

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Computer Science and Engineering

By

ABDUL RAHUMAN M

REG. NO. 40733001



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCHOOL OF
COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI –600119, TAMILNADU

APRIL-2023



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI- 600119
www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of
ABDUL RAHUMAN M (Reg. No: 40733001) who carried out the project entitled
"Age and Gender Detection" under my supervision from January 2023 to May
2023.

Internal Guide

Dr.B.U. Anu Barathi, ME, Ph.D

Head of the Department

Dr. S. Vigneshwari, M.E.,Ph.D

Submitted for Viva voce Examination held on

17-04-2023

DECLARATION

I ,**ABDUL RAHUMAN M**, hereby declare that the project report entitled “**Age and Gender Detection**” done by me under the guidance of **Dr.B.U.Anu Barathi, ME, Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE: 17-04-2023



PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean, School of Computing, Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.B.U.Anu Barathi, ME., Ph.D,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

CERTIFICATE

ABSTRACT

Automatic prediction of age and gender from face images has drawn a lot of attention recently, due to its wide applications in various facial analysis problems. However, due to the large intra-class variation of face images (such as variation in lighting, pose, scale, occlusion), the existing models are still behind the desired accuracy level, which is necessary for the use of these models in real-world applications. In this work, we propose a deep learning framework, based on the ensemble of attentional and residual convolutional networks, to predict gender and age group of facial images with high accuracy rate. Using attention mechanism enables our model to focus on the important and informative parts of the face, which can help it to make a more accurate prediction. We train our model in a multi-task learning fashion, and augment the feature embedding of the age classifier, with the predicted gender, and show that doing so can further increase the accuracy of age prediction.

Contents of Table

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	i
	LIST OF ABBREVIATIONS	ii
	LIST OF FIGURES	iii
1	Introduction	1
	1.1 Advantages of Gender Detection	3
	1.2 Disadvantages of Age and Gender Detection	4
	1.3 Python Programming	4
	1.4 Advantages of Python	6
	1.5 Disadvantages of Python	7
2	2.1 Aim	8
	2.2 Scope	8
3	3.1 Technology used	9
	3.2 Python	9
	3.3 Python libraries used	9
	3.4 Convolutional Neural Network	12
	3.5 CNN Architecture	13
	3.6 Convolutional Neural Network Design	14
	3.7 Artificial Neural Network	14
	3.8 Image Extraction	15
	3.9 Nearest Neighbor Classification	16
	3.10 Feature Extraction	16
	3.11 Computer Vision	17
	3.12 Image Processing	17
	3.13 Face Detection with Haar Cascades	18
	3.14 Landmark Detection and Face Alignment	18
	3.15 Age Recognition with CNN	18
	3.16 Gender Recognition with CNN	19
	3.17 Architecture Diagram	20
	3.18 Ideation Map	20
4	4.1 Result	21
5	5.1 Summary and Conclusion	22
	5.2 Future Work	23
	Reference	23
	Appendix Source Code Screenshots	24

LIST OF FIGURES

Figure No.	Figure Name	Page No.
1.1	The pipeline of framework for age group and gender	3
1.2	Age and Gender Detection	4
3.1	CNN	13
3.2	Image Preprocessing	17
3.3	Architecture Diagram	20
3.4	Ideation Map	20
4.1	Result	21

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
CNN	Convolutional Neural Network
DNN	Deep Neural Network
ANN	Artificial Neural Network
NLP	Natural Language Processing
CV	Computer Vision
PCA	Principal Component Analysis
NCC	Nearest Neighbor Classification
KNN	K Nearest Neighbor
GUI	Graphical User Interface
PIL	Python Imaging Library
NumPy	Numerical Python

CHAPTER – 1

INTRODUCTION

Age and gender information are very important for various real-world applications, such as social understanding, biometrics, identity verification, video surveillance, human-computer interaction, electronic customer, crowd behaviour analysis, online advertisement, item recommendation, and many more. Despite their huge applications, being able to automatically predicting age and gender from face images is a very hard problem, mainly due to the various sources of intra-class variations on the facial images of people, which makes the use of these models in real world applications limited. There are numerous works proposed for age and gender prediction in the past several years. The earlier works were mainly based on hand-crafted features extracted facial images followed by a classifier. But with the great success of deep learning models in various computer vision problems in the past decade, the more recent works on age and gender predictions are mostly shifted toward deep neural networks-based models. Here, we use deep learning framework to jointly predict the age and gender from real time situations. Given the intuition that some local regions of the face have clearer signals about the age and gender of an individual (such as beard and moustache for male, and wrinkles around eyes and mouth for age), we use an attentional convolutional network as one of our backbone models, to better attend to the salient and informative part of the face. As we can see, the model outputs are mostly sensitive to the edge patterns around facial parts, as well as wrinkles, which are important for age and gender prediction.

