In this project, we are introducing a deep learning-based web application that detects the wrinkles on your face with the help of a convolution neural network model, the CNN model, and face recognition system which give the results in percentage based on training and comparisons. This application also compares the previous and current results and gives the information that how many percent of your wrinkles have been decreased or increased.

ACKNOWLEDGEMENTS

All praise to Almighty Allah, the Most Beneficial and Most Merciful, for His countless favors throughout this venture and particularly in moments of pressing need when His belief granted us the power to pursue successful attempts throughout this project and especially during times of dire need when his faith gave us the strength to keep making efforts to succeed.

We are highly grateful to our parents for their constant support and prayers, and our senior friends for their encouragement and guidance.

We would like to give Dr. Maria Waqas (Internal) our utmost appreciation for promoting our thoughts, for her exceptional advice and support throughout the whole project journey, and for helping us and for her extraordinary guidance through all the difficult moments.

S. No. Contents Page No.

CHAPTER 1

Introduction

This chapter introduces the research and terminology for facial wrinkles analysis. Several important terms are defined. It explains the problem statement, highlights the report contributions, and outlines the report's structure.

1.1 Background

There is a huge number of research on facial wrinkle analysis over the last few decades, driven by applications in diverse areas such as age estimation, instructional technology, marketing, beauty contests, and entertainment (Cohn and De la Torre, 2014) [1]. Face wrinkle estimation is a wide topic that has grown considerably in recent years and various methods have been proposed. In general, facial wrinkles analysis interprets facial images in terms of attributes of face such as uniqueness, expression, anxiety, age, stress, and facial diseases. These can be done in an efficient and automated way through several computer vision methods such as face recognition, facial age recognition, and face wrinkles identification. In this project, facial wrinkles are the region of interest. Face wrinkle detection labels a face with wrinkles automatically with some accuracy of the individual face. From the literature review, there are two approaches: the global approach tends to describe as a facial image as a local approach represents aging features in a series of image patches. In this work novel, methods based on this approach are developed, using a fixed template of face regions and detect wrinkle lines. To extract wrinkles accurately from a face image, two novel wrinkle detection methods are proposed, using a ridge valley pattern. In addition, a hierarchical approach of both global and regional features is developed where, performance of age prediction is further improved by combining both features. But the automatic facial wrinkles detection is an important, yet largely unsolved, challenging problem. Facial images contain great variations in expression, pose, and illumination in the uncontrolled environment. From these variations, local aging features such as wrinkles, pores, and spots are now beginning to receive increasing attention in the facial aging community (Batool and Chellappa, 2014) [3]. In our project, facial feature analysis using deep learning is considered for its simplicity, accuracy, quick result, and user-friendly.

1.2 Study Of Wrinkles

Wrinkle Face is the part of the body generally noticeable to other people, and its wrinkles are a sign of maturing. Normally, wrinkles are surface skin includes that show up in pictures as line portions as opposed to persistent lines, genuinely like those in fingerprints and palm prints. A surface line with no profundity, as a rule very fine regarding width; A wrinkle which has some profundity and frequently has more width than a surface line; wrinkle describes a spot on the face where certain wrinkles may show up. There are four fundamental kinds of wrinkles can be recognized by their histological viewpoints and pathogenesis (as appeared in Figure 1.1). Type 1 wrinkles that are (atrophic) are delicate, practically corresponding to one another, and they disappear when the skin is put under cross-over pressure. They are because of the decay of the collagen groups both in the reticular dermis and the hypodermal connective tissue strands. Where are type 2 wrinkles that are (elastotic) become logically perpetual lines, created on certain sun-uncovered skin regions where sunlight-based elastosis is hypertrophic?

Overcompensating in volume, the collagen decay. Type 3 wrinkles (expressional) become lasting lines, consistently orientated in a generalized example as indicated by the powers forced by facial muscles. Type 4 wrinkles (gravitational) result from gravitational powers prompting collapsing and listing of skin which has lost its bloat. Ramanathan and Chellappa (2008) [4] proposed a two-venture way to deal with displaying facial maturing in grown-ups. It incorporates a shape and surface, variety models. The surface variety model was intended to unequivocally portray facial wrinkles in predesignated facial districts and zones. They recognize four facial areas that will in general have a high penchant towards creating wrinkles, likely brow locale, the eye-tunnel district, the nasal district, and the lower jaw locale. At that point, they characterized the district based facial wrinkle varieties across age change into one of the three classes: refined wrinkle change, moderate wrinkle change, and high wrinkle change. This grouping is performed by learning the pixel-based varieties in angle sizes and directions. They found that the rates at which face wrinkles are showed on any people of various ages are frequently abstract. They guaranteed that their proposed surface distinction model can likewise be utilized to foresee the diverse wrinkle designs that might have been seen all over.

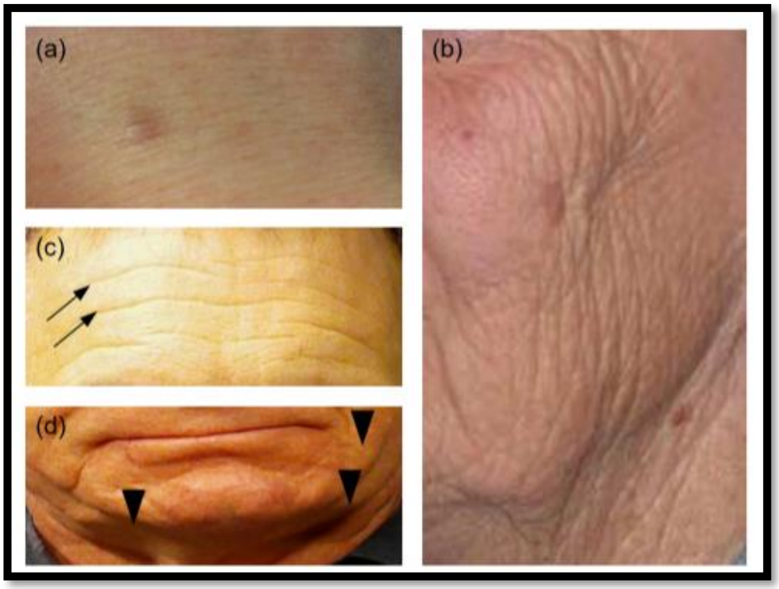


Fig 1.1- Types of Wrinkles

1.3. Motivation and Need

for the medical profession It is a useful program, especially on the facial skin lines, and for the audience who are conscious about their skin, wrinkles, age, and their home treatments. The program will use it with them to evaluate the methods to remedy of their problem by using the truth image from the program that shows the lines of wrinkles on the human face. The problem can be decreased that nowadays we lack doctors and they do not have a chance to discuss or analyze the patient's problems frequently. In China and Us, there are many researches and some work in this field, but we want to gather all this knowledge in a user-friendly mobile application with some enhancements based on requirements.

CHAPTER 2

Literature Review

In this paper, another generative model for wrinkles on maturing human faces utilizing Marked Point Processes (MPP) is proposed. Wrinkles are viewed as the thought of as irregular deliberation of arrangements of line portions and identified in an image by legitimate restriction of line sections. The power slopes are utilized to considered extra probable areas and a past probability model is utilized to oblige properties of line fragments. Wrinkles are restricted by examining MPP utilizing the Reversible Jump Markov chain Monte Carlo (RJMCMC) algorithmic guideline. We additionally present an investigation arrangement to quantify the measure of the extended model. We present outcomes on a scope of pictures acquired from the web to outline the presentation of the proposed model [5].

To identify face wrinkles by using an external feature extractor is not an easy task. In some countries, some robots measure the beauty of a person. But in Pakistan, there are no such working projects. Some researchers researched wrinkles detection and measuring face attractiveness. Wrinkles can be a sign of premature aging. Usually, wrinkles can be seen plainly in specific zones of the face and palms. Wrinkles are brought about by an absence of versatility of the skin, so the skin loses adaptability ad flexibility [6]. The most extreme impact is it can upset an individual's psychosocial perspectives which somebody will feel degrade and be overlooked by their condition. Accordingly, loads of individuals begin to treat their skin primarily for wrinkles.

A few scientists utilizing profound learning move an unblinking eye to wrinkles, age spots, and elective biomarkers that may flag the condition of soundness of an individual's skin and serving to decide the adequacy of shifted medicines. Examination group's initial work concerned getting ready 300 manually named pictures for wrinkles. They likewise ran tests utilizing a VGG organization and segnet, a profound encoder-decoder plan for pixel division. At that point, sending a NVIDIA Tesla K80 for preparing and testing gave Technical progressions in PC vision, facial acknowledgment, AI, and neural network organizations have permitted Kiselev and the group to see facial skin biomarkers utilizing a symbol of somebody's face. A 20x speed over a CPU [7].

The wrinkles recognition arrangement of Nazre Batool and Rama Chellappa examination was done by utilizing Gabor channels, picture morphology, and mathematical limitations to discover wrinkles lines on the face. The examination will watch wrinkles lines just on the temple with a profound wrinkles line, not onto the complete face.

In this paper, another procedure for assessing the age of an individual is depicted. The extended strategy passes on a solid procedure that assesses the age of a person from a gathering of face pictures with various ages. At first, most significant highlights like separations between various facial items, investigation of wrinkle zones are thought of. It is seen that the wrinkle region highlight offers the least complex outcome to appraise an individual's age contrasted with different alternatives. In this way, wrinkle zone examination is such a helpful strategy to assess the age of a person. The got outcomes are significant and momentous. Face pictures with exhibition makes an issue for legitimate eye and eyeball discovery. Pictures ought to be of a frontal view picture with uniform light on an aspect of the face. Pictures should contain a solitary face in particular and the brow should be past hair. Thus, it must be expanded more which can consider more facial highlights which will improve the exactness matured assessment. This examination work can be utilized for the expectation of future appearances [8].

A few examinations were directed to distinguish wrinkle consequently. Batool and Chellappa in their investigation accepted about the mathematical qualities and wrinkles highlights. They said wrinkles show up in the curves structure, constant, and not converging with one another [9].

The other examination is led by Nur Hayatin, which characterizes age dependent on facial pictures utilizing significant highlights especially face human-centric and wrinkles [10]. The wrinkle includes that are determined are the width and profundity of wrinkles. The exactness of that framework is 65th, whereas this examination didn't show the wrinkles highlights acquired.

As indicated by the past examinations, wrinkles lines in the facial picture are frequently recognized by using picture measure methods. Subsequently, this investigation attempts to frame a model that may recognize the presence of wrinkles through facial pictures consequently without the prerequisite to utilize uncommon instruments claimed uniquely by excellence facilities. This investigation expects to identify the area of wrinkles lines on the face picture. The framework is foreseen to help clients to locate wrinkle lines on their countenances.

In addition, this procedure might be applied as one of the highlights of excellence item skin guides. Another exploration paper was distributed during which inside which the reason for the examination is to style a framework that would distinguish the exact area of the line of wrinkles that happen from one face picture [11].

2.1 Real World Applications

As humans age, numerous changes occur chronically within the craniofacial complex. Notable soft tissue modifications may be seen across each decade of adult life that by-passes. As well, delicate hard tissue or bony changes slightly alter the overall shape of the human face, mainly in the dentoalveolar region (a portion of the alveolar bone instantly about teeth). These age-related changes affect the accuracy and effectiveness of face-related applications. Characterizing the progressive, but delicate changes in facial appearances such as facial wrinkle have many significant implications.

2.2 Human Perception

Changes in skin disorders throughout a person's life are indicators of the irreversible aging processes that occur in the human body. The texture and appearance of skin slowly shift from a smooth, translucent, and uniformity youth to a rougher and more haphazardly pigmented skin. Facial wrinkling, sagging, and pore enlarging, all generally irreversible without the interference, mark the enactment of time or, perhaps more precisely, the accumulation of photo-damage. Though these aging phenomena are important social and self-respect determinants, only recently have the biochemical and physiological changes associated with them begun to be investigated.

Mark et al. Investigated the changes in the shape of a human head and the development of facial wrinkles as potential sources of information regarding the age level. In the first experiment, participants approximate the ages of faces that had been produced by systematically influencing characteristic head shapes and levels of wrinkles associated with ages 15, 30, 50, and 70 years [14]. The results indicated that participants used both causes of craniofacial change in making age estimates, but the effects of any source of change on perceived age depended upon the level of the other source of change. In the second experiment, a participant in the ratings of apparent conflict between levels of head shape and wrinkles further validated the conclusion that the participants are susceptible to the coordination of products of the two sources of change. These findings suggest that the information specifying the perceived age level is a complex relationship among different types of craniofacial transformation.

2.3 Wrinkle Detection Techniques

Even Though the relationship between age and physical and physiology skin characteristics like wrinkles has been studied for the last century, there was no attempt prior to the advent of image processing to accurately measure visual characteristics of skin aging [15]. The apparent signs of damage to the skin, such as wrinkling, sagging, irregular pigmentation, telangiectasia, and hyperkeratosis are often clinically assessed using either descriptive scales or photographically standardized ones. The well-known flaws of subjective assessment limit the scientific study of treatment and environmental impacts on skin aging. Because of facial wrinkle is very similar to the blood vessel in the retinal image, different segmentation methods in facial and retinal images are evaluated here. [15] showed the use of image processing techniques to the study of profilometric traces of skin models. It is to isolate specific wrinkles and measure their depth. There are four steps in the processing: concealing, shading correction, enhancement and division, and object identification and morphological measurement. They argued that the proposed method is a complement to the measurement of wrinkle width and length as taken from the conventional optical images. In comparing the image-analysis measuring of subjects' pictures to their perceived ages, a high correlation of perceived-age with the overall length of wrinkles. Thus, they stated that the visual perception technique should be helpful for the validation of age-related measurements and for evaluation of medical treatments which change the facial appearance. They also demonstrated that the suggested technique able to differentiate the old from the young and have sufficient sensitivity to detect small skin aging associated effects. [16] built an algorithm for the automatic detection of facial wrinkles. It is based on estimating the professional development and the frequency of extended spatial features, capture via digital image filtering. They use the different idea of finger print detection where noise is removed, and the contrast of the image is enhanced to capture the irregularities of the linear surface. However, the correctness of wrinkle detection is not validated where the found wrinkles might be noises or lines of interest.

They pointed out a few image traits of wrinkles as follows.

- Wrinkles are causing power angles which will be featured by Gabor filter bank.
- Wrinkles show up as curvilinear highlights as opposed to mass like highlights.
- Wrinkles are persistent and it's anything but a progression of separated curvilinear sections.
- Wrinkles are less probable to intersect with one another or to be clogged in little regions.

Experiments are performed on two sets of low and high-resolution images and the results indicated that the proposed algorithm is significantly a lot quicker than the conventional approach but also provides visually improved results. However, their work is the absence of intra- and inter reliability measurements in which the annotation of a single coder can be biased. In addition, their method is a time consuming due to a maximum of 30 iterations is necessary to verify the neighboring sites of connected sectors. There are two limitations to the assessment setup. This established subjectivity in the quantitative results computed from the experiments. On the other hand, they also emphasized the factors that contributed to the degradation of localization results as follows.

2.3.1 Very light wrinkles.

The most well-known reason for low discovery rates was wrinkled' being extremely light. The quality angles brought about by the granular skin surface resemble those brought about by wrinkles. Thusly, the calculation picks non-wrinkle locales as wrinkles which developing bogus alert rate.

2.3.2 Skin discoloration

This issue was more observable in subjects of lighter skin shading which is more inclined to getting moles and earthy colored/dim spots.

2.3.3 Illumination and bright spots

Because of the freely accessible pictures being under unreasonable obtaining settings, a large portion of the pictures had some splendor variety. The variety caused 'brilliant spots' in pictures where light is reflected from the skin surface because of oily skin

2.4 Our Performance for Wrinkle Detection

Maturing is a characteristic phenomena that includes various pieces of the human body by the influence of different biological and environmental variables. In light of the writing survey, people depend on numerous facial highlights like wrinkles to anticipate the face age, i.e., face shape, hair, mustache, whiskers, mole, scar, and wrinkle. The most observable changes that occur on the face is the presence of wrinkles, that are the focal point of this exploration. Exact wrinkle location naturally is a basic errand in face investigation. Some have been proposed in the writing, however, the helpless restriction restricts the execution of wrinkle recognition. It will bring about bogus wrinkle identification and accordingly influence the cycles, for example, age assessment and clinician score assessment. Prior to building up a powerful calculation, it is essential to acknowledge how people perform wrinkle discovery. In this part, we three members explain the percentage of wrinkles of human face pictures by trained model. A dataset of five classes was made for this module where 25000 pictures were arbitrarily chosen.

