Identifying Gender from Facial Features

PROJECT REPORT

Of Summer Internship

BACHELOR OF TECHNOLOGY

Computer Science

Submitted by

Divyanshu Gupta - R110218053

Pratham Pandey - R134218123

Pulkit Mittal - R134218125

Daksh Kaushik - R142217033

Anurag - R178218006

Guided by

Gunjan Chhabra

INDEX

1. Background

1.1 Aim

The project aims at exploring various aspects of Gender classification, a binary classification problem that infers whether the person in a given picture is male or female. It uses CNN to achieve its target.

Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition. Image classifications, object detections, recognition faces etc. are some of the areas where CNNs are widely used. CNN image classification takes an input image, process and classifies it under certain categories.

1.2 Technologies

The face detection was performed using **OpenCV**, a library of programming functions mainly aimed at real-time computer vision. Following that processed facial image data was matched with knowledge base containing classified male and female faces. A model was created with the help of **tensorflow**, an end-to-end open source platform for machine learning, that was later used for real time face detection and gender classification.

A frontend was created using **flask,** a micro web framework written in Python with support of **JavaScript**, **HTML** and **CSS**.

1.3 Hardware Architecture

The project is designed using a typical steep learning architecture as displayed below.

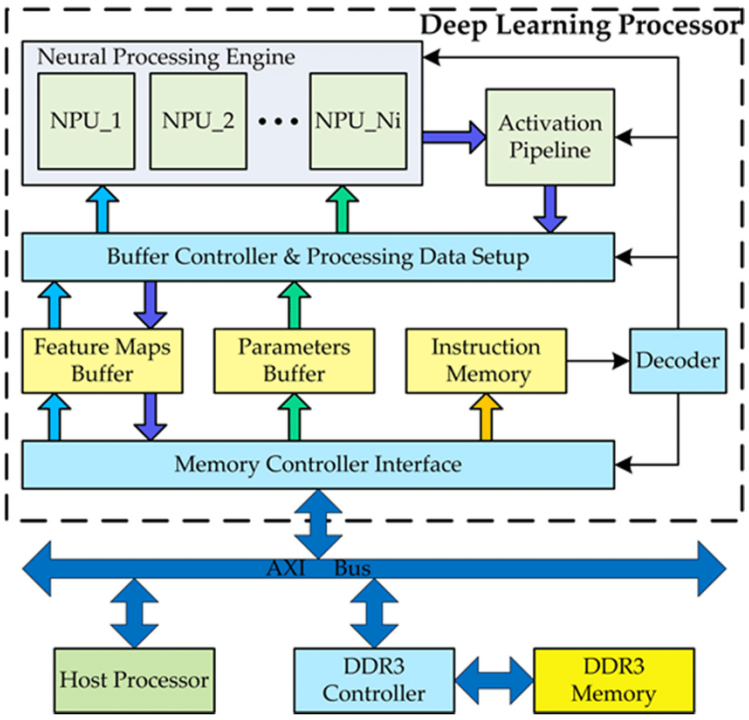


Fig 1. Taken from [The overall diagram of the designed hardware architecture of the deep... | Download Scientific Diagram (researchgate.net)]

1.4 Software Architecture

The project comprises of two modules, a training module that is the enrolment module and a test module which is the verification module. In the training module an image dataset is taken which is then pre-processed and analysed to create a model. To the other module which is the working module involves a front end which is designed using flask and style using JavaScript HTML and CSS. It takes image as input using webcam or using an upload option and then predicts the gender of the person in the image.