As predicting age and gender from faces are very related, we use a single model with multi-task learning approach to jointly predict both gender and age bucket. Also, given that knowing the gender of someone, we can better estimate her/his age, we augment the feature of the age-prediction branch with the predicted gender output. Through experimental results, we show that adding the predicted gender information to the age prediction branch, improves the model performance. To further improve the prediction accuracy of our model, we combine the prediction of attentional network with the

residual network, and use their ensemble model as the final predictor. Here are the contributions of this work:

- I propose a multi-task learning framework to jointly predict the age and gender of individuals from their face images.
- I developed an ensemble of attentional and residual networks, which outperforms both individual models. The attention layers of this model learn to focus on the most important and salient parts of the face.
- With the help of the attention mechanism, we can explain the predictions of the classifiers after they are trained, by locating the salient facial regions they are focusing on each image.

In addition, the machine learning strategies utilized by these frameworks did not completely abuse the huge quantities of image cases and information accessible through the Internet keeping in mind the end goal to enhance characterization capacities. In this paper we endeavor to close the gap between automatic face recognition abilities and those of age and gender classification techniques. To this end, we take after the fruitful sample set around late face recognition frameworks: Face recognition systems portrayed in the most recent couple of years have demonstrated that gigantic advancement can be made by the utilization of profound convolutional neural networks (CNN) . We show comparative additions with basic system engineering, composed by considering the somewhat constrained accessibility of precise age and gender classification names in existing face information sets.

We formulate the age and gender classifications task as a classification problem in which the CNN model learns to predict the age and gender from a face image. Figure 1 displays the overall idea of the proposed model.

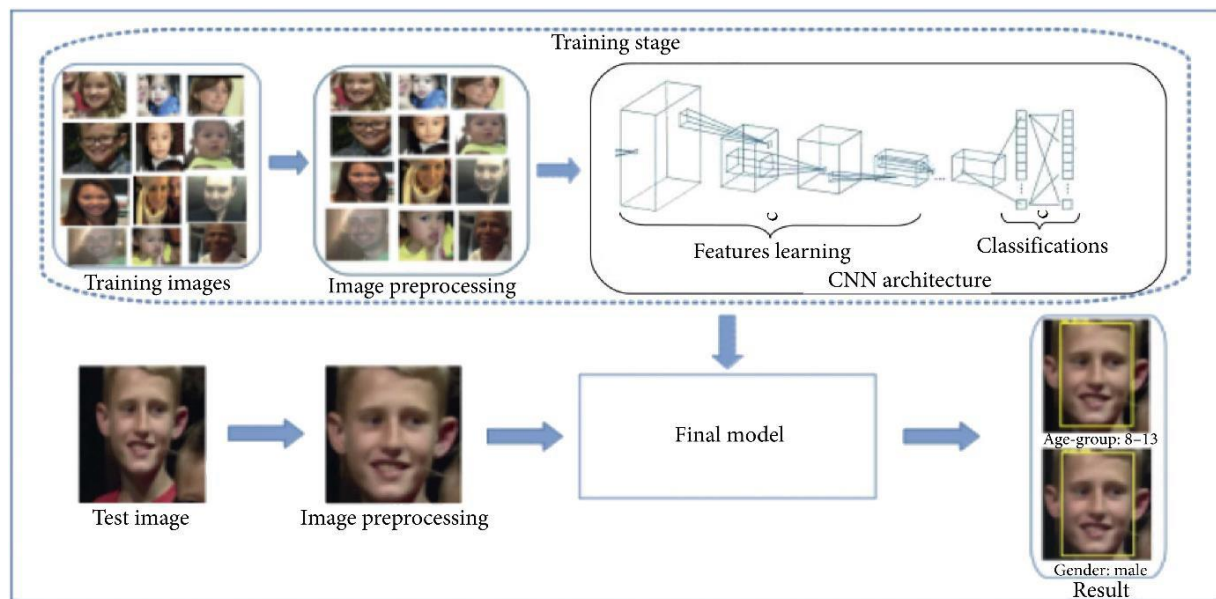


Fig 1.1

The pipeline of framework for age group and gender.

1.1 Advantages of Age and Gender Detection

Age and gender detection is essential for authentication, human-computer interaction, behavior analysis, product recommendation based on user preferences, and many other areas. Many companies needed age and gender data capture, but few solutions were available. Significant developments have been made in the past few years to meet this need. In the last decade, artificial intelligence classification systems have been used instead of manual classification systems for age and gender detection. With the introduction of artificial intelligence, the success rate of solving the problem has increased.