CHAPTER 3

SOFTWARE, TOOLS AND PLATFORM

Following are the tools that we have used in making this project work:

3.1 Colaboratory (Colab):

It is one among the simplest tools used for coding. it's often used for performing a large set of tasks like Machine learning, Deep learning applications with the assistance of PyTorch, TensorFlow or OpenCV. Colab notebooks allow an access to mix both rich text and executable code in a single document, in conjunction with images, HTML, LaTeX and lots of more. The way it works is that whenever you create Colab notebooks, they're instantly stored in your Google Drive account. It also allows you to share your Colab notebooks with friends or co-workers, also allowing them to comment on them or oven edit them.

3.1.1 Data Science:

By using Colab, we can be able to harness the full power of Python libraries in order to visualize and analyze data. After that, the data can be imported into Colab notebook from Google Drive account, from spreadsheets, Github and many other sources.

3.1.2 Machine Learning:

Another application of Colab is importing an image data set, train an image classifier thereon and evaluate the model. All of this may easily be exhausted just a few of lines of code. Colab notebooks execute code on cloud server of Google, which allows you to leverage the power of Google hardware regardless of the power of your machine, which includes GPUs and TPUs. Within the machine learning, Colab has numerous applications like working with TensorFlow, Development of neural network and its training, do experiments with TPUs and the Creation of tutorials.

For free computational resources, Colab requires to care of the pliability so on regulate the limitations of usage and hardware availability. In Colab, different resources are available.

Some users wish to be ready to do additional in Colab than the resource limit allow. It has been ascertained from the users who wish quicker GPUs, longer running notebooks and a lot of memory, additionally as usage limits that are higher and don't fluctuate the utmost amount. Introducing Colab professional is that the initiative towards serving users who want to undertake to to more in Colab. The long goal is to continue providing a free version of Colab, whereas also growing throughout a property fashion to satisfy the needs of the users. Different types of GPUs are available in Colab. Colab provides free access to some resources. Nvidia K80s, T4s, P4s and P100s are available in Colab. There is no possible because of choose what kind of GPU we'll attach with in Colab at any given instant. Whereas, users who are interested in additional access to Colab's fastest GPUs might be interested by Colab Pro

In Colab virtual machines, the amount of memory accessible varies over time. The resources available in Colab are actually prioritized for users who have recently used less resources. so, on stop the monopolization of limited resources by alittle number of users. so, on urge the foremost out of Colab, one should consider closing the tabs when he's through together with his work.

3.2 Jupyter Notebook:

The IPython notebook is additionally referred to as the Jupyter Notebook. Using Notebooks is now a significant part of the data science workflow at companies across the world. Therefore, if your goal is to figure with data then employing a Notebook will speed up your workflow and make it easier to speak and also share your results. It is an interactive computational surrounding in which you can execute code without problems, wealthy text, mathematics, plots and wealthy media. With JupyterLab, you may have an interactive development surroundings for Jupyter notebooks, code and data.

There are various features of Jupyter Notebook:

3.2.1 Pluggable authentication:

It offers the control of users and authentication with PAM, OAuth or combine along your very own listing carrier system.

3.2.2 Centralized deployment:

In your organization, deployation the Jupyter Notebook to lots of customers on centralized infrastructure on- or off-site.

3.2.3 Container Friendly:

To scale your deployment use Docker and Kubernetes, isolate consumer techniques and simplify software program installation

3.2.4 Code Meets Data:

Deploy the code nest in your information to offer unified software program control and information get admission to inside your organization.

Jupyter Notebooks are an open file layout supported JSON. It maintains the whole records of the user’s sessions. It additionally consists of code, narrative text, equations and wealthy output. It builds communique with computational Kernels the usage of the Interactive Computing Protocol. Moreover, it's possible to use many various programming languages in Jupyter Notebooks. the simplest way for a beginner to urge started with Jupyter Notebook is by installing Anaconda. Anaconda is that the most generally used Python distribution for data science and comes pre-loaded with all the foremost popular libraries and tools. On windows, you'll run Jupyter via the shortcut Anaconda adds to your start menu, which can open a replacement tab in your default browser. The facility of using Jupyter Notebooks should even be evident.

3.3 Django:

It is a high-degree Python Web framework. It stimulates the fast growth. It has been constructed with the aid of using skilled developers, it looks after a good deal of the trouble of Web development, so that you can extra cognizance on writing your software without having to reinvent the wheel.

The important features of Django are as follows:

3.3.1 Very Fast:

The main reason behind the development of Django is to help the developers. It does its work from beginning to ending as soon as possible.

3.3.2 Secure:

It used to help the developers in regards of security issues and assist them to correct the basic mistakes of security such as SQL injection, cross-site scripting, cross-site forgery and clickjacking. It helps to manage users' personal information securely.

3.3.3 Fully Loaded:

Many essential tasks are managed by Django such as authentication, administration, site maps and many more other tasks.

3.4 HTML:

HTML is a necessary and basic tool of web development. If you are a frond end developer then you will surely know the importance of HTML in your project. Following are five highly rated HTML editor tools:

3.4.1 CoffeeCup HTML Editor:

It gives a chance to use HTML editor for free. It also provides a visual editor for an additional cost. It might be used to create both CSS and HTML contents and to edit already developed websites content.

3.4.2 Komodo Edit:

It gives a chance to use multilanguage HTML editor for free. This HTML editor consists of some powerful attributes. It can also be used for editing many other coding languages such as “Python”, “Perl” and “Node.js”.

3.4.3 Apache NetBeans:

If you are working with “HTML5”, “PHP”, “JavaScript”, “C++” and several other coding languages, you can easily use Apache NetBeans. Many different code templates, generators and project management tools are offered by the Apache NetBeans.

3.4.4 Notepad++:

It offers the entirely feasible user interface and the pattern of syntax can be decided by the user.

3.4.5 Visual Studio Code:

Visual Studio Code provides platform to use any languages for coding. It offers many different tools which are necessary for coding. A lot of technical person recommends Visual Studio Codes because it has many features and it is easy for use as well.

3.5 CSS:

CSS proves to be the most basic and essential technology for working in web development. It makes the tooling tamed and enjoyable as well. Top five CSS tools for web developers and designers are as follows:

3.5.1 PostCSS:

It enfolds several powerful packages and attributes together. It provides satisfied results in simple workflow it deals with CSS.

3.5.2 PurgeCSS:

It helps to avoid bundled of unuse CSS within the output and from addition of it to workflow as well.

3.5.3 Tailwind:

It is a CSS framework, but it is different from the workflow of normal routine..

3.5.4 Stylistically Awesome Style Sheets (SASS):

It works as a superset of “CSS”. It offers modular design, include files, mix ins, inheritance etc.

3.6 JavaScript:

It is one of the most widely used scripting languages by the developers. It is quite powerful, resourceful and intuitive which comes into action when we require our website to be a little different. The true quality of JavaScript is not the language itself, but the ecosystem of thousands of tools and libraries developed by resourceful developers around the world. Following are some tools including frameworks, debuggers, text editors that will help you code efficiently using JavaScript.

3.6.1 Sublime Text:

It is a cross-platform text editor with Python API. It supports multiple languages and also markup languages.

3.6.2 Gulp:

It is a task automation tool. The file that is being created in this tool, is a plain JavaScript file that you can run to automate your tasks. It is quite developer friendly and easy to learn

3.6.3 NPM:

It is an acronym for “NodeJs Package Manager”. For publishing projects which are developed using NodeJS, it works as an online repository.

3.6.4 Webpack:

It helps you in loading your modules with some dependencies like .js, .css, .html etc. It works as a bundler which bundles your similar kind of files into a single precise file.

3.6.5 Eslint:

It is a tool which is used for finding errors. It is also a JavaScript linting tool, it will help you in programming errors and it’s a must have for JavaScript testers

3.6.6 Jasmine:

It is a test suite which comes into action when JavaScript is installed into browser. It is an automated behavior-based test suite

3.6.7 Mocha:

It is a simple and flexible testing framework based on the JavaScript. It further offers asynchronous testing and any assertion library can be used with it.

3.7 Bootstrap:

For creating effective and attractive websites, Bootstrap prove itself an amazing tool. It also provides connection to components of web development platform in a single place. All browsers support Bootstrap. It also develops a responsive web. Following are the available tools of Bootstrap:

- Flatstrap
- Bootstrap Designer
- jQuery Mobile Bootstrap
- Bootply
- Bootstrap Magic

3.8 Database:

If you want to store your data in an organized manner with security then you should know what is database? It is a secure place which keeps information in a manageable way. we are able to do different operations on the information within seconds. Database is the essential part of any organization's IT department. Database tools might be a wide term that covers all the tools, utilities and assistants that are used to perform various "database administration tasks". Every tools of Database perform a particular task and not all the tasks which are specify for admins. within the most typical situation, whereas checking out the correct database tool as a result of you have got encountered a particular downside or scenario and need to remedy it. Its additionally a possibility that you simply predict the prevalence of such problems and need to look into your database to ignore it.

Another vital side to think about is that the infrastructure of “database management system” which would vary between different companies, fields and projects too. To find the correct tools and to utilize your efforts you should know the structure of “database” and the issues associate with it. Many times, you may need database tools which will handle a particular practicality like examination the info across 2 or additional databases. Many tools do not go well with many operations. So, you should know the compatibility between any certain tool and “software system”. In fact, integration of third party tools might not be simple and is need further effort.

3.8.1 Firebase:

It helps user to develop a good quality application and grow user base. It consists of many different attributes that you can mix and match to fit your needs. It helps to save your time, reduce the developer workload more than 50% for even simple projects. It provides fully integrated services. You can bootstrap your initial prototype using Firebase on Google-scale infrastructure.

Firebase tools deploys five different modules, each of which can be deployed individually:

- Hosting
- Database
- Firestore
- Storage
- Functions

CHAPTER 4

Methodology & System Design

Our project is based on Deep learning's algorithm named as Convolutional Neural Network (CNN) and database is maintained using firebase. User interface is developed using Django so that the user can upload images easily on the system. In this chapter, we will describe our backend model briefly.

4.1 Convolutional Neural Network

To develop the model for the classification of images we used Convolutional Neural Network which is the multi-layer Neural Network that includes input layer, convolutional layers, sample layers and output layer. It is also known as fully connected network which means each and every neurons of a previous layers are connected to each and every neurons of next layers.[1]

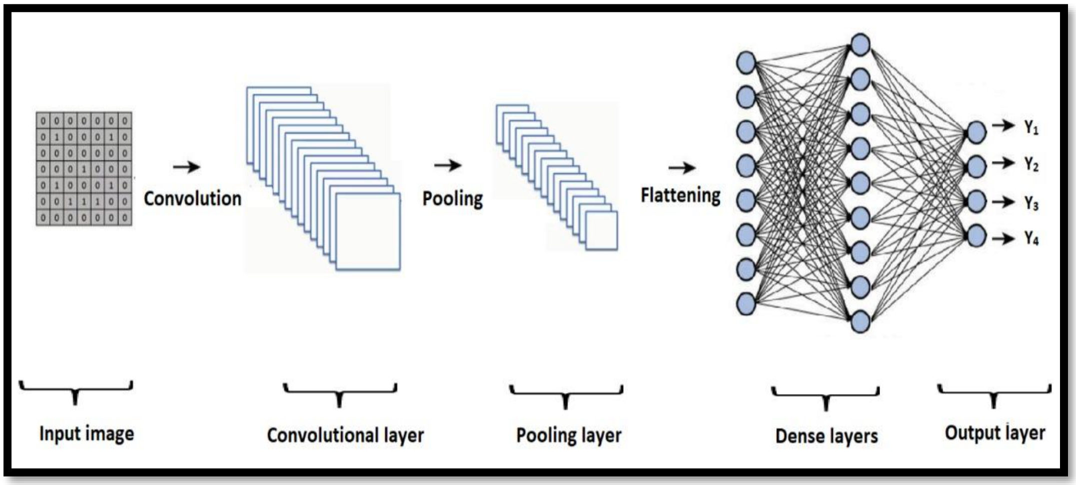


Fig. 4.1: CNN Architecture

4.1.1 Convolutional Layer

This layer used to extract features from the image and apply different functions on the image like sharpening, detection of edge, reduction of noise and many others functions.[17]

4.1.2 Pooling layer

This layer is used to turn down the dimensions and parameters of the image. It helps to lessen the cost of computation.

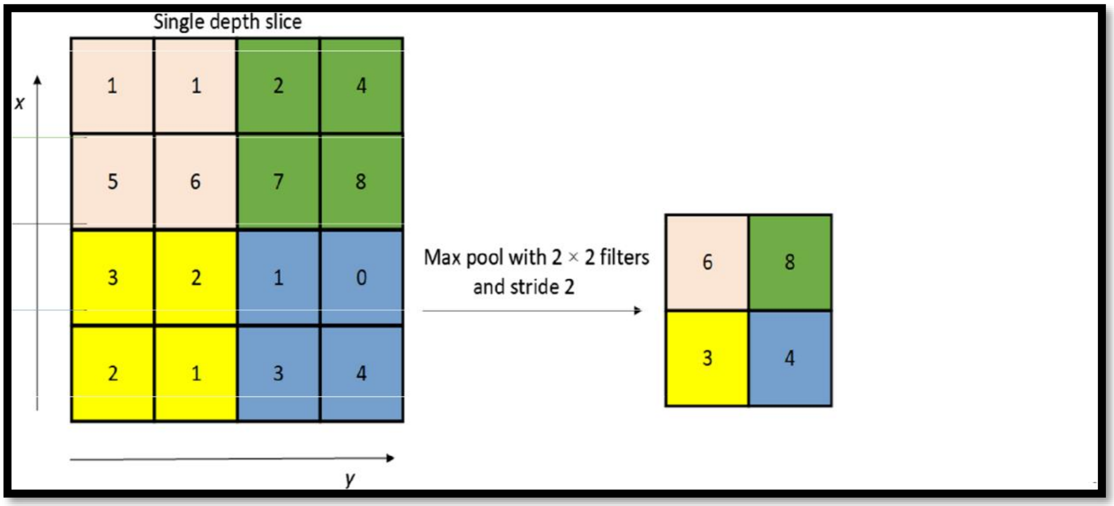


Fig. 4.2: Pooling operation

4.1.3 Dense layer

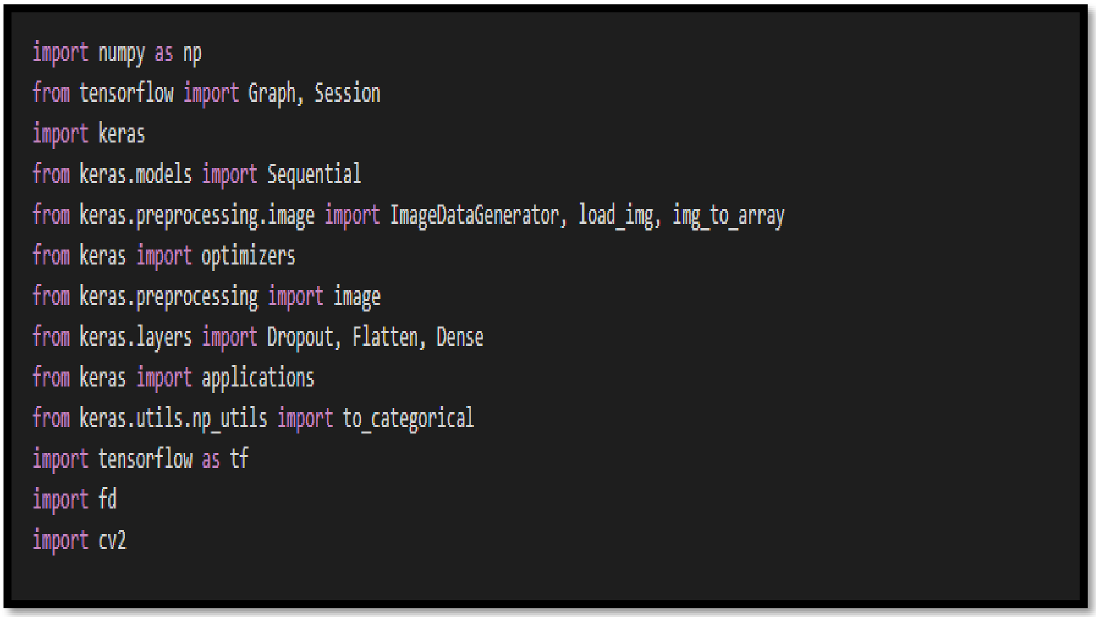
This layer is also called fully connected layer. Inputs are taken by every neurons of previous layer and send to the neurons of next layer.