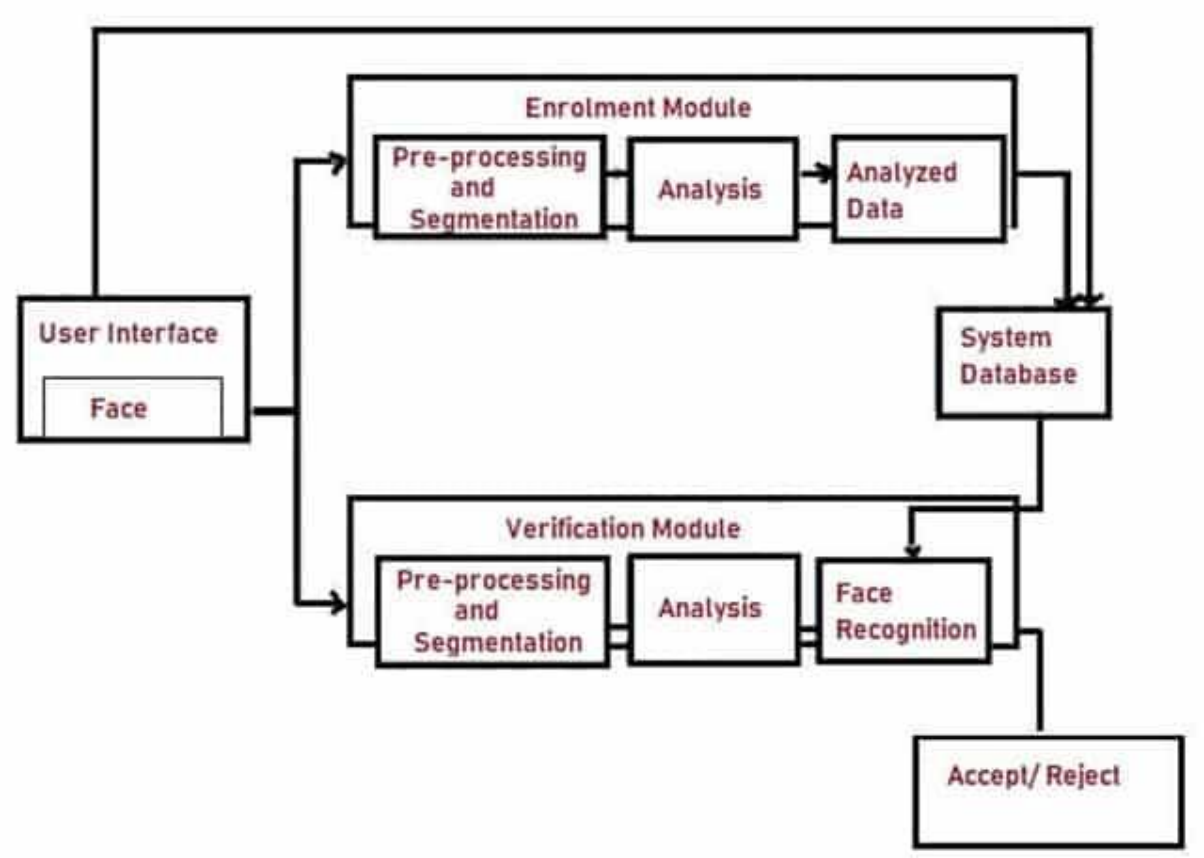


Fig 2. Taken from [Facial Recognition System - How it Works, Architecture & Applications (electricalfundablog.com)]

2. System

2.1 Requirements

2.1.1 Functional Requirements: The program is expected to take image input and accurately predict gender of the people in the picture.