Age detection: Age detection can be used to place ads in the types of media most consumed by your target audience.

Gender detection: Gender detection can be used to determine whether a social media platform is more likely to show your product to men or women.



Fig 1.2

Age and Gender Detection

1.2 Disadvantages of Age and Gender Detection

Face analysis in images in the wild still pose a challenge for automatic age and gender recognition tasks, mainly due to their high variability in resolution, deformation, and occlusion. Although the performance has highly increased thanks to Convolutional Neural Networks (CNNs), it is still far from optimal when compared to other image recognition tasks, mainly because of the high sensitiveness of CNNs to facial variations.

1.3 Python Programming

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax like the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions, and classes. Other programming languages often use curly-brackets for this purpose.

Facts about Python Programming Language:

- Python is the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms.
- Python programs generally are smaller than other programming languages like Java. Programmers must type relatively less and indentation requirement of the language, makes them readable all the time.
- Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc.
- The biggest strength of Python is huge collection of standard library which can be used for the following:
 - Machine Learning
 - GUI Applications (like Kivy, Tkinter, PyQt etc.)
 - Web frameworks like Django (used by YouTube, Instagram, Dropbox)
 - Image processing (like OpenCV, Pillow)
 - Web scraping (like Scrapy, BeautifulSoup, Selenium)
 - Test frameworks
 - Multimedia
 - Scientific computing
 - Text processing and many more.

1.4 Advantages of Python Programming Language:

- Smooth learning curve
- High speed of development
- Portability and extensibility to other languages
- Versatility plus extensive toolset for almost everything

1.5 Disadvantages of Python Programming Language:

- Speed limitations
- No multithreading
- High memory consumption
- Hassles with mobile and front-end development

CHAPTER – 2

AIM AND SCOPE OF AGE AND GENDER

2.1 AIM

To build a gender and age detector that can approximately guess the gender and age of the person (face) using Deep Learning.

2.2 SCOPE

Automatic age and gender classification has become relevant to an increasing number of applications, particularly since the rise of social platforms and social media.

Nevertheless, performance of existing methods on real-world images is still significantly lacking, especially when compared to the tremendous leaps in performance recently reported for the related task of face recognition.

A convolutional neural network is a deep neural network (DNN) widely used for the purposes of image recognition and processing and NLP. Also known as a ConvNet, a CNN has input and output layers, and multiple hidden layers, many of which are convolutional. In a way, CNNs are regularized multilayer perceptron.

It is very difficult to accurately guess an exact age from a single image because of factors like makeup, lighting, obstructions, and facial expressions. And so, we make this a classification problem instead of making it one of regression.

The major applications are:

- Detect faces
- Classify into Male/Female
- Classify into one of the 8 age ranges
- Put the results on the image and display it

It will help us in employee identification to human identification, defense security and CCTV footage identification. It can be used to identify people in blurred images.

CHAPTER-3

METHODOLOGY AND ALGORITHMS USED

3.1 Technologies used:

1. OpenCV: Provide the Haar-Cascades and OpenCV library
2. Computer Vision
3. Image processing
4. CNN (Convolutional Neural Networks)

3.2 Python

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and is not specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today.

3.3 Python Libraries Used

1. Tkinter

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

- To create a tkinter app:
- Importing the module – tkinter
- Create the main window (container)
- Add any number of widgets to the main window

- Apply the event Trigger on the widgets.

Importing tkinter is same as importing any other module in the Python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x it is 'tkinter'.

You can install tkinter using the following command:

```
pip install tk
```

2. PIL

PIL stands for Python Imaging Library adds image processing capabilities to your Python interpreter.

This library provides extensive file format support, an efficient internal representation, and powerful image processing capabilities.

The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool.

3. CV2

OpenCV is a Python library that is used to solve computer vision problems. Computer vision include understanding and analyzing digital images by the computer and process the images or provide relevant data after analyzing the image.

OpenCV is an open-source library used in machine learning and image processing. It performs tasks such as recognizing handwritten digits, human faces and objects.

To use OpenCV, we need to install it.

```
pip install opencv-python
```

4. Argparse

Python argparse is a command-line parsing module that is recommended to work with the command line argument. This module was released as a part of the standard library with Python on 20th February 2011. It is like the **GETOPT MODULE** but it is slightly hard to use and requires more code lines to perform the same task. However, the argparse module is the better replacement of the **Python getopt** and **optparse module**. It provides a few important features that are given below.

- It allows us to use to positional argument.
- It allows us to customize the prefix chars.
- It supports variable numbers of parameters for a single option.
- It supports subcommands.

5. Numpy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely.

NumPy stands for Numerical Python

Why Use NumPy?

In Python we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called **ndarray**, it provides a lot of supporting functions that make working with **ndarray** very easy.