4.2 Gathering and classification of data

First, we gathered data from different data sets and websites and then classified them with respect to the intensity of wrinkles and labelled data set. Secondly, we developed a convolutional Neural network model and trained our data set.

We have divided our data set into three parts. First is training, second is validation and third is testing data set. 85% to 90% of total dataset is referred to as training data set. This dataset is used to train our model. 5% to 10% of total dataset is referred to as validation dataset. This dataset is used to determine the performance of our model. The remaining part of total dataset is referred to as testing data which must be unlabelled. This dataset is used to for testing the model.

4.3 Importing libraries

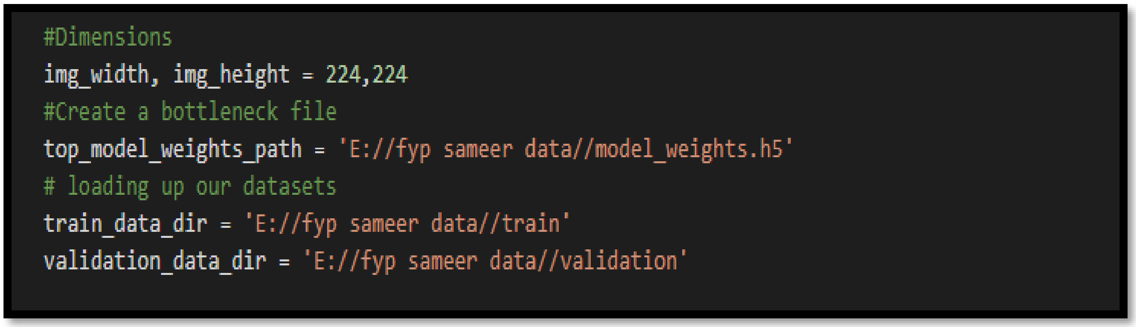


```
import numpy as np
from tensorflow import Graph, Session
import keras
from keras.models import Sequential
from keras.preprocessing.image import ImageDataGenerator, load_img, img_to_array
from keras import optimizers
from keras.preprocessing import image
from keras.layers import Dropout, Flatten, Dense
from keras import applications
from keras.utils.np_utils import to_categorical
import tensorflow as tf
import fd
import cv2
```

Fig. 4.3: libraries

4.4 Defining dimensions

After importing libraries, we defined the dimensions and location as shown in fig. 4.4:

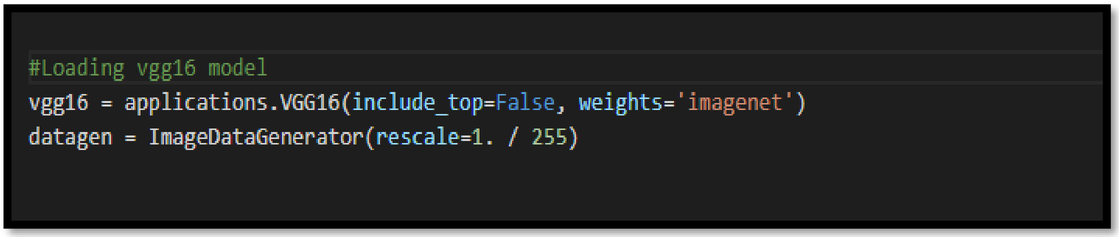


```
#Dimensions
img_width, img_height = 224,224
#Create a bottleneck file
top_model_weights_path = 'E://fyp sameer data//model_weights.h5'
# loading up our datasets
train_data_dir = 'E://fyp sameer data//train'
validation_data_dir = 'E://fyp sameer data//validation'
```

Fig. 4.4 Dimensions

We have set the dimensions of images 224x224. The bottleneck file system is also developed by which the image pixels is converted into the NumPy array which is then store in the given location. Also determine the address where validation data and training data is in our system.

4.5 VGG16 transfer learning model



```
#Loading vgg16 model
vgg16 = applications.VGG16(include_top=False, weights='imagenet')
datagen = ImageDataGenerator(rescale=1. / 255)
```

Fig. 4.5 Importing Vgg16 Model

In fig 4.4, we imported a transfer learning model VGG16.

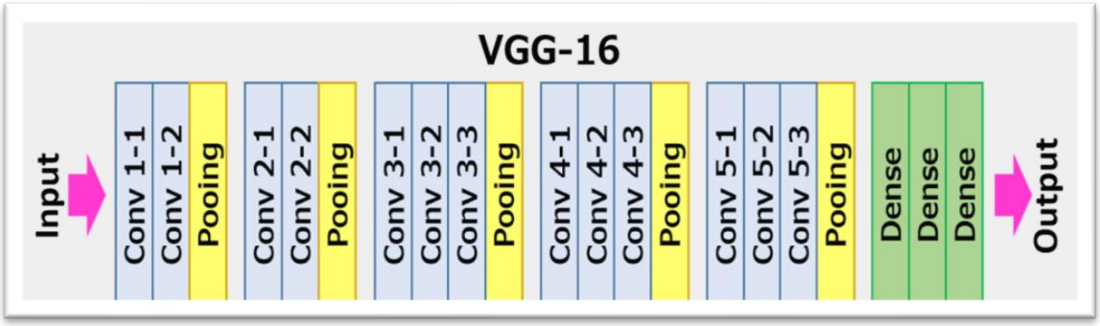
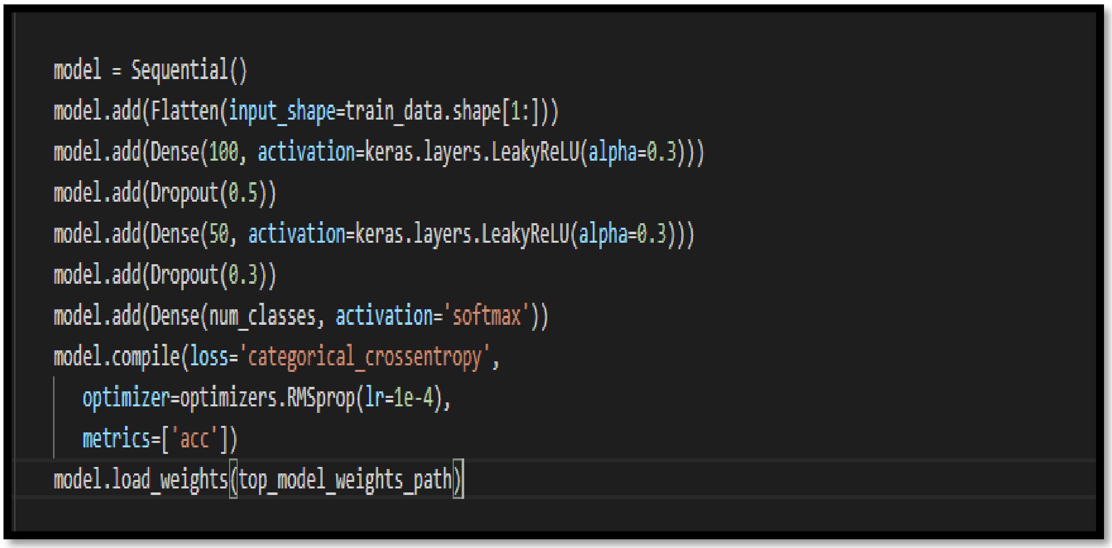


Fig. 4.6 Importing Vgg16 Model

It is a convolutional neural network architecture which is pre-trained on 16 convolutional layers. Due to VGG16, the efficiency and accuracy of model has been increased. Image is passed through the input layer and then transfer to the many convolutional layers as described in fig. 4.5. In the end, there are three dense layers which is also called fully connected layers.[18]

4.6 Development of Neural Network:

In fig 4.6, we added the upper layers to the model. At this stage, our model has been ready for training. First, we set our model as Sequential, then apply flatten layer on our data. After this, we have developed some hidden layers. We used ‘SoftMax’ as an activation function. We have trained our model on Colab and when the bottleneck files and weight files have been created and saved on the system, we loaded these files on Jupiter and then compiled the model.



```
model = Sequential()
model.add(Flatten(input_shape=train_data.shape[1:]))
model.add(Dense(100, activation=keras.layers.LeakyReLU(alpha=0.3)))
model.add(Dropout(0.5))
model.add(Dense(50, activation=keras.layers.LeakyReLU(alpha=0.3)))
model.add(Dropout(0.3))
model.add(Dense(num_classes, activation='softmax'))
model.compile(loss='categorical_crossentropy',
   optimizer=optimizers.RMSprop(lr=1e-4),
   metrics=['acc'])
model.load_weights(top_model_weights_path)
```

Fig. 4.7 Remaining Layers

4.6.1 Sequential model:

It means model is sequential. Sequential model is the linear collection of layers of convolutional neural network.

4.6.2 Flatten layer:

In this layer, data will be flattened means all the dimensions are removed and only one dimension will be remained. Two-dimension array will be converted into one-dimension array.

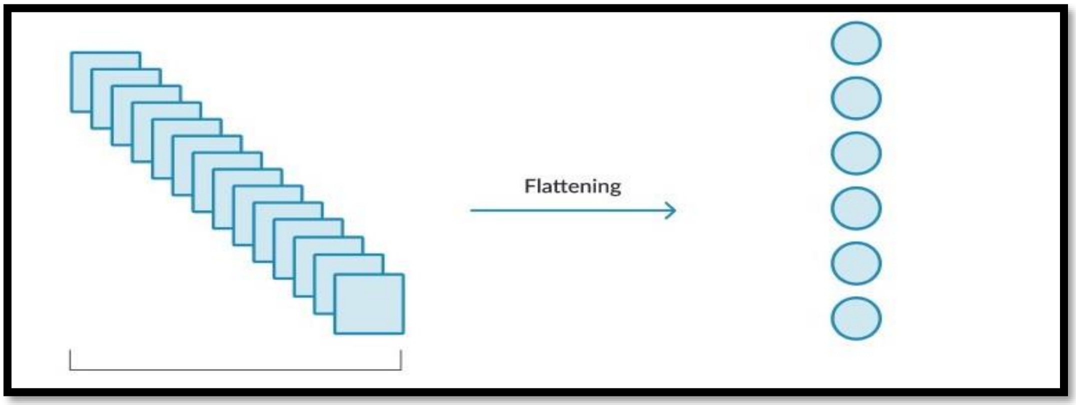


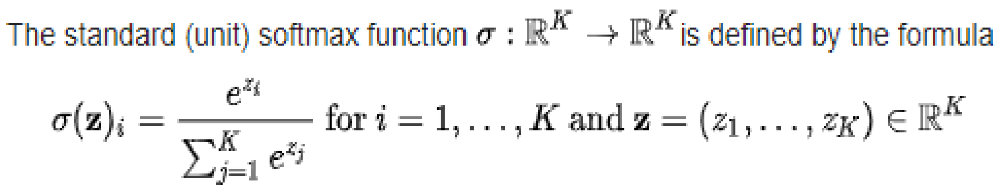
Fig. 4.8: flattening operation

4.6.3 Dropout:

In this layer, some units are dropped with some probability and the remaining units are used for training. Units are dropped out randomly. We need this layer to avoid ‘overfitting’ and for regularization of network.

4.6.4 Activation Function:

We used the ‘SoftMax’ activation function which transforms the values of vector into the values between 0 and 1 so that these values can be treated as probabilities. It works good for categorical cross entropy loss as well.

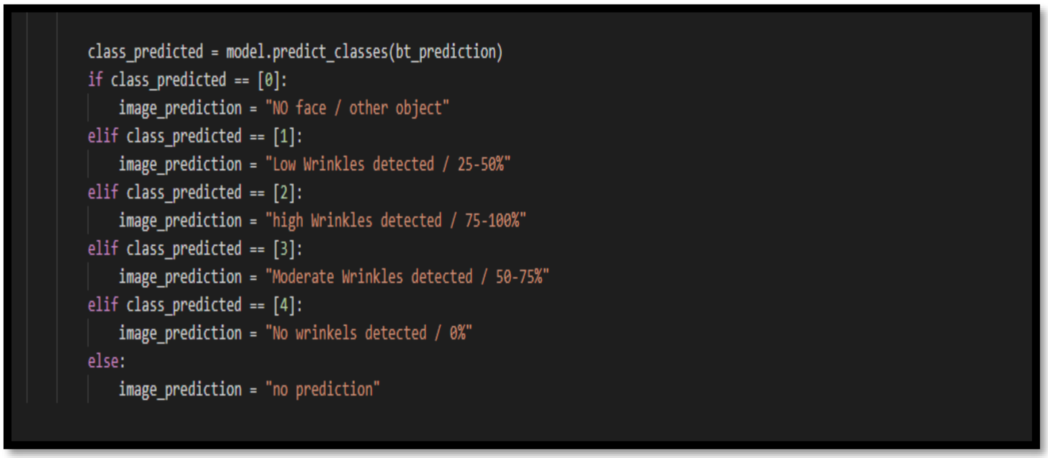


The standard (unit) softmax function $\sigma : \mathbb{R}^K \to \mathbb{R}^K$ is defined by the formula

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{ for } i = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K$$

Fig. 4.9: formula for SoftMax function

4.7 Classes:



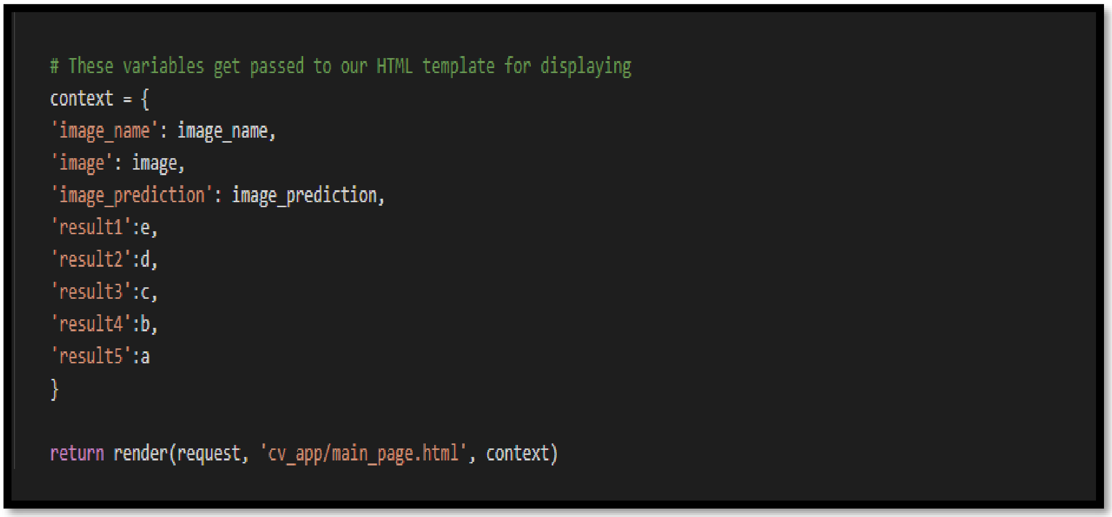
```
class_predicted = model.predict_classes(bt_prediction)
if class_predicted == [0]:
    image_prediction = "NO face / other object"
elif class_predicted == [1]:
    image_prediction = "Low Wrinkles detected / 25-50%"
elif class_predicted == [2]:
    image_prediction = "high Wrinkles detected / 75-100%"
elif class_predicted == [3]:
    image_prediction = "Moderate Wrinkles detected / 50-75%"
elif class_predicted == [4]:
    image_prediction = "No wrinkels detected / 0%"
else:
    image_prediction = "no prediction"
```

Fig. 4.10 Classification

In fig. 4.8, these are the classes of our data set on which our model has been trained. When we give the image to the system then our model processes the image and predict the result according to the given classes. Our project also has face detection mechanism. When the image is uploaded by the use to test the system first our project detects the face in the image, If the image has no face then our model places the image in first class ‘No face’. Next, if the image has a face but it is wrinkles free then our model places the image into the last class ‘No wrinkles’. If the image has a face and has some wrinkles on face, then our model places this image on ‘low wrinkles’ class. If the image has a face and has some moderate wrinkles, then the model places this image in ‘Moderate wrinkles’ class. At the last, if the image has a face and also have high percent of wrinkles then our model places the image into ‘High wrinkles’ class.