2.1.2 User Requirements: A fluid and user-friendly interface which allows easy access choose the features provided.

2.1.3 Environmental Requirements:

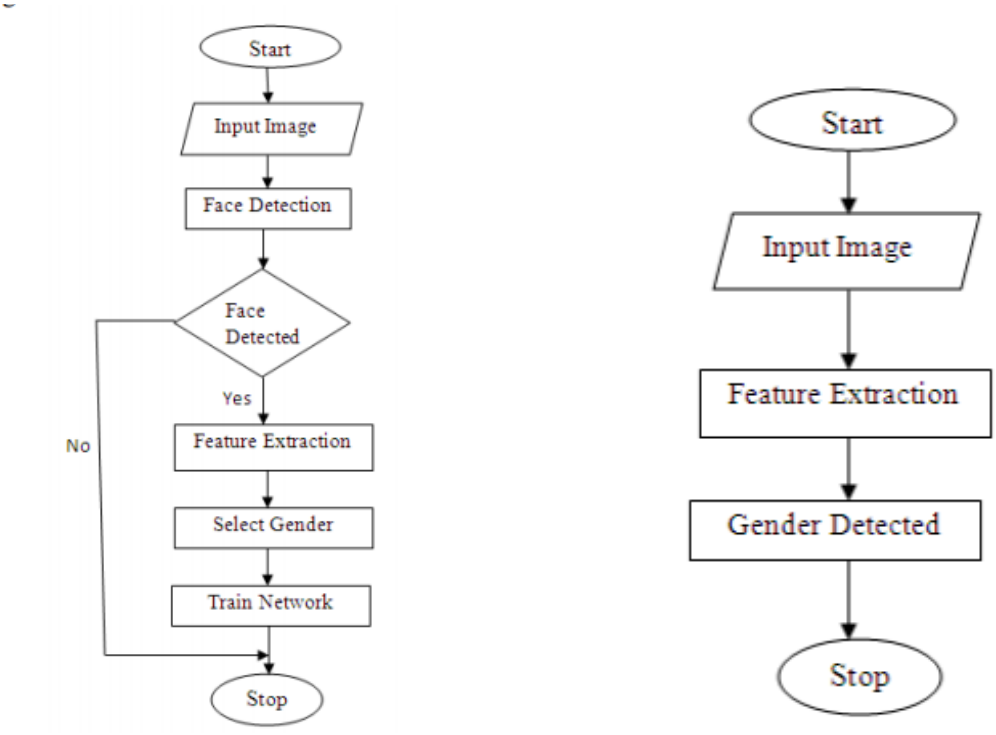
Hardware: 8GB RAM 8GB Disk Space GPU capabilities for deep learning

Software: Anaconda3

OS: Windows 7/10, MacOS (>10.0), Ubuntu

2.2 Design and Architecture

The project involves two iterations. First involves training of the model while the second one involves using the trained model to make predictions. They're represented by the follo0ing flowcharts.



Training Testing

Fig. 3 Taken from Microsoft Word - 20. ijcsit paper (psu.edu)

The project is comprised of multiple components which have been integrated into one to work effectively. The components are as follows:

- Initially, a piece of code is used to train a model to perform gender detection. It includes defining tuning parameters like number of epochs, taking training dataset as input, data preprocessing, dividing the dataset, creating the model and testing the model using certain metrics. This is done with the help of tensorflow, matplotlib and opencv.
- Now that we've trained the model, we need to use the model to perform predictions. This part is done using python in app.py, which includes different functions to take image as input via upload or run a live feed using webcam. The images are handled using opencv. Flask is used to work with the frontend, which is then integrated with HTML pages, styled using CSS and Javascript. Different components of functions have different endpoints which render a certain component according to functionality needed.
- The HTML component includes three files namely base.html, index.html and webcam.html. base.html contains a basic template for the frontend which is inherited by index.html and webcam.html. Index.html is rendered upon starting the server and opening the page for the project in a web browser. Webcam.html is rendered if webcam function is called from app.py.
- CSS is used to beautify the webpage by adding design elements and a beautiful background.
- The image request and response are handled using javascript.

2.3 Implementation

The Model is trained on 10,000 images of male and 10,000 images of female. The model is made using multiple number of convolution layers having different filters like 32, 64, 128 using activation function **relu**. The output is normalized using BatchNormalization and after each convolution layer a maximumpooling layer is applied to extract the best features from it. After each layer 25% of neurons are dropped to avoid overfitting. The layers are then flattened to get the output in 1D which is passed through sigmoid activation function so that we can get the output in the range of 0 and 1

2.4 Testing

2.4.1 Test Plan Objectives

The program involves simple data input in form of image and after the image is processed the output is received. There isn't a much of a security risk from that perspective, but we still need to consider a scenario where random images with no faces are uploaded.