Arrays are very frequently used in data science, where speed and resources are very important.

6. Matplotlib

Matplotlib is one of the most powerful libraries in python for data visualization. **Matplotlib** is the basic visualizing or plotting library of the python programming language. **Matplotlib** is a powerful tool for executing a variety of tasks. It can create different types of visualization reports like line plots, scatter plots, histograms, bar charts, pie charts, box plots, and many more different plots. This library also supports 3- dimensional plotting.

We can install matplotlib by the following command:

```
pip install matplotlib
```

3.4 Convolutional Neural Network

A convolutional neural network, or CNN, is a deep learning neural network designed for processing structured arrays of data such as images. Convolutional neural networks are widely used in computer vision and have become the state of the art for many visual applications such as image classification, and have also found success in natural language processing for text classification.

Convolutional neural networks are very good at picking up on patterns in the input image, such as lines, gradients, circles, or even eyes and faces. It is this property that makes convolutional neural networks so powerful for computer vision. Unlike earlier computer vision algorithms, convolutional neural networks can operate directly on a raw image and do not need any pre-processing.

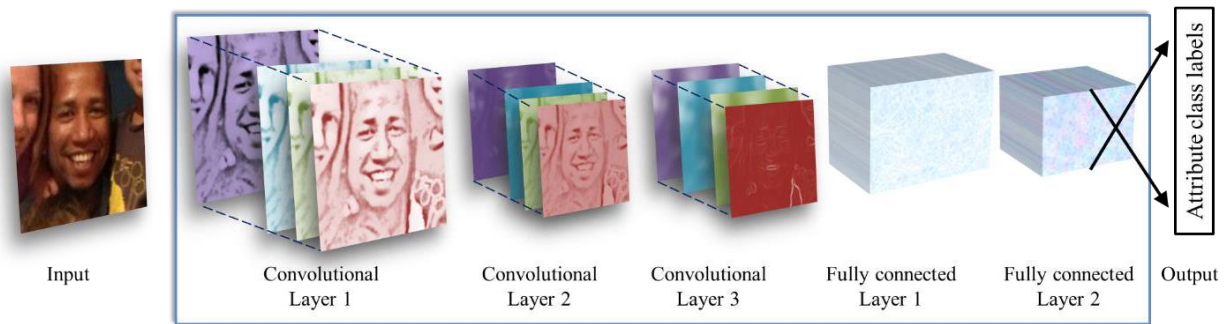


Fig 3.1

3.5 CNN Architecture

Our CNN architecture is a novel six-layer network, comprising four convolutional and two fully connected layers. The CNN design is an end-to-end sequential deep learning architecture, including feature extraction and classification phases. The feature extraction phase has four convolutional layers, with the corresponding parameters, including the number of filters, the kernel size of each filter, and the stride. It contains the convolutional layer, activation layer (rectified linear unit (ReLU)), batch normalization (instead of the conventional Local Response Normalization), max-pooling layer, and a dropout [56] (all the convolutional layers have a fixed dropout of 25%). The classification stage, on the other hand, contains two fully connected layers, that handle the classification phase of the model. The first fully connected layer contains 512 neurons, followed by a ReLU, then batch normalization, and, finally, a dropout layer at a dropout ratio of 0.5. The second and the last fully connected layer output 512 features which are densely mapped to 8 or 2 neurons for classification tasks.

A convolutional neural network is a feed-forward neural network, often with up to 20 or 30 layers. The power of a convolutional neural network comes from a special kind of layer called the convolutional layer.

Convolutional neural networks contain many convolutional layers stacked on top of each other, each one capable of recognizing more sophisticated shapes. With three or four convolutional layers it is possible to recognize handwritten digits and with 25 layers it is possible to distinguish human faces.

The usage of convolutional layers in a convolutional neural network mirrors the structure of the human visual cortex, where a series of layers process an incoming image and identify progressively more complex features.

3.6 Convolutional Neural Network Design

The architecture of a convolutional neural network is a multi-layered feed-forward neural network, made by stacking many hidden layers on top of each other in sequence. It is this sequential design that allows convolutional neural networks to learn hierarchical features.

The hidden layers are typically convolutional layers followed by activation layers, some of them followed by pooling layers.

A simple convolutional neural network that aids understanding of the core design principles is the early convolutional neural network LeNet-5, published by Yann LeCun in 1998. LeNet can recognize handwritten characters.

3.7 Artificial Neural Networks

Artificial neural networks (ANN) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labelled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge of cats, for example, that they have fur, tails, whiskers, and cat-like faces. Instead, they automatically generate identifying characteristics from the examples that they process. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it. In ANN implementations, the "signal" at a

connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times. The original goal of the ANN approach was to solve problems in the same way that a human brain would. But over time, attention moved to performing specific tasks, leading to deviations from biology. ANNs have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games, medical diagnosis, and even in activities that have traditionally been considered as reserved to humans, like painting.