4.8 Displaying result:

When the model predicts the result of the test image then the result along with the image and image name will be saved in variables and send back to the HTML page. Hence, the result will then be shown on web page.



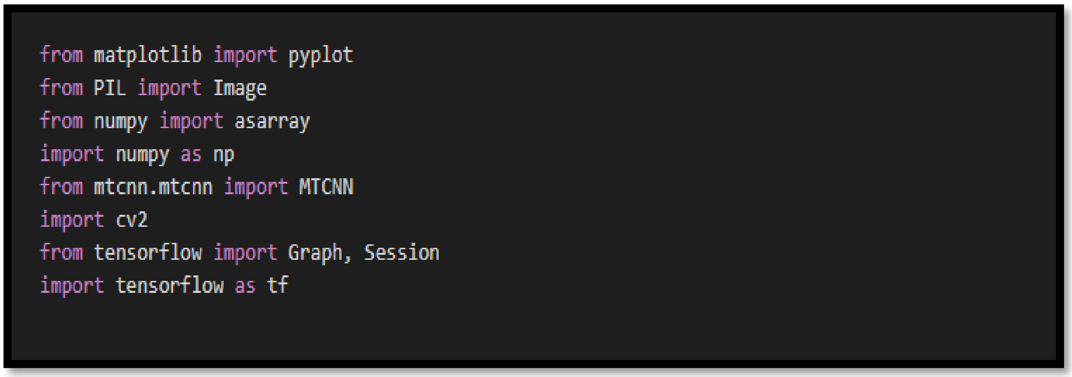
```
# These variables get passed to our HTML template for displaying
context = {
'image_name': image_name,
'image': image,
'image_prediction': image_prediction,
'result1':e,
'result2':d,
'result3':c,
'result4':b,
'result5':a
}

return render(request, 'cv_app/main_page.html', context)
```

Fig. 4.11 Display Result

4.9 Face detection mechanism:

When the user uploads an image on the system then first this image must go through the face detection process. If the image has no face then it will be placed on the ‘no face’ class and if the face is detected in the image then the image will be passed to the model in order to predict the result. We have used Multi-Task Cascaded Neural Network (MTCNN) to detect the faces and facial landmarks in the image.[19]

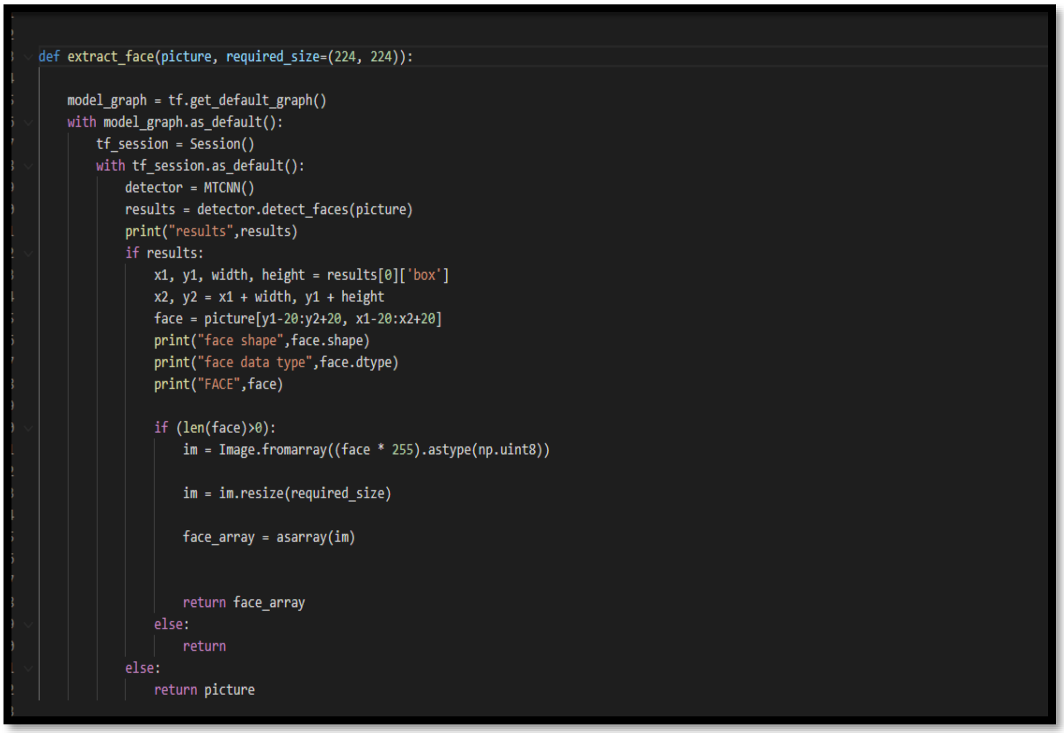
In fig. 4.10, all the necessary libraries are mentioned which are used in face detection algorithm:



```
from matplotlib import pyplot
from PIL import Image
from numpy import asarray
import numpy as np
from mtcnn.mtcnn import MTCNN
import cv2
from tensorflow import Graph, Session
import tensorflow as tf
```

Fig. 4.12 Libraries For MTCNN

MTCNN is basically a pre-trained neural network. In fig 4.13, coding is shown:

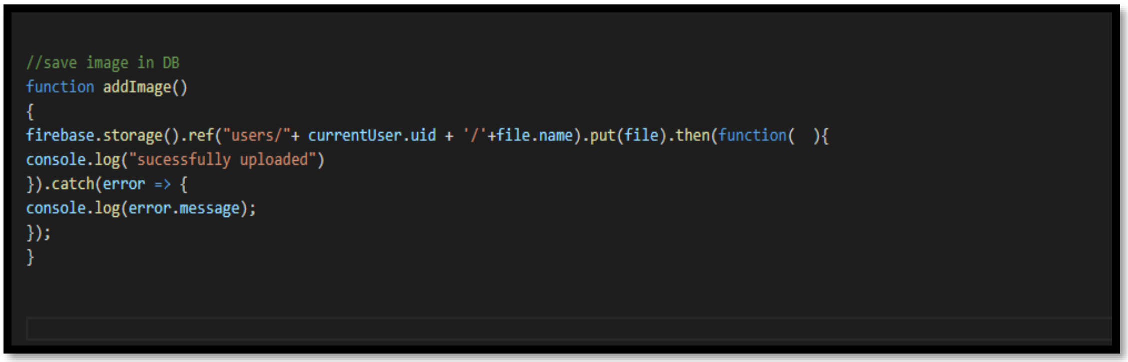


```
def extract_face(picture, required_size=(224, 224)):

    model_graph = tf.get_default_graph()
    with model_graph.as_default():
        tf_session = Session()
        with tf_session.as_default():
            detector = MTCNN()
            results = detector.detect_faces(picture)
            print("results",results)
            if results:
                x1, y1, width, height = results[0]['box']
                x2, y2 = x1 + width, y1 + height
                face = picture[y1-20:y2+20, x1-20:x2+20]
                print("face shape",face.shape)
                print("face data type",face.dtype)
                print("FACE",face)

                if (len(face)>0):
                    im = Image.fromarray((face * 255).astype(np.uint8))

                    im = im.resize(required_size)

                    face_array = asarray(im)


                    return face_array
                else:
                    return
            else:
                return picture
```

Fig. 4.13 MTCNN

4.10 Saving image in database:

Since, we are using firebase database, so the image is also saving in the firebase database against respective user Ids. In this way, all the history of respective users is maintained in the database which can be accessed by the users.

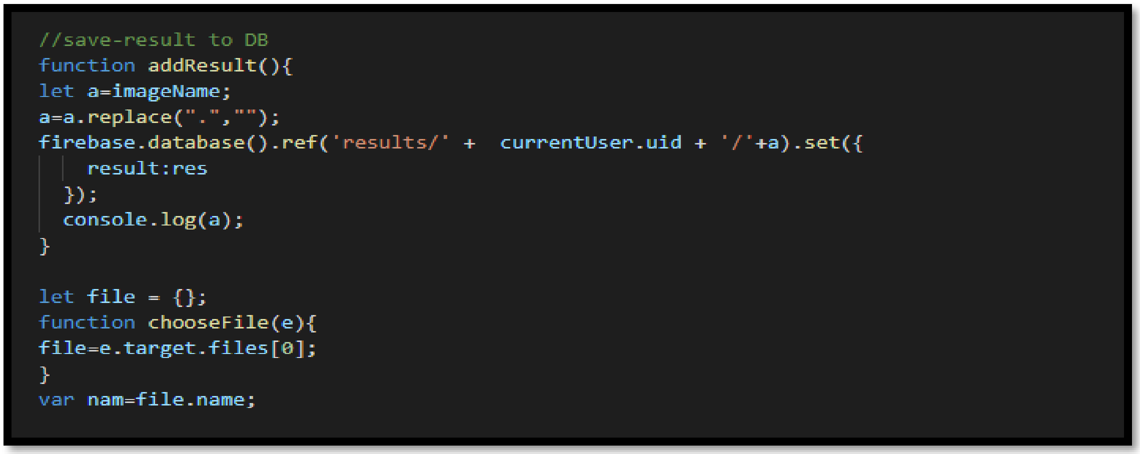


```
//save image in DB
function addImage()
{
firebase.storage().ref("users/"+ currentUser.uid + '/'+file.name).put(file).then(function(  ){
console.log("sucessfully uploaded")
}).catch(error => {
console.log(error.message);
});
}
```

Fig. 4.14 Saving Image

4.11 Saving result in database:

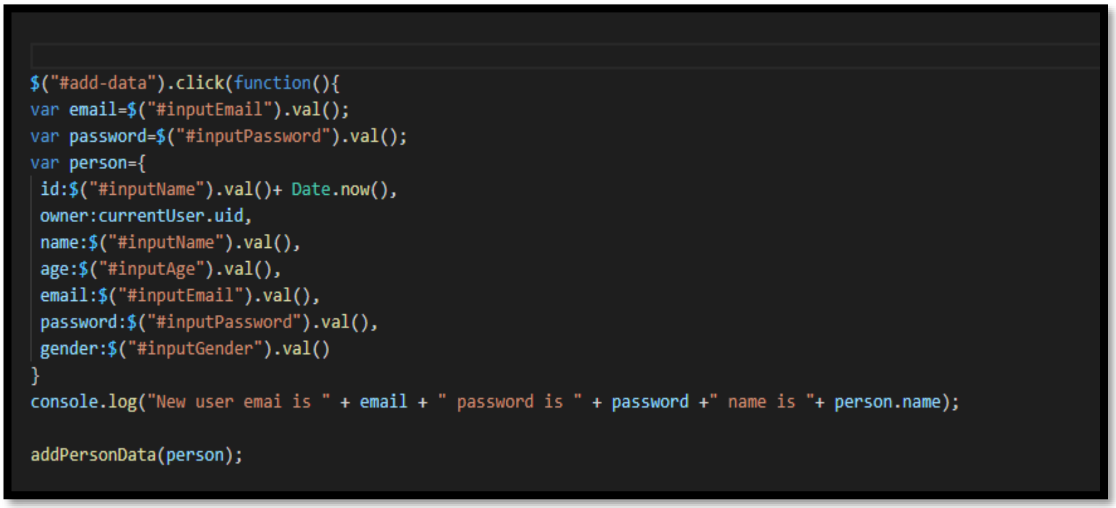
Along with the images, results are also saved in firebase database so that the history of the individual user will be maintained. Hence, the user can identify the difference in their wrinkled face by just viewing the history log.



```
//save-result to DB
function addResult(){
let a=imageName;
a=a.replace(".","");
firebase.database().ref('results/' +  currentUser.uid + '/'+a).set({
    result:res
  });
  console.log(a);
}

let file = {};
function chooseFile(e){
file=e.target.files[0];
}
var nam=file.name;
```

Fig. 4.15 Saving Result

4.12 Save personal information to Database:



```
$("#add-data").click(function(){
var email=$("#inputEmail").val();
var password=$("#inputPassword").val();
var person={
 id:$("#inputName").val()+ Date.now(),
 owner:currentUser.uid,
 name:$("#inputName").val(),
 age:$("#inputAge").val(),
 email:$("#inputEmail").val(),
 password:$("#inputPassword").val(),
 gender:$("#inputGender").val()
}
console.log("New user emai is " + email + " password is " + password +" name is "+ person.name);

addPersonData(person);
```

Fig 4.16 Saving Data

When the user signup on our system, user has to add their personal information such as name, age, gender etc. then this information will be saved in firebase database against each user id. In fig. 4.16, code is mentioned for saving information provided by the user.

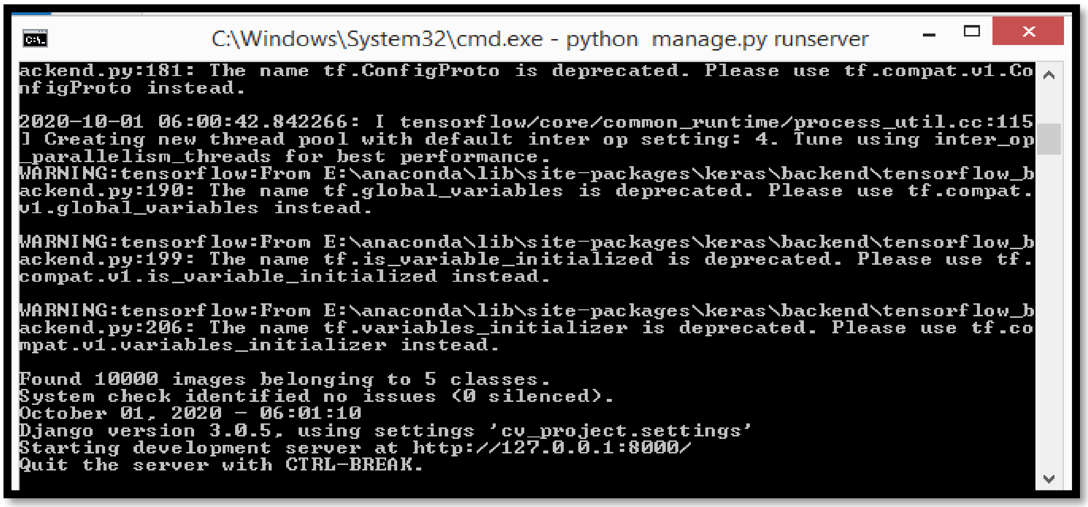
CHAPTER 5

Web Application

A web application is a software or program that can be accessed using any web browser. Our web application is designed to facilitate users to create an account, input image, find results, recommendations, and view his history to compare his all uploaded results. Out frontend is typically designed by using languages like HTML, CSS, JavaScript, Bootstrap that are supported by major browsers. While our backend was created by using a programming stack like Django, and firebase as our database, etc. For a better explanation, I am attaching screenshots With Further Descriptions.

5.1 Run The Server

First, we need to run the server that will fetch all the data and functions y using Django from our model that consists of 10000 images divided into 5 classes.