2.4.2 Data Entry

Data entry is done using safe POST method.

2.4.3 Test Strategy

All the modules of the application were separately tested and their final integration was tested by checking each option and giving different formats of images as input.

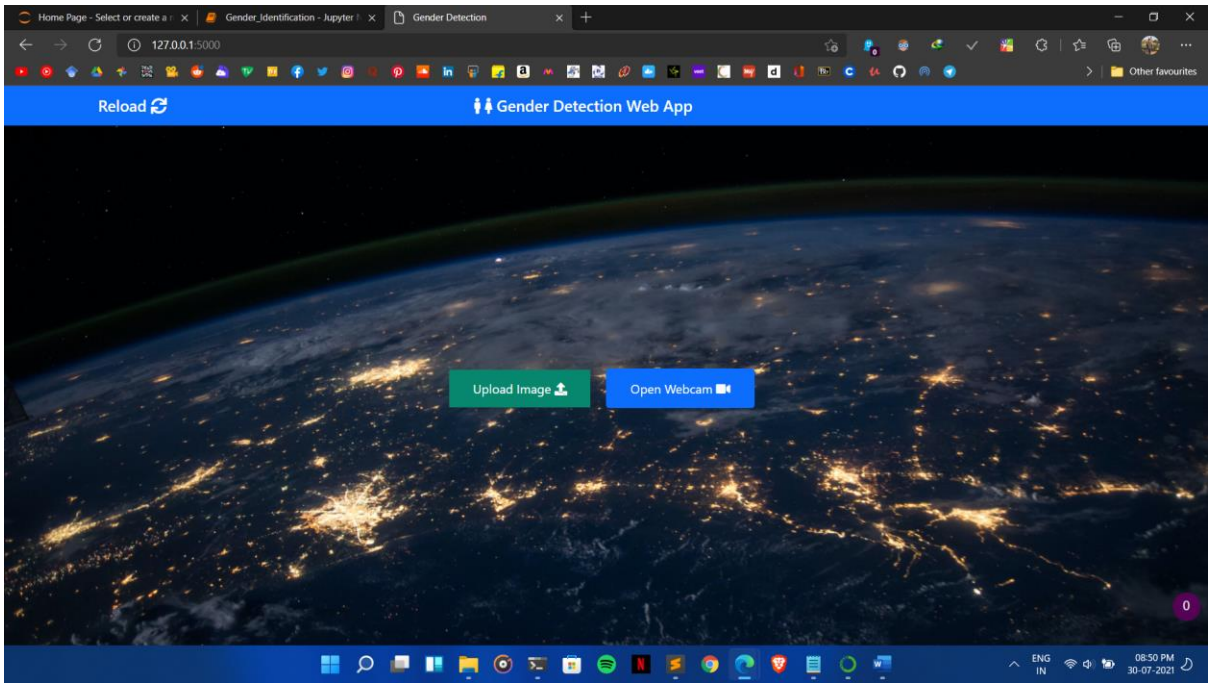
2.4.4 Basic test

A scope of crashing the program lies on putting the program into an infinite loop by giving irrelevant images with no face but this problem was tackled and “no faces found” response was received.