3.8 Image Extraction

There are many methods have been proposed in the literature for the age estimation and gender classification. However, all of them have still disadvantage such as not complete reflection about face structure, face texture. We classified the gender and age based on the association of two methods: geometric feature-based method and Principal Component Analysis (PCA) method for improving the efficiency of facial feature extraction stage. The face database contains the 13 individual groups. Within a given database, all weight vectors of the persons within the same age group are averaged together. Experimental results show that better gender classification and age estimation. Gender classification is important visual tasks for human beings, such as many social interactions critically depend on the correct gender perception. As visual surveillance and human-computer interaction technologies evolve, computer vision systems for gender classification will play an increasing important role in our lives. Age prediction is concerned with the use of a training set to train a model that can estimate the age of the facial images. Amount once paid is not refundable or adjustable under any circumstances in future. This project contains full non editable files and database

images that we have used. The Principal Component Analysis (PCA) can do prediction, redundancy removal, feature extraction, data compression, etc. Because PCA is a classical technique which can do something in the linear domain, applications having linear models are suitable. Let us consider the PCA procedure in a training set of M face images. Let a face image be represented as a two-dimensional N by N array of intensity values, or a vector of dimension N^2 . Then PCA tends to find a M -dimensional subspace whose basis vectors correspond to the maximum variance direction in the original image space. New basis vectors define a subspace of face images called face space.

All images of known faces are projected onto the face space to find sets of weights that describe the contribution of each vector. By comparing a set of weights for the unknown face to sets of weights of known faces, the face can be identified.

3.9 Nearest Neighbor Classification

One of the most popular non-parametric techniques is the Nearest Neighbor classification (NNC). NNC asymptotic or infinite sample size error is less than twice of the Bayes error. NNC gives a trade-off between the distributions of the training data with a priori probability of the classes involved. KNN (Kth nearest neighbor classifier) classifier is easy to compute and very efficient. KNN is very compatible and obtain less memory storage. So, it has good discriminative power. Also, KNN is very robust to image distortions (e.g., rotation, illumination). So, this paper can produce good result by combining (PCA and KNN). Euclidian distance determines whether the input face is near a known face. The problem of automatic face recognition is a composite task that involves detection and location of faces in a cluttered background, normalization, recognition, and verification.

3.10 Feature Extraction

The gender classification procedure is described in this section. Features extraction deals with extracting features that are basic for differentiating one class of object from another. First, the fast and accurate facial features extraction algorithm is developed. The training positions of the specific face region are applied.

3.11 Computer Vision

Computer vision is an interdisciplinary field that deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do. "Computer vision is concerned with the automatic extraction, analysis and understanding of useful information from a single image or a sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding." As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multi- dimensional data from a medical scanner. As a technological discipline, computer vision seeks to apply its theories and models for the construction of computer vision systems.

3.12 Image Preprocessing

Intelligent age and gender classifiers tackle the classification task under unfiltered real-world settings. Most of those face images are not aligned and nonfrontal and also with different degrees of variations in pose, appearance, lighting, and background conditions. Therefore, those in-the-wild face images need first to be detected, then aligned, and, finally, used as input for the classifiers. The image preprocessing phase as shown in Figure is explained in more details below.

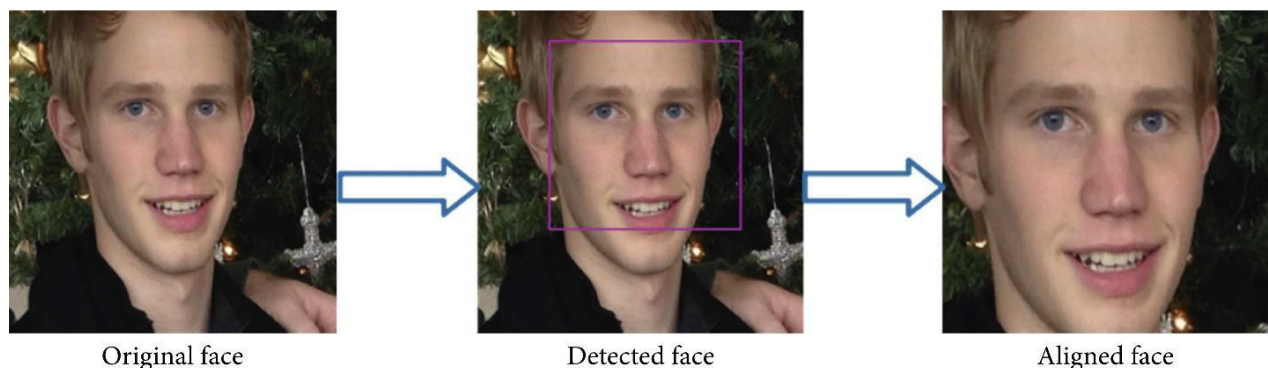


Fig 3.2

3.13 Face detection with Haar Cascades:

This is a part most of us at least have heard of. OpenCV provide direct methods to import Haar-cascades and use them to detect faces. I will not be explaining this part in deep. You guys can refer to my previous article to know more about face detection using OpenCV