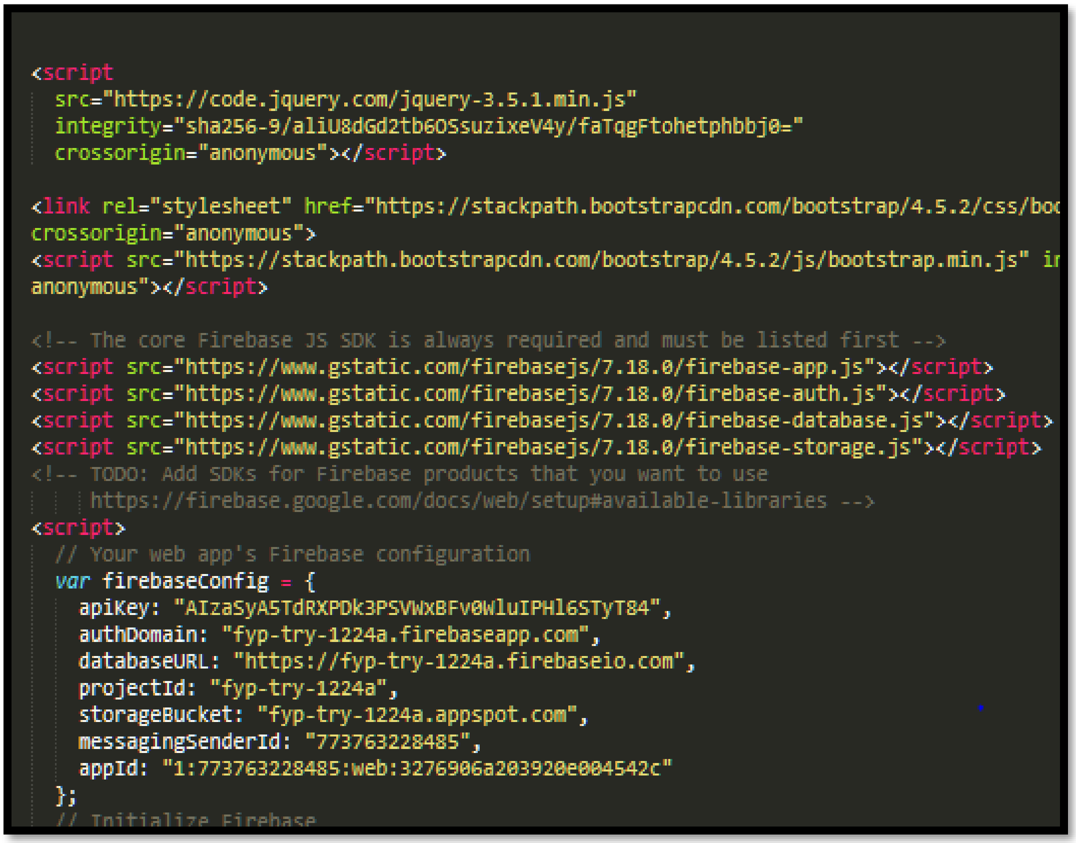


```
ackend.py:181: The name tf.ConfigProto is deprecated. Please use tf.compat.v1.Co
nfigProto instead.

2020-10-01 06:00:42.842266: I tensorflow/core/common_runtime/process_util.cc:115
] Creating new thread pool with default inter op setting: 4. Tune using inter_op
_parallelism_threads for best performance.
WARNING:tensorflow:From E:\anaconda\lib\site-packages\keras\backend\tensorflow_b
ackend.py:190: The name tf.global_variables is deprecated. Please use tf.compat.
v1.global_variables instead.

WARNING:tensorflow:From E:\anaconda\lib\site-packages\keras\backend\tensorflow_b
ackend.py:199: The name tf.is_variable_initialized is deprecated. Please use tf.
compat.v1.is_variable_initialized instead.

WARNING:tensorflow:From E:\anaconda\lib\site-packages\keras\backend\tensorflow_b
ackend.py:206: The name tf.variables_initializer is deprecated. Please use tf.co
mpat.v1.variables_initializer instead.

Found 10000 images belonging to 5 classes.
System check identified no issues (0 silenced).
October 01, 2020 - 06:01:10
Django version 3.0.5, using settings 'cv_project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Fig 5.1- Run Server

5.2 Useful Links And Scripting

User interface design or UI design commonly refers to the visual layout of the aspects that a user might interact with a website or technical product. This might be the control buttons of a radio or the visual layout of a web page. User interface designs should not only be appealing to prospective users but must also be well-designed and created functional by using and think about the user's preference in mind. For this we need to add some links and scripts that declare the respect languages that we are using. In this, we also introduced our database configurations that will collect our data and give output by fetching from firebase.



```
<script
  src="https://code.jquery.com/jquery-3.5.1.min.js"
  integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0="
  crossorigin="anonymous"></script>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/boo
crossorigin="anonymous">
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" ir
anonymous"></script>

<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/7.18.0/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.18.0/firebase-auth.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.18.0/firebase-database.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.18.0/firebase-storage.js"></script>
<!-- TODO: Add SDKs for Firebase products that you want to use
     https://firebase.google.com/docs/web/setup#available-libraries -->
<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyA5TdRXPDk3PSVWxBFv0WluIPHl6STyT84",
    authDomain: "fyp-try-1224a.firebaseapp.com",
    databaseURL: "https://fyp-try-1224a.firebaseio.com",
    projectId: "fyp-try-1224a",
    storageBucket: "fyp-try-1224a.appspot.com",
    messagingSenderId: "773763228485",
    appId: "1:773763228485:web:3276906a203920e004542c"
  };
  // Initialize Firebase
```

Fig 5.2- Firebase Configuration

5.3 Sign Up

On the first screen of the web page, the user will see when he reaches our site is sign-up page. That is design for the user if he is new and does not have an account. So, he first needs to create his account.

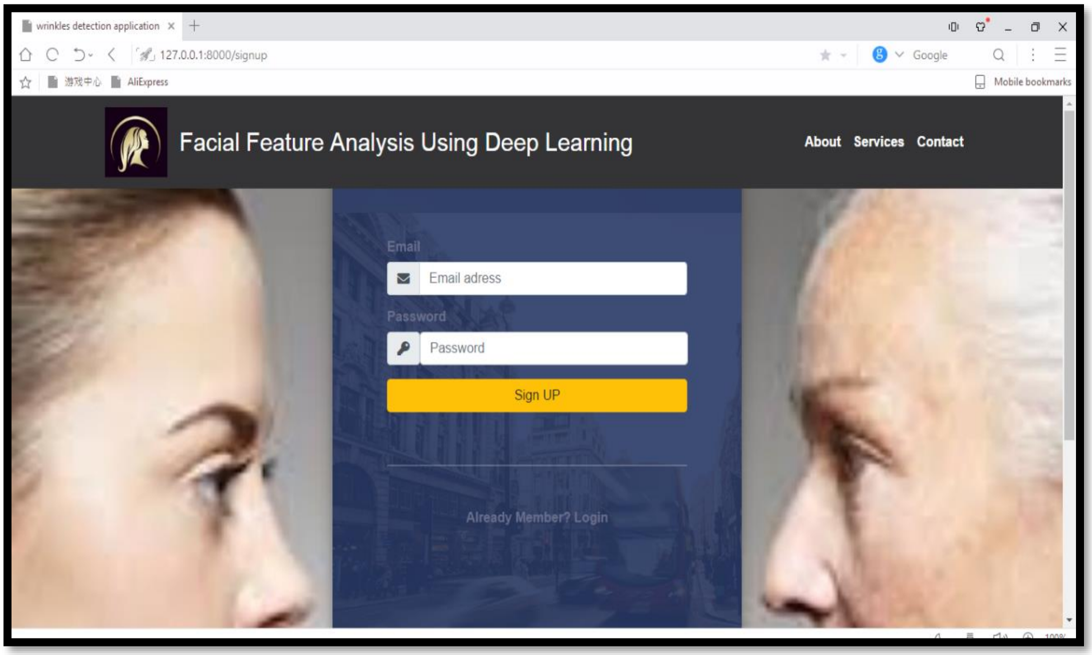
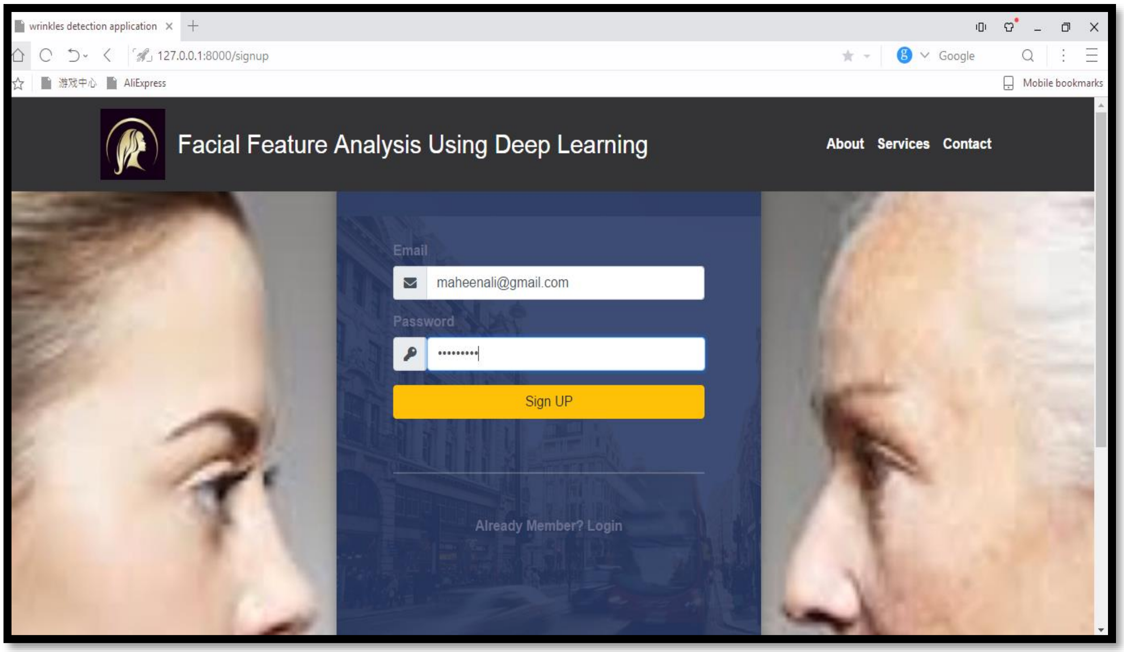


Fig 5.3- First Sign Up Page

If the user is new, he has to fill all the required fields of email and password then click on the sign-up button. By doing this our database first validates the user's input email address already exists or not.



Facial Feature Analysis Using Deep Learning

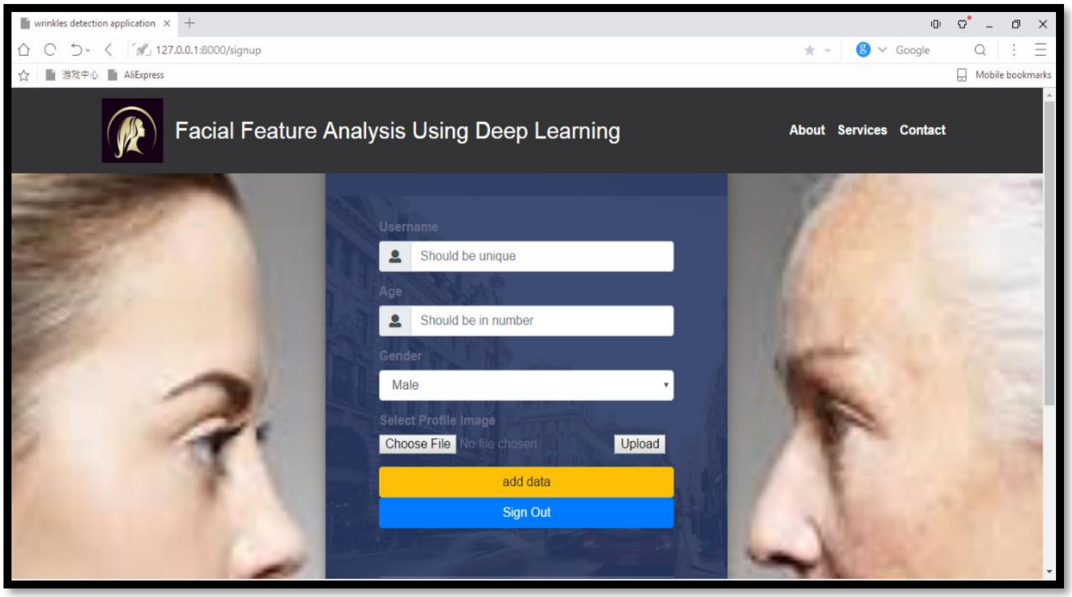
Email: maheenali@gmail.com

Password: •••••••••

Fig 5.4- Data Input in Sign Up

5.3.1 Creating an Account

If everything went well then, the user will be asked for further details of username age and gender. There are some validations of format. User has to follow it otherwise it will show an error. The user also has to upload his photo that will be his profile image. That will identify him and his details in our database. Then he will click the add data button.



Facial Feature Analysis Using Deep Learning

Username

Should be unique

Age

Should be in number

Gender

Male

Select Profile Image

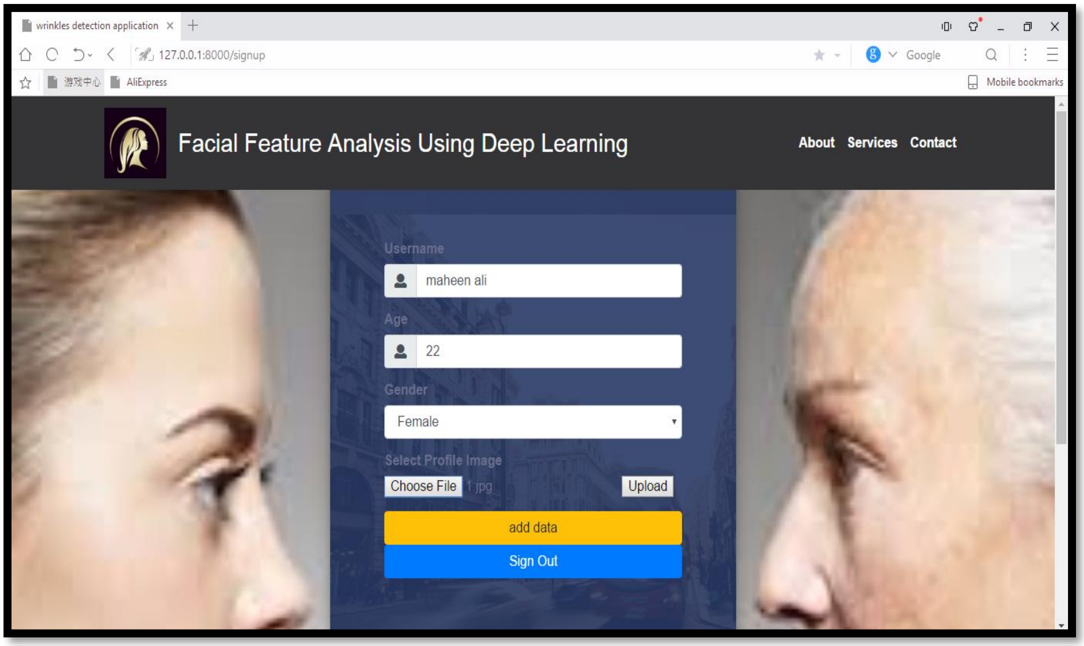
Choose File No file chosen Upload

add data

Sign Out

Fig 5.5- Add Data in Sign Up

The coding below shows the creation of all the fields.



Facial Feature Analysis Using Deep Learning

Username

maheen ali

Age

22

Gender

Female

Select Profile Image

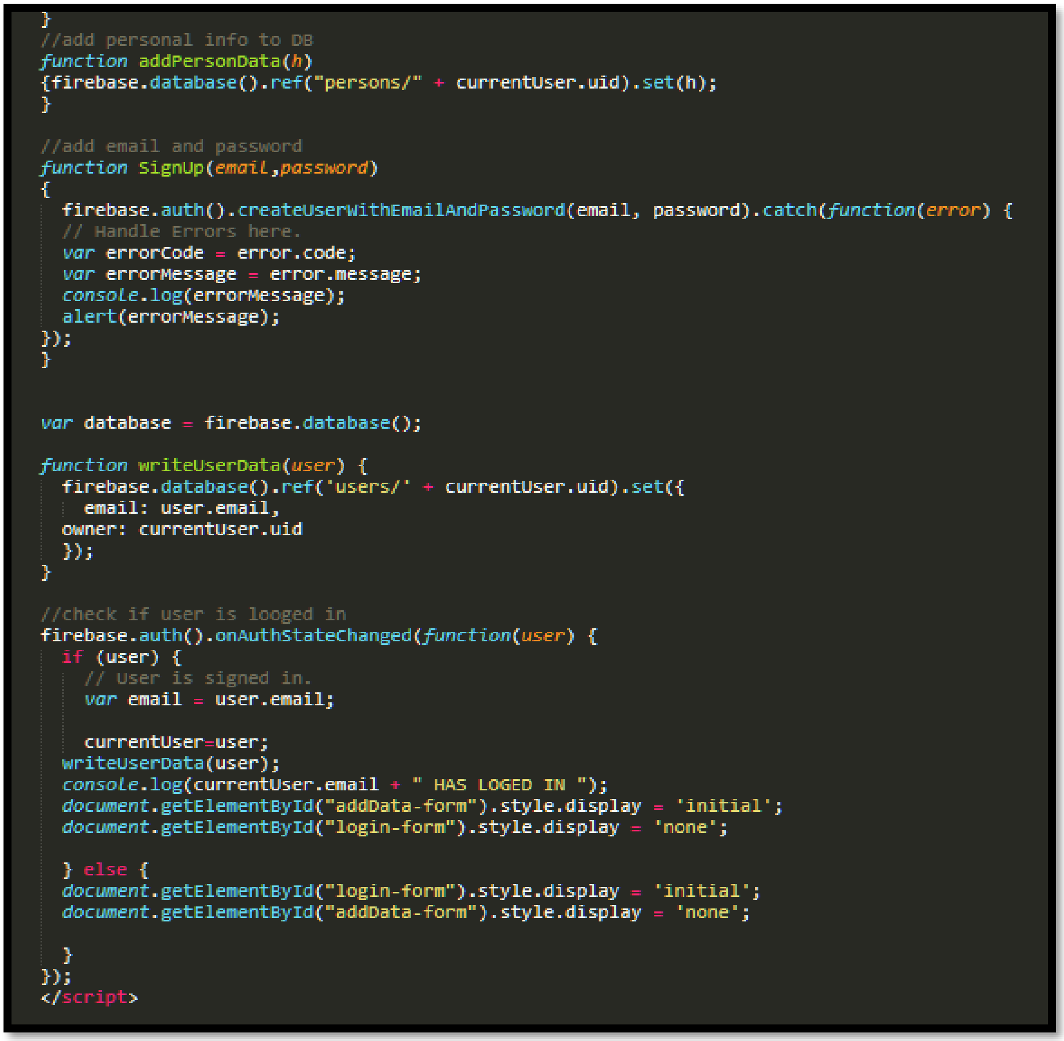
Choose File 1.jpg Upload

add data

Sign Out

Fig 5.6- Input Data in Required Format

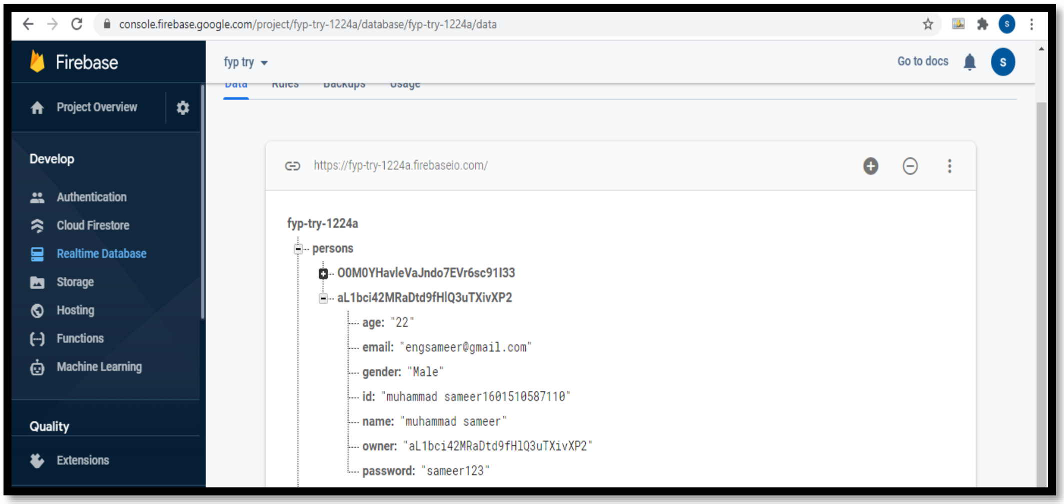
Let us talk about the logic behind all the buttons. A logout button is placed there that made it easy for the user to logout after direct creation. When the user will put email; and password the given function will search in the database first that the user already has an account or not by using his email. If the user is new, then the sign-up button will move him toward the next fields of age gender and profile picture. By clicking on the button of additional data that input data will be saved in the database. After clicking add data a person would be created in our database with his profile image and all his respective details.