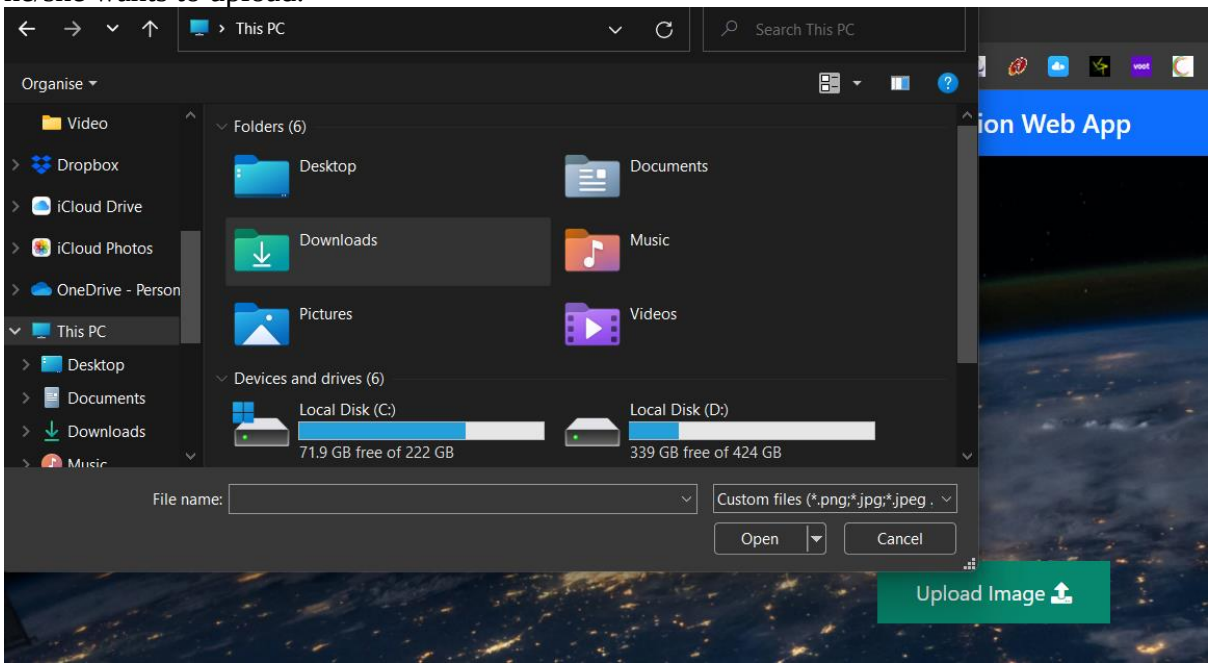
2.5 Graphical User Interface (GUI) Layout

The GUI was built using Flask, JavaScript, HTML and CSS.

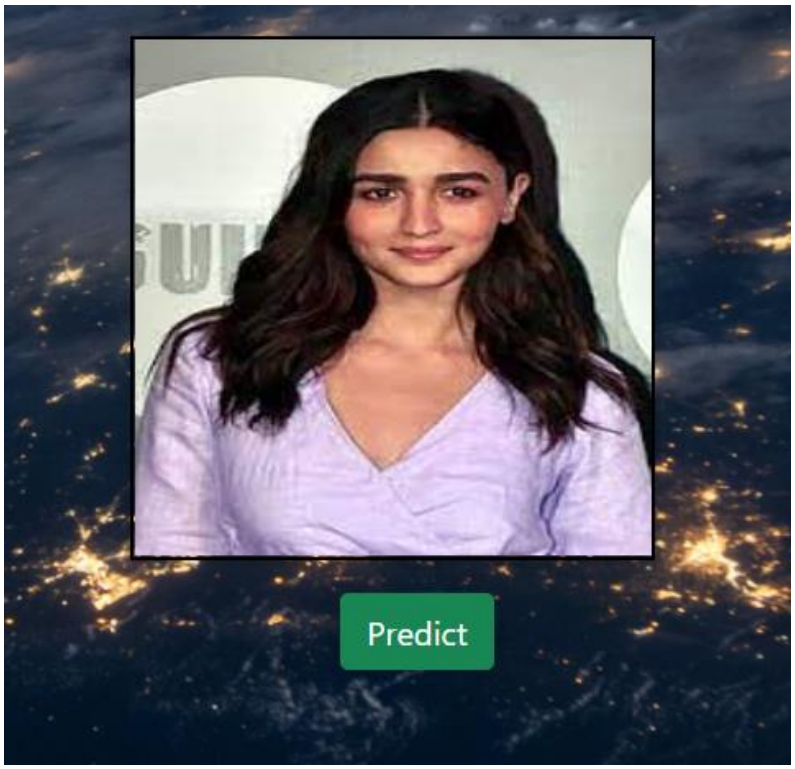
The welcome window consists of two options which include webcam and image upload functionality.