3.14 Landmark Detection and Face Alignment

After face detection, is the facial landmark detection and face alignment phase, where we employ the state-of-the-art solution. This image pre-processing solution is an open-source Multiview facial landmark detection algorithm that uses five landmark detection models, including a frontal model, two half-profile models, and two full profile models. All these five models are trained to work on one of the corresponding facial poses. The face alignment phase, on the other hand, entails running of all the five facial landmark models, on the detected faces. An affine transformation is then performed on the model, with the highest confidence score, to the predefined optimal settings of those landmarks.

3.15 Age Recognition with CNN

This is almost similar to the gender detection part except that the corresponding prototxt file and the caffe model file are “deploy_agenet.prototxt” and “age_net.caffemodel”. Furthermore, the CNN’s output layer (probability layer) in this CNN consists of 8 values for 8 age classes (“0–3”, “3–7”, “7–14”, “14–22”, “22– 35”, “35–44”, “44–57” and “57-100”)

A caffemodel has 2 associated files:

1. **prototxt**—The definition of CNN goes in here. This file defines the layers in the neural network, each layer’s inputs, outputs, and functionality.
2. **Caffemodel**—This contains the information of the trained neural network (trained model)

3.16 Gender Recognition with CNN:

Gender recognition using OpenCV's fisherfaces implementation is quite popular and some of you may have tried or read about it also. But, in this example, I will be using a different approach to recognize gender. This method was introduced by two Israel researchers, Gil Levi and Tal Hassner in 2015. I have used the CNN models trained by them in this example. We are going to use the OpenCV's dnn package which stands for "Deep Neural Networks."

In the dnn package, OpenCV has provided a class called Net which can be used to populate a neural network. Furthermore, these packages support importing neural network models from well-known deep learning frameworks like caffe, tensorflow and torch. The researchers I had mentioned above have published their CNN models as caffe models. Therefore, we will be using the CaffeImporter import that model into our application.

Steps for practicing gender and age detection

- face_detector.pbtxt
- face_detector.pb
- age.prototxt
- age.caffemodel
- gender.prototxt
- gender.caffemodel