```
}
//add personal info to DB
function addPersonData(h)
{firebase.database().ref("persons/" + currentUser.uid).set(h);
}

//add email and password
function SignUp(email,password)
{
  firebase.auth().createUserWithEmailAndPassword(email, password).catch(function(error) {
  // Handle Errors here.
  var errorCode = error.code;
  var errorMessage = error.message;
  console.log(errorMessage);
  alert(errorMessage);
});
}


var database = firebase.database();

function writeUserData(user) {
  firebase.database().ref('users/' + currentUser.uid).set({
    email: user.email,
  owner: currentUser.uid
  });
}

//check if user is looged in
firebase.auth().onAuthStateChanged(function(user) {
  if (user) {
    // User is signed in.
    var email = user.email;

    currentUser=user;
  writeUserData(user);
  console.log(currentUser.email + " HAS LOGED IN ");
  document.getElementById("addData-form").style.display = 'initial';
  document.getElementById("login-form").style.display = 'none';

  } else {
  document.getElementById("login-form").style.display = 'initial';
  document.getElementById("addData-form").style.display = 'none';

  }
});
</script>
```

Fig 5.7- Model-Signup-Database

The view in the database is can be seen in the picture below. An individual UID will be assigned to each user that will differentiate the user form, other users. This ID will contain all the data thread.



```
fyp-try-1224a
  persons
    O0M0YHavleVaJndo7EVr6sc91I33
    aL1bci42MRaDtd9fHlQ3uTXivXP2
      age: "22"
      email: "engsameer@gmail.com"
      gender: "Male"
      id: "muhammad sameer1601510587110"
      name: "muhammad sameer"
      owner: "aL1bci42MRaDtd9fHlQ3uTXivXP2"
      password: "sameer123"
```

Fig 5.8- User's Database Created

5.4 Sign In

If the user already has an account, he will click on the already account button and he will go to the sign-in page.

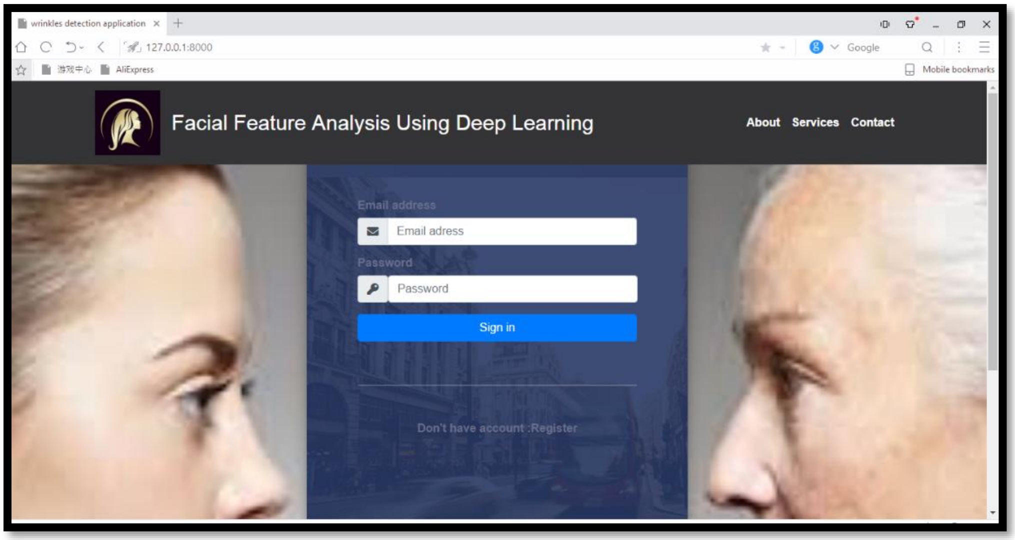
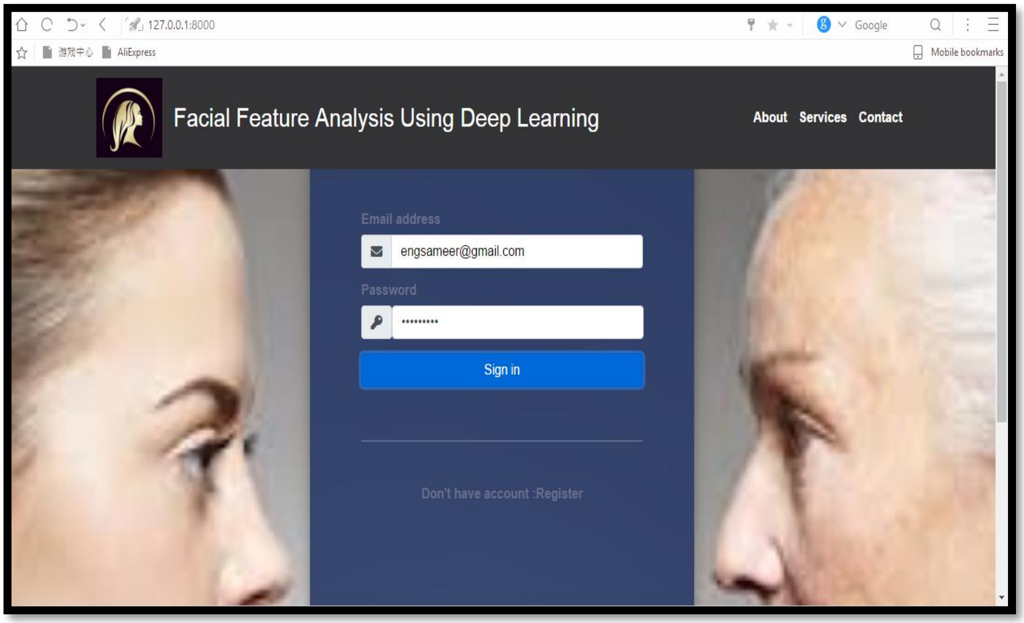


Fig 5.9- Sign In

If the user already has an account, he will enter email and password and then click on the sign-in button that will reach him to the main page of this application.

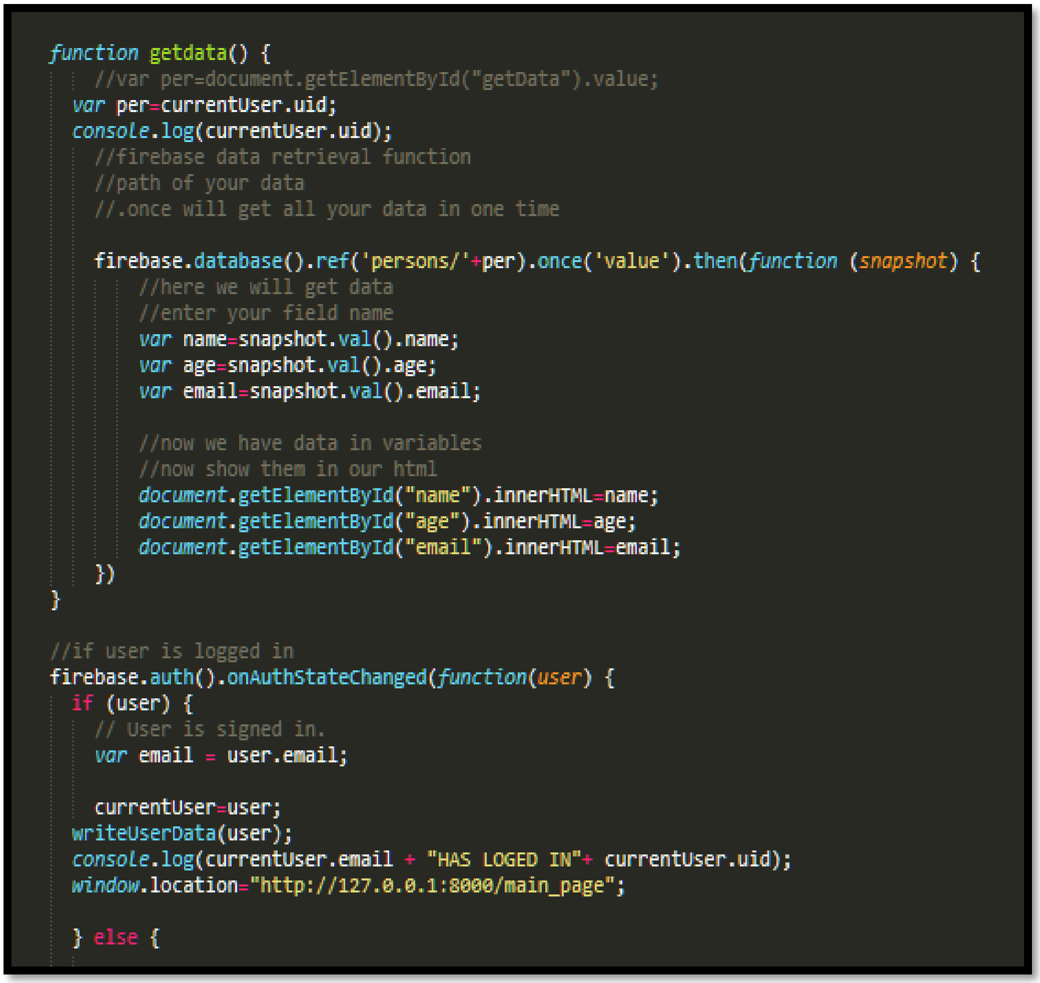


Email address: engsameer@gmail.com

Password: •••••••••

Fig 5.10- user will fill his details

User will enter his details and then click sign in. His entered details will be matched in the database that the password is correct or not, is the user is existing or not by the following code. Here the get data function is fetching the details from firebase and after authentication from the database, he will be redirected on the main page.

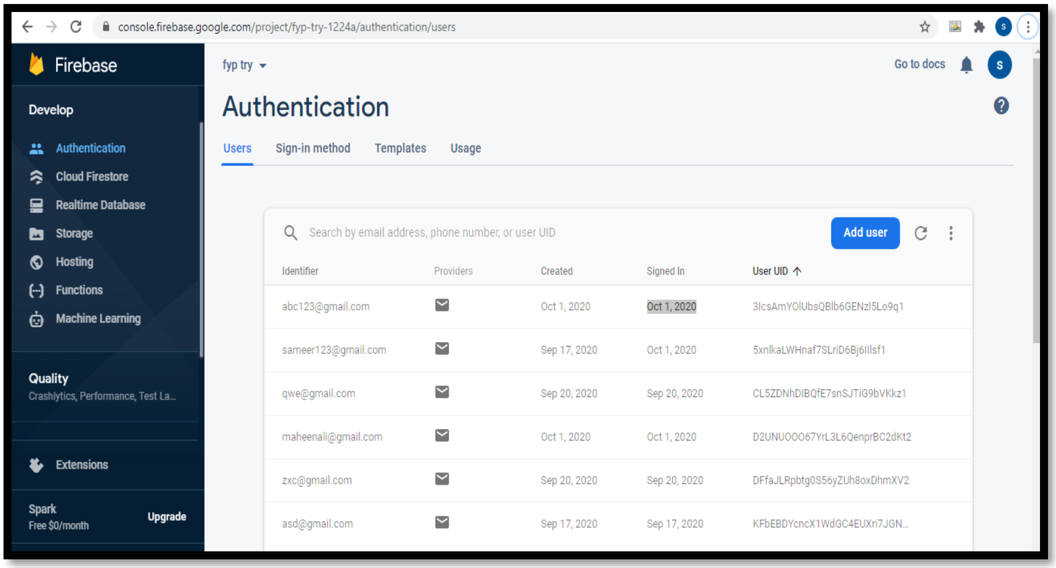


```
function getdata() {
    //var per=document.getElementById("getData").value;
  var per=currentUser.uid;
  console.log(currentUser.uid);
    //firebase data retrieval function
    //path of your data
    //.once will get all your data in one time

    firebase.database().ref('persons/'+per).once('value').then(function (snapshot) {
        //here we will get data
        //enter your field name
        var name=snapshot.val().name;
        var age=snapshot.val().age;
        var email=snapshot.val().email;

        //now we have data in variables
        //now show them in our html
        document.getElementById("name").innerHTML=name;
        document.getElementById("age").innerHTML=age;
        document.getElementById("email").innerHTML=email;
    })
}

//if user is logged in
firebase.auth().onAuthStateChanged(function(user) {
  if (user) {
    // User is signed in.
    var email = user.email;

    currentUser=user;
  writeUserData(user);
  console.log(currentUser.email + "HAS LOGED IN"+ currentUser.uid);
  window.location="http://127.0.0.1:8000/main_page";