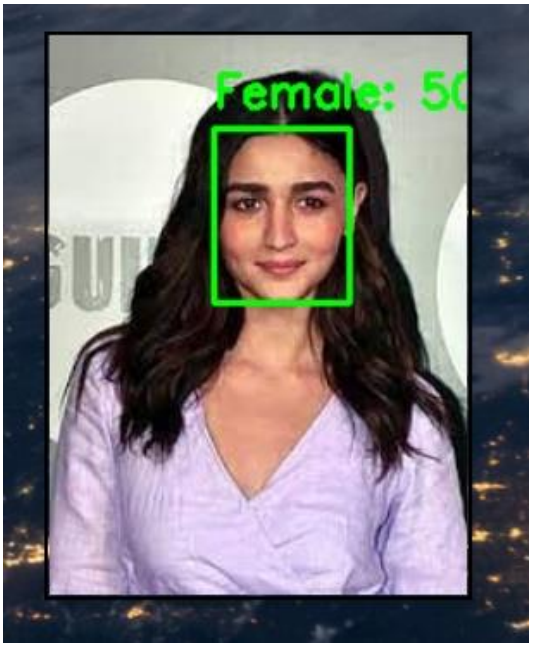
The upload image option opens a file explorer window, allowing the user to find the image he/she wants to upload.



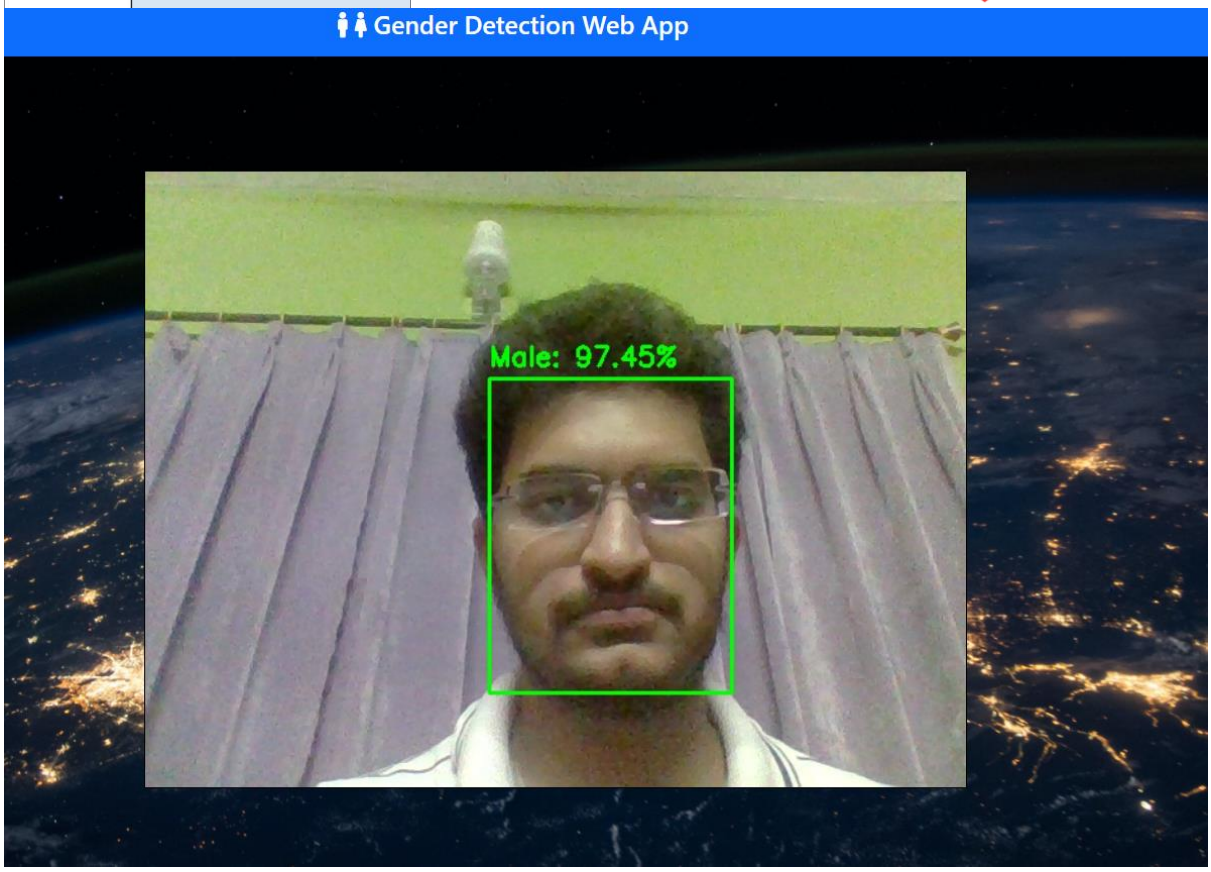
Once the user has selected the picture, he/she can click on predict and wait for results.