  } else {
```

Fig 5.11- Signing- Authentication

5.4.1 Authentication

The authentication of the user is happening in the following way. All the created accounts are in our database. Here for the privacy password are encrypted and a distinct user ID is assigned to each respective different email that will authenticate the user existence.



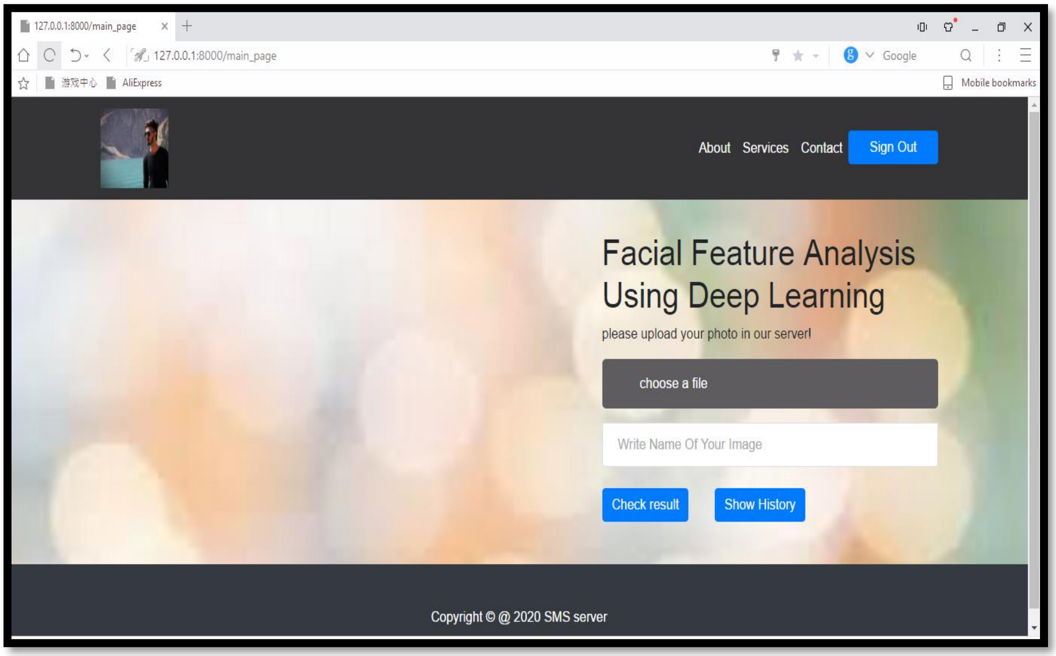
Authentication

Identifier	Providers	Created	Signed In	User UID
abc123@gmail.com	✉	Oct 1, 2020	Oct 1, 2020	3IcsAmYOlUbsQBlb6GENzl5Lo9q1
sameer123@gmail.com	✉	Sep 17, 2020	Oct 1, 2020	5xnlkaLWHnaf7SLriD6Bj6Illsf1
qwe@gmail.com	✉	Sep 20, 2020	Sep 20, 2020	CL5ZDNhDIBQfE7snSJTiG9bVKkz1
maheenali@gmail.com	✉	Oct 1, 2020	Oct 1, 2020	D2UNUOOO67YrL3L6QenprBC2dKt2
zxc@gmail.com	✉	Sep 20, 2020	Sep 20, 2020	DFfaJLRpbtg0S56yZUh8oxDhmXV2
asd@gmail.com	✉	Sep 17, 2020	Sep 17, 2020	KFbEBDYcncX1WdGC4EUXri7JGN...

Fig 5.12- Firebase Authentication

5.5 Main Page

When the user will log in into his account, the fully functional main page will be view as his main screen with his profile picture and username.



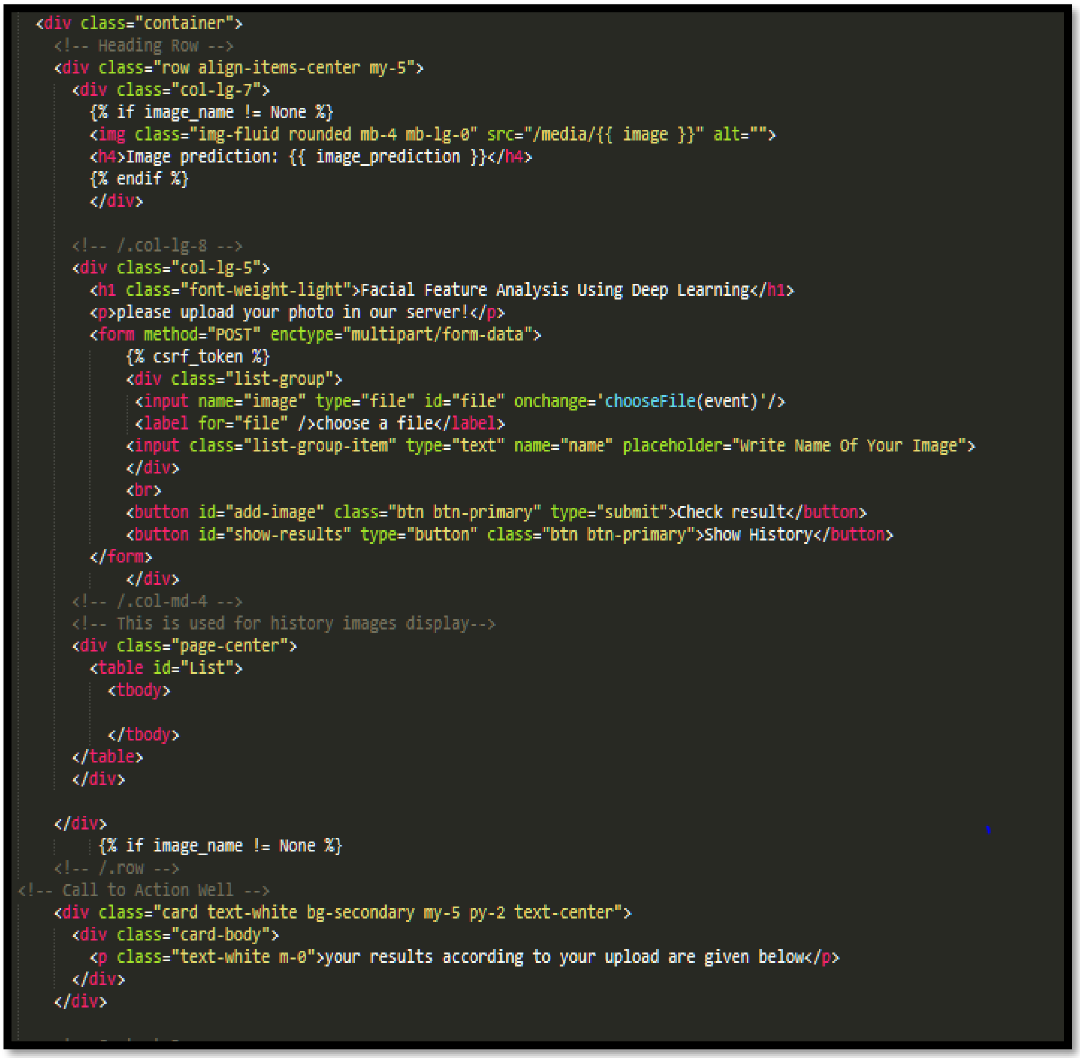
Facial Feature Analysis Using Deep Learning

please upload your photo in our server!

Copyright © @ 2020 SMS server

Fig 5.13- Main Page

I created a container in which I put all the functions and buttons. Here image prediction means a variable that will fetch the result of an image in percentage from our trained model on the basis of 5 classes.

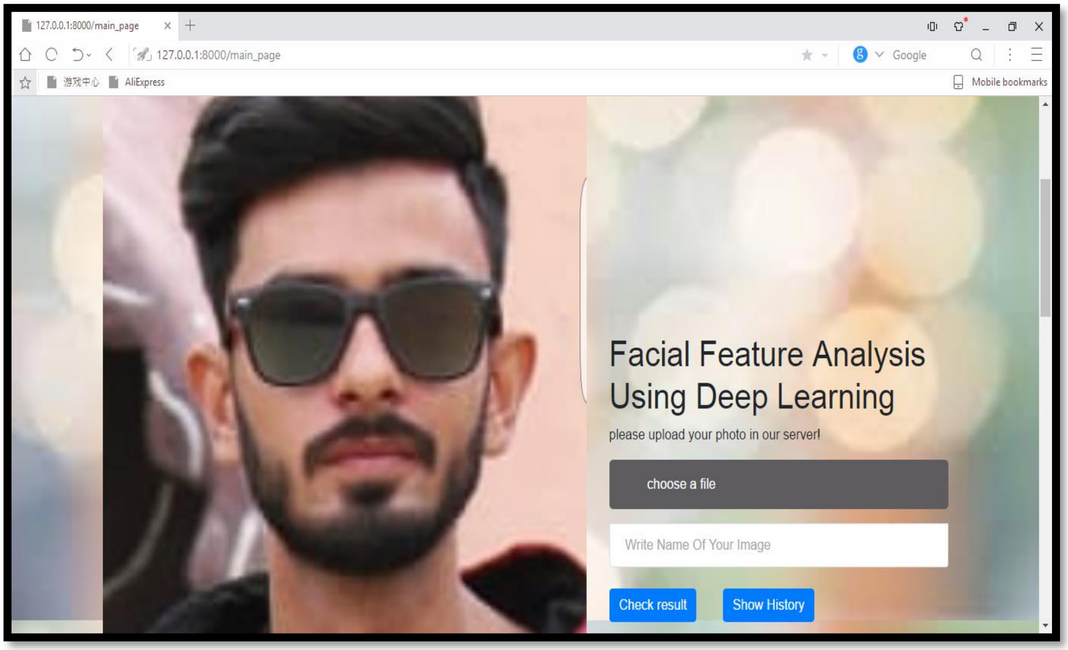


```
  <div class="container">
    <!-- Heading Row -->
    <div class="row align-items-center my-5">
      <div class="col-lg-7">
        {% if image_name != None %}
        <img class="img-fluid rounded mb-4 mb-lg-0" src="/media/{{ image }}" alt="">
        <h4>Image prediction: {{ image_prediction }}</h4>
        {% endif %}
        </div>

      <!-- /.col-lg-8 -->
      <div class="col-lg-5">
        <h1 class="font-weight-light">Facial Feature Analysis Using Deep Learning</h1>
        <p>please upload your photo in our server!</p>
        <form method="POST" enctype="multipart/form-data">
            {% csrf_token %}
            <div class="list-group">
             <input name="image" type="file" id="file" onchange='chooseFile(event)'/>
             <label for="file" />choose a file</label>
            <input class="list-group-item" type="text" name="name" placeholder="Write Name Of Your Image">
            </div>
            <br>
            <button id="add-image" class="btn btn-primary" type="submit">Check result</button>
            <button id="show-results" type="button" class="btn btn-primary">Show History</button>
        </form>
            </div>
      <!-- /.col-md-4 -->
      <!-- This is used for history images display-->
      <div class="page-center">
        <table id="List">
          <tbody>

          </tbody>
      </table>
      </div>

    </div>
         {% if image_name != None %}
    <!-- /.row -->
<!-- Call to Action Well -->
    <div class="card text-white bg-secondary my-5 py-2 text-center">
      <div class="card-body">
        <p class="text-white m-0">your results according to your upload are given below</p>
      </div>
    </div>
```

Fig 5.14- Main Page Creation

5.5.1 Input Data

The first user will choose an image for which he is testing and finding the result. the picture should be in good lightening and closed face. a picture with good and vivid pixels will give precise result and helps to find accurate percentage of wrinkles.

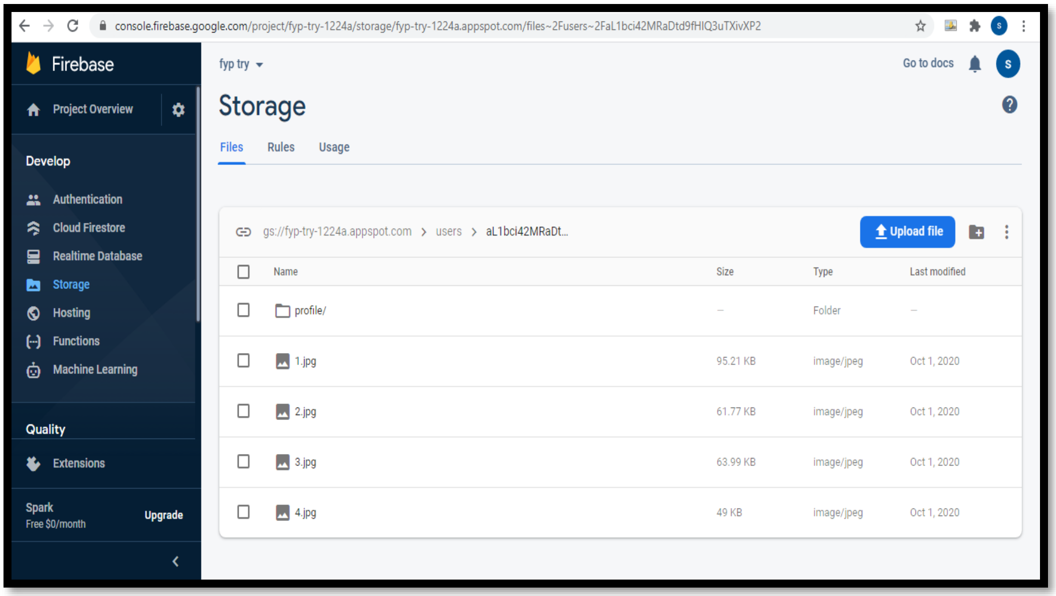


Facial Feature Analysis Using Deep Learning

please upload your photo in our server!

Fig 5.15- user will upload his image

All the images that the user will choose timely and find the result on them will be store in the storage of our database as a record. That helps the user to find the difference of results.



□	Name	Size	Type	Last modified
□	profile/	–	Folder	–
□	1.jpg	95.21 KB	image/jpeg	Oct 1, 2020
□	2.jpg	61.77 KB	image/jpeg	Oct 1, 2020
□	3.jpg	63.99 KB	image/jpeg	Oct 1, 2020
□	4.jpg	49 KB	image/jpeg	Oct 1, 2020

Fig 5.16- Images Will Be Store in Database Storage

5.5.2 Check Result

On the main page, you can see a button of check result that will show the individual result from the classes. And the main result is shown just below the uploaded picture. Our result is only based on the classification of 5 classes. More classes will give us more accuracy. But due to COVID-19, we could not move on a large amount of dataset and more number of classes because it will required a good speed of the internet to train and more storage space that was difficult to arrange for us in our limited resources at home.

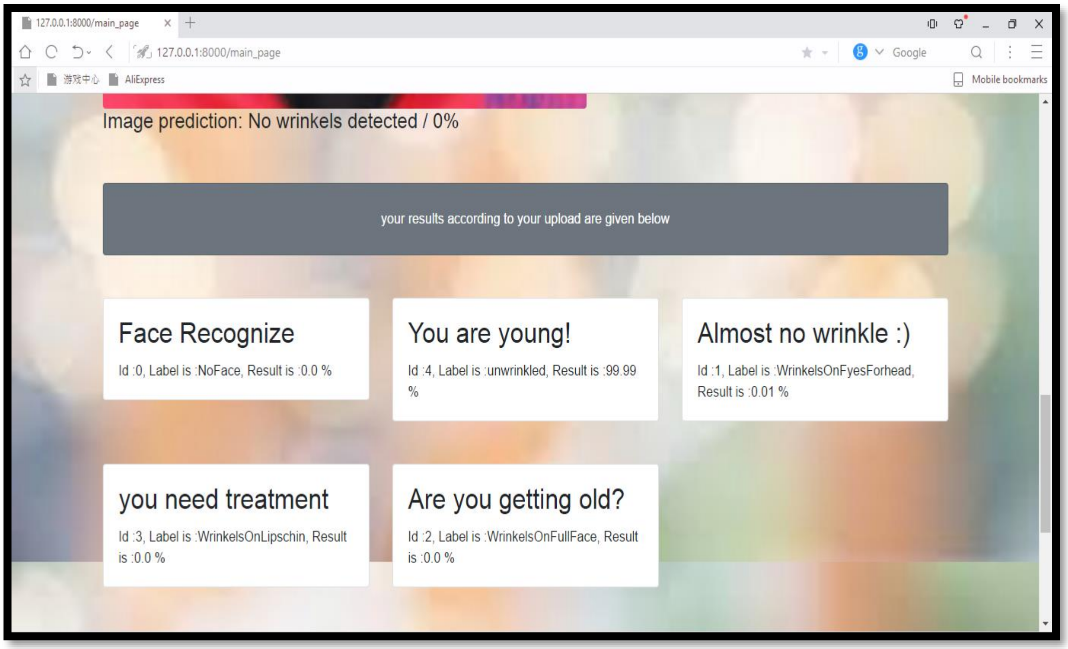


Image prediction: No wrinkels detected / 0%

your results according to your upload are given below

Face Recognize

Id :0, Label is :NoFace, Result is :0.0 %

You are young!

Id :4, Label is :unwrinkled, Result is :99.99 %

Almost no wrinkle :)

Id :1, Label is :WrinkelsOnFyesForhead, Result is :0.01 %

you need treatment

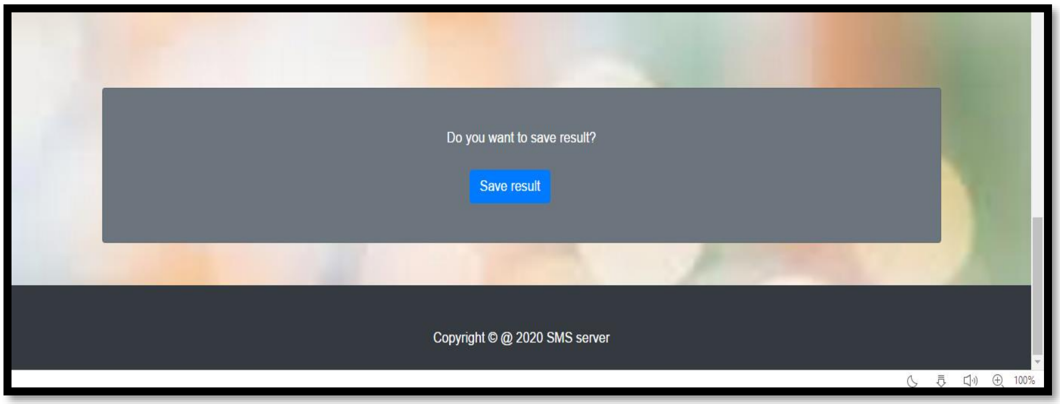
Id :3, Label is :WrinkelsOnLipschin, Result is :0.0 %

Are you getting old?

Id :2, Label is :WrinkelsOnFullFace, Result is :0.0 %

Fig 5.17- Check Result

If you want to save the result in our database, then click on the save result button that will maintain your record in our firebase. Here’s the output is showing that the user uploaded image’s human face has been detected that’s why it’s showing 0% for no face. The image belongs to young face user that’s fall in class 1 of low wrinkles.



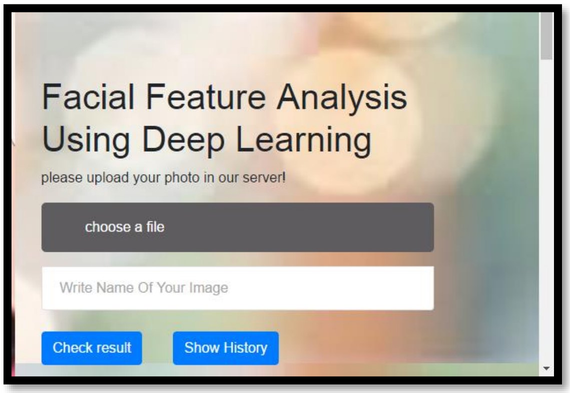
Do you want to save result?

Copyright © @ 2020 SMS server

Fig 5.18- Save Results

5.5.3 User's Record

If you want to see all your saved results, then click on the show history button.

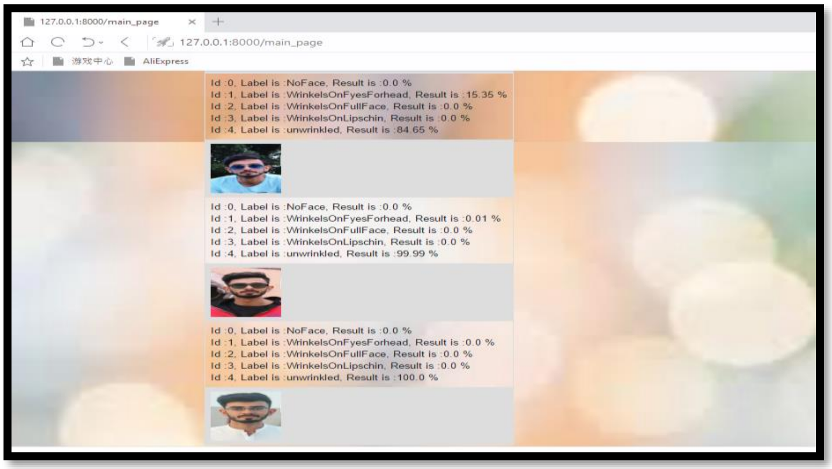


Facial Feature Analysis Using Deep Learning

please upload your photo in our server!

Fig 5.19- History

The history will be displayed in the form of a table as shown below.



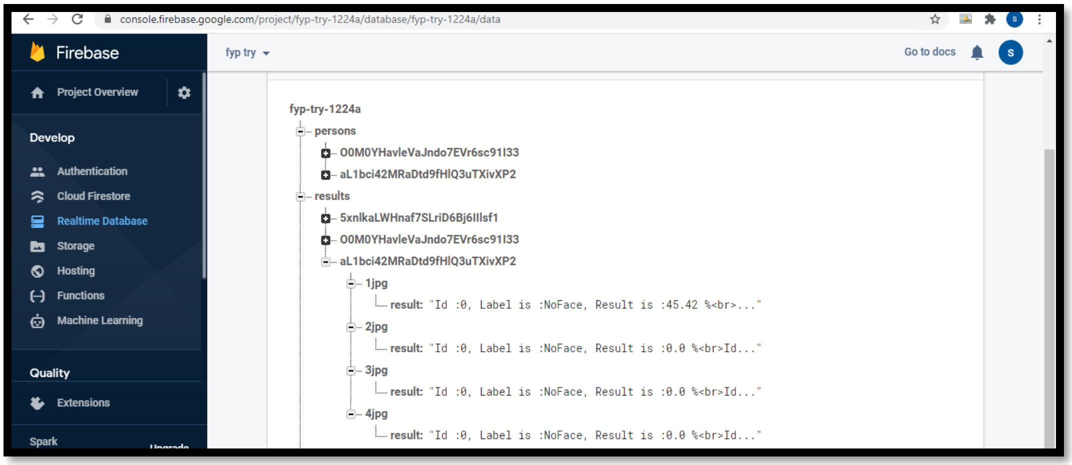
```
Id :0, Label is :NoFace, Result is :0.0 %
Id :1, Label is :WrinkelsOnFyesForhead, Result is :15.35 %
Id :2, Label is :WrinkelsOnFullFace, Result is :0.0 %
Id :3, Label is :WrinkelsOnLipschin, Result is :0.0 %
Id :4, Label is :unwrinkled, Result is :84.65 %
```

```
Id :0, Label is :NoFace, Result is :0.0 %
Id :1, Label is :WrinkelsOnFyesForhead, Result is :0.01 %
Id :2, Label is :WrinkelsOnFullFace, Result is :0.0 %
Id :3, Label is :WrinkelsOnLipschin, Result is :0.0 %
Id :4, Label is :unwrinkled, Result is :99.99 %
```

```
Id :0, Label is :NoFace, Result is :0.0 %
Id :1, Label is :WrinkelsOnFyesForhead, Result is :0.0 %
Id :2, Label is :WrinkelsOnFullFace, Result is :0.0 %
Id :3, Label is :WrinkelsOnLipschin, Result is :0.0 %
Id :4, Label is :unwrinkled, Result is :100.0 %
```

Fig 5.20- User's Tested Images Record

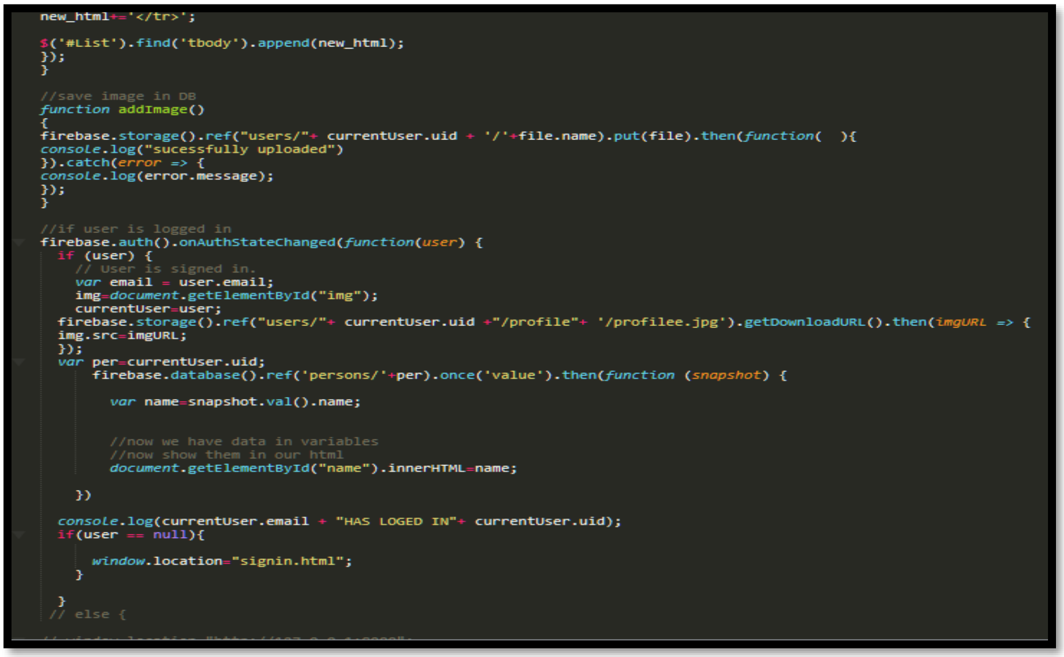
The records of the user's input tested image are marinating in the firebase in this format individually for all images that are extracted from the database and display on the user interface. All the result comprises on the 5 classes.



```
fyp-try-1224a
  persons
    O0M0YHavleVaJndo7EVr6sc91I33
    aL1bci42MRaDtd9fHlQ3uTXivXP2
  results
    5xnlkaLWHnaf7SLriD6Bj6Illsf1
    O0M0YHavleVaJndo7EVr6sc91I33
    aL1bci42MRaDtd9fHlQ3uTXivXP2
      1jpg
        result: "Id :0, Label is :NoFace, Result is :45.42 %<br>..."
      2jpg
        result: "Id :0, Label is :NoFace, Result is :0.0 %<br>Id..."
      3jpg
        result: "Id :0, Label is :NoFace, Result is :0.0 %<br>Id..."
      4jpg
        result: "Id :0, Label is :NoFace, Result is :0.0 %<br>Id..."
```

Fig 5.21- User History Maintaining

In this snapshot the details of the functional buttons are given first is the logout button that will sign off the user from his account. Other is add result, here add result is used to upload image in the database. Next shows result, it is used to display all the previous results of the user's history.



```
new_html+='</tr>';

$('#List').find('tbody').append(new_html);
});
}

//save image in DB
function addImage()
{
firebase.storage().ref("users/"+ currentUser.uid + '/'+file.name).put(file).then(function(  ){
console.log("sucessfully uploaded")
}).catch(error => {
console.log(error.message);
});
}

//if user is logged in
firebase.auth().onAuthStateChanged(function(user) {
  if (user) {
    // User is signed in.
    var email = user.email;
    img=document.getElementById("img");
    currentUser=user;
  firebase.storage().ref("users/"+ currentUser.uid +"/profile"+ '/profilee.jpg').getDownloadURL().then(imgURL => {
  img.src=imgURL;
  });
  var per=currentUser.uid;
      firebase.database().ref('persons/'+per).once('value').then(function (snapshot) {

        var name=snapshot.val().name;


        //now we have data in variables
        //now show them in our html
        document.getElementById("name").innerHTML=name;

    })

  console.log(currentUser.email + "HAS LOGED IN"+ currentUser.uid);
  if(user == null){

      window.location="signin.html";
    }

  }
 // else {
```

Fig 5.22- Snapshot Of Authentication Functions

s

CHAPTER 6

Conclusion

This report has proposed a technique of detecting wrinkles on a facial image with the help of a deep learning method, that is based on the convolutional neural network (CNN) using multiple face input feeds. Result of 70-80% accuracy is obtained by using a database of only five classes comprising of 10000 images dataset. VGG16 model is used that trained a huge amount of data and test image to analyses wrinkle, non-wrinkle, low wrinkle, and moderate level wrinkles. The analysis and experimental results suggest that for vivid pixels as the inputs, the CNN model with deeper structure, larger input images, and smaller convolution kernels, reaches better performance with good accuracy. At this stage of our final evaluation, our project is giving our promising result of wrinkle detection on the human face by the web-based application that maintains a record of user and displays individual input's result. It could further classify wrinkles with some more accuracy that precise the result and will give an 85-90% accurate result.

REFERENCES

[1] Cohn and F. De la Torre, “Automated face analysis for affective,” The Oxford Handbook of Affective Computing, pp. 131–150, 2014.

[2] Batool and R. Chellappa, “Modeling and detection of wrinkles in aging human faces using marked point processes,” in European Conference on Computer Vision Workshops. Springer, 2012, pp. 178–188.

[3] Batool and R. Chellappa, “Detection and inpainting of facial wrinkles using texture orientation fields and Markov random field modeling,” IEEE Trans. On Image Processing, vol. 23, no. 9, pp. 3773–3788, 2014.

[4] Ramanathan and R. Chellappa, “Modeling shape and textural variations in aging faces,” in IEEE International Conference on Automatic Face & Gesture Recognition, 2008, pp. 1–8.

[5] Batool and R. Chellappa, “Modeling and Detection of Wrinkles in Aging Human Faces Using Marked Point Processes,” Computer Vision – ECCV 2012. Workshops and Demonstrations Lecture Notes in Computer Science, pp. 178–188, 2012.

[6] Lemperle, R. E. Holmes, S. R. Cohen, and S. M. Lemperle, “A Classification of Facial Wrinkles,” Plastic and Reconstructive Surgery, vol. 108, no. 6, pp. 1751–1752, 2001.

[7] “RYNKL wrinkle analysis app,” RYNKL wrinkle analysis app. [Online]. Available: http://www.rynkl.com/. [Accessed: 21-Feb-2020].

[8] Jana, D. Datta, and R. Saha, “Age Estimation from Face Image Using Wrinkle Features,” Procedia Computer Science, vol. 46, pp. 1754–1761, 2015.

[9] Batool and R. Chellappa, “Fast detection of facial wrinkles based on Gabor features using image morphology and geometric constraints,” Pattern Recognition, vol. 48, no. 3, pp. 642– 658, 2015.

[10] Hayatin, “Pengelompokan usia berdasarkan citra wajah menggunakan algoritma neural network dengan fitur face anthropometry dan kedalam kerutan,” Teknologi, vol. 6, no. 2, p. 61, Jan. 2016.

[11] Muhimmah, F. Abriyani, and A. Kurniawardhani, “Automatic wrinkles detection on face image,” IOP Conference Series: Materials Science and Engineering, vol. 482, p. 012026, Nov. 2019.

[12] Kloeppel, “Step right up, let the computer look at your face and tell you your age,” 2010. [Online]. Available: http://news.illinois.edu/news/08/0923age.html

[13] Fu, G. Guo, and T. Huang, “Age synthesis and estimation via faces: a survey,” IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 32, no. 11, pp. 1955–1976, 2010.

[14] Mark, J. Pittenger, H. Hines, C. Carello, R. Shaw, and J. Todd, “Wrinkling and head shape as coordinated sources of age-level information, ”Attention, Perception,& Psychophysics, vol. 27, no. 2, pp. 117–124, 1980.

[15] gartstein ands. A. Shaya, “Assessment of visual signs of skin aging,” Bioengineering of the Skin: Cutaneous Blood Flow and Erythema, pp. 331–344, 1996.

[16] Cula, P. R. Bargo, A. Nkengne, and N. Kollias, “Assessing facial wrinkles: automatic detection and quantification,” skin research and technology, vol.19, no.1, pp. E243–e251, 2013.

[17] https://missinglink.ai/guides/keras/using-keras-flatten-operation-cnn-models-code-examples/

[18] https://neurohive.io/en/popular-networks/vgg16/

[19] https://arsfutura.com/magazine/face-recognition-with-facenet-and-mtcnn/