The result is displayed with a green square around the detected faces, along with the prediction.



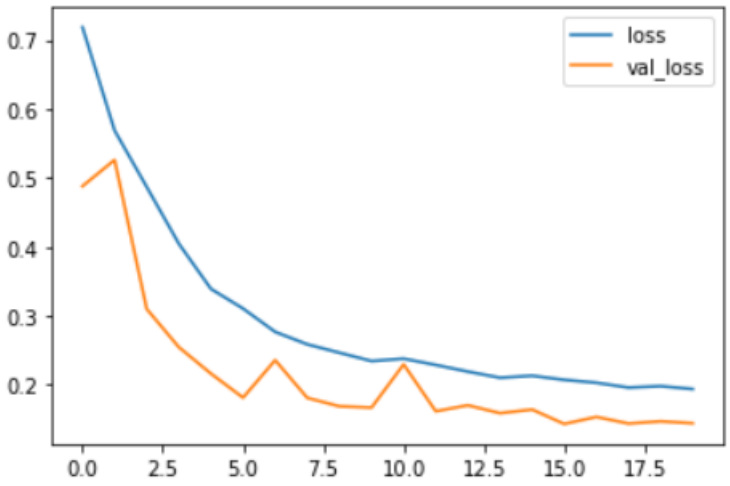
Upon clicking the open webcam option the webcam starts and a live feed is started where gender prediction takes place in real time.



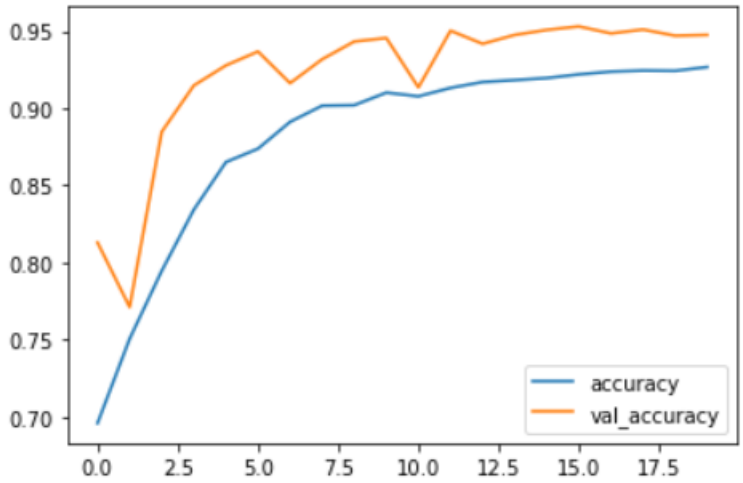
2.6 Evaluation

2.6.1 Performance

The main performance evaluation criteria of this project lies on the accuracy it provides. The following graph shows training vs validation accuracy loss of the model:



The following graph shows training accuracy vs validation accuracy



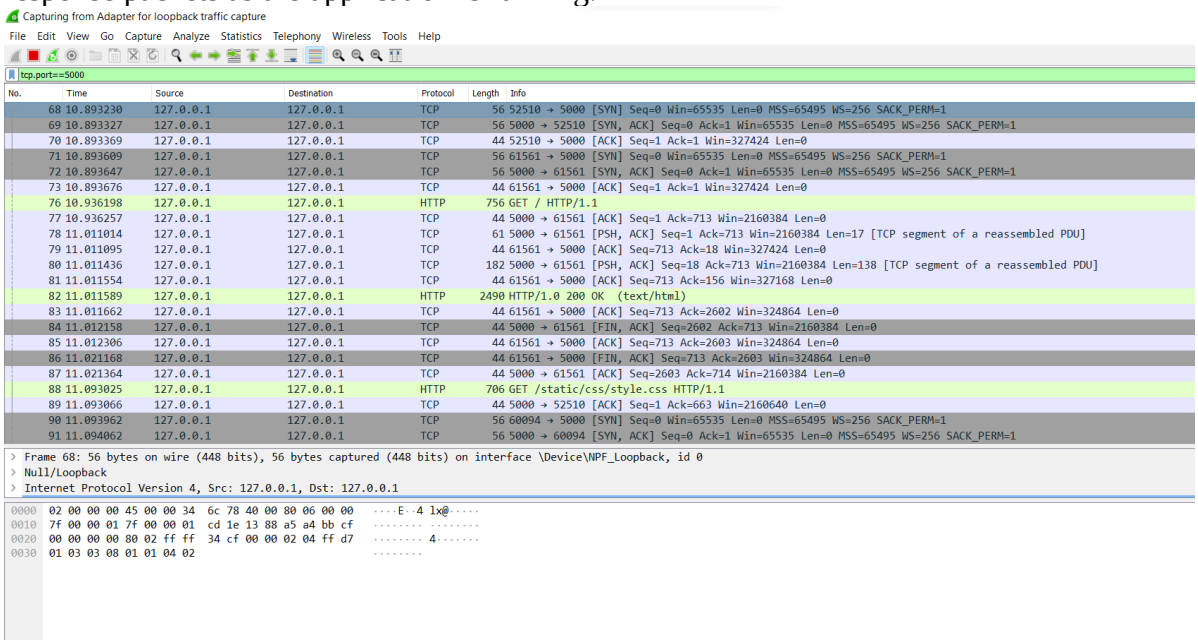
Upon working with a wide dataset of images of people from multiple ethnicities, we were able to get a pretty high accuracy.

2.6.2 Static Code Analysis

A manual analysis of code was done with the help of a debugger and satisfactory results were obtained, with no syntactical error.

2.6.3 Wireshark

The application is running on local server on port 5000. It can be analyzed using local host loopback analysis along with a tcp.port==5000 filter. Following is a screenshot of request and response packets as the application is running.



3. Conclusion

The built model adds custom layers to successfully recognize the gender giving certain picture with high accuracy over the test data. Nevertheless, there are some limitations detected and opportunities for improvements:

Due computational resource limitation, the model was train with a subset of images. Having an appropriate machine, the model can be trained including all the images. This will make the algorithm to learn from different context of the picture giving it more experience in order to predict better never seen images.

Using different structures for the CNNs. This approach could give better performance to the mode. Although it is an expensive task, as the model can be measured on the test data set after it is trained, and this takes time and computational resources.

4. References

[1] https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

[2] D.Gowtami Annapurna, Dantu Lakshmi Pujita, Dindi Mounika "Gender Identification from Facial Features" (IJIET) http://dx.doi.org/10.21172/ijiet.154.02

[3] https://towardsdatascience.com/building-a-convolutional-neural-network-cnn-in-keras-329fbbadc5f5

[4] Age and gender estimation by using hybrid facial features | IEEE Conference Publication | IEEE Xplore

[5] AFIF4: Deep gender classification based on AdaBoost-based fusion of isolated facial features and foggy faces – ScienceDirect