3.17 Architecture Diagram

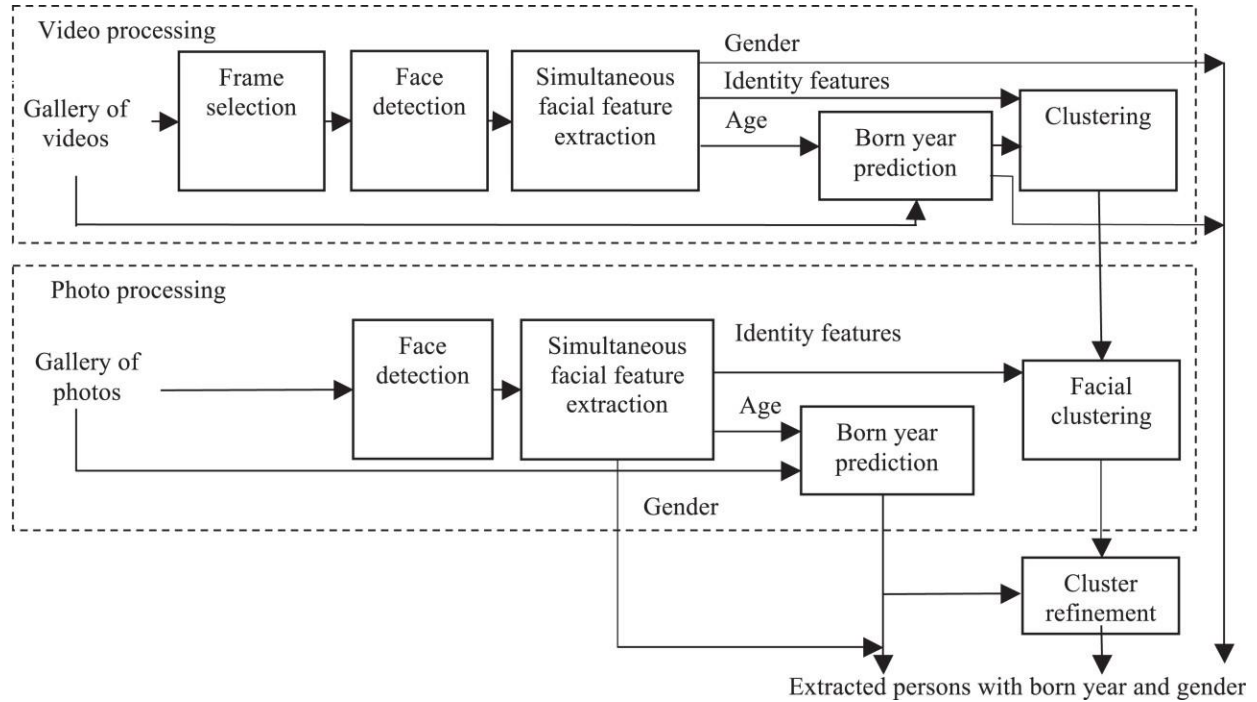


Fig 3.3

3.18 Ideation Map

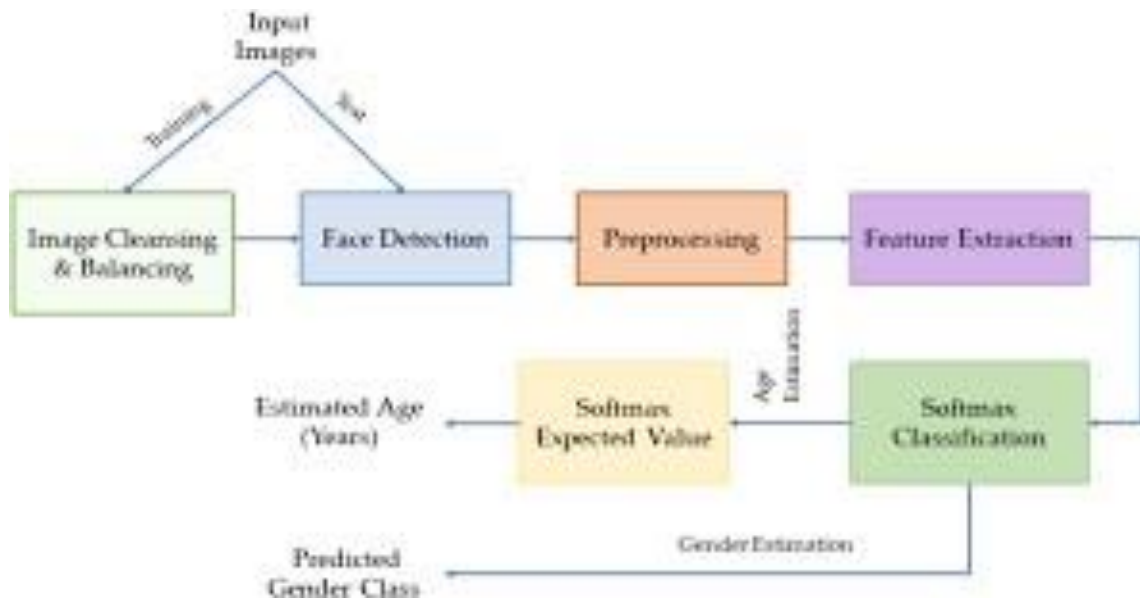


Fig 3.4

CHAPTER – 4

4.1 Results:

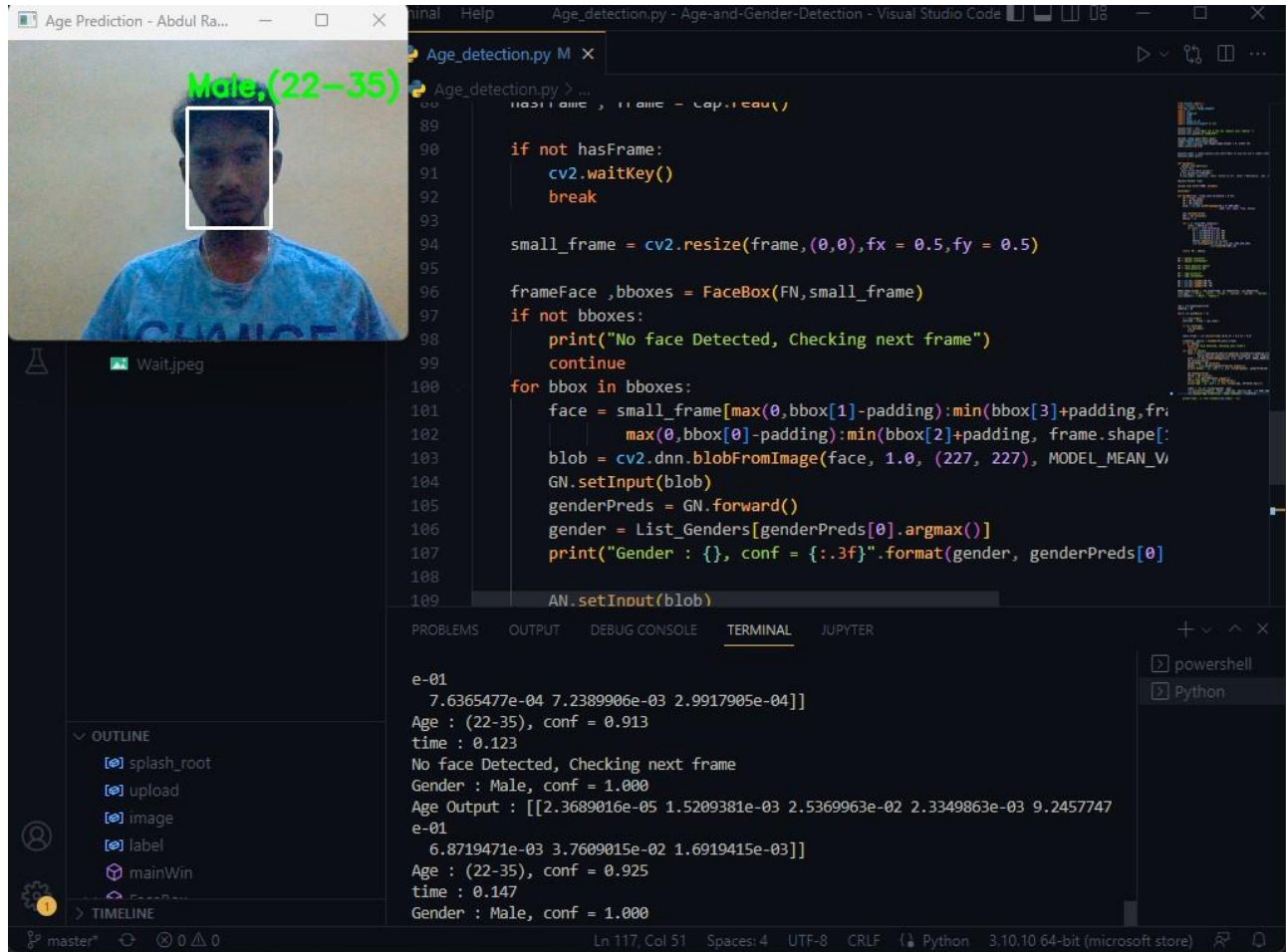


Fig 4.1

```
>python3 Age_detection.py
```

Gender: Male
Age:17-22 years

CHAPTER – 5

5.1 SUMMARY AND CONCLUSION

In this work, it is concluded that age and gender research has been the focus of the last few years. Even though many of the strategies of the past focused on issues of age and sexuality, not so long ago, this work certainly focuses on the compelling images taken in laboratory settings. Such settings do not adequately reflect the general appearance types of current reality photos on social networking sites and online archives. Web images, anytime, are not just about how complex they are: they are equally saturated.

Easy access to great collections of high-quality video readings of a learning machine with ongoing preparation information. CNN can be used to provide effects of age and age order, not by looking at the smallest size of the uneducated image of age and sexuality. Finally, I hope that more training material will be found with work age and gender cohesion that will allow effective techniques from other forms of big data sets to be used this place. We hope you found this paper well read and useful in your quest.

Taking illustration from the related issue of face acknowledgment, we investigate how well profound CNN perform on these assignments utilizing Internet information. We provide results with an incline profound learning architecture designed to keep away from over fitting because of the impediment of constrained marked information. Our system is "shallow" contrasted with a portion of the late system designs, along these lines diminishing the quantity of its parameters and the chance for over fitting. We advance swell the extent of the preparation information by falsely including trimmed variants of the images in our preparation set. The subsequent framework was tried on the Audience benchmark of unfiltered images and appeared to fundamentally beat late cutting edge. Two critical conclusions can be produced using our experimental outcomes. In the first place, CNN can be utilized to give enhanced age and gender arrangement results, notwithstanding considering the much little size of contemporary unconstrained image sets named for age and gender classification. Second, the straight forwardness of our model suggests that more involved frameworks utilizing more preparing information might well be able to do significantly enhance.

5.2 Future Works

When changing a dataset, the same model can be trained to predict the feelings of race etc. Age and gender classifications can be used to predict age and gender in uncontrolled real-time situations such as train stations, banks, buses, airports, etc. For example, depending on the number of male and female passengers by the age on the train station, toilets and restrooms can be built to facilitate transportation.

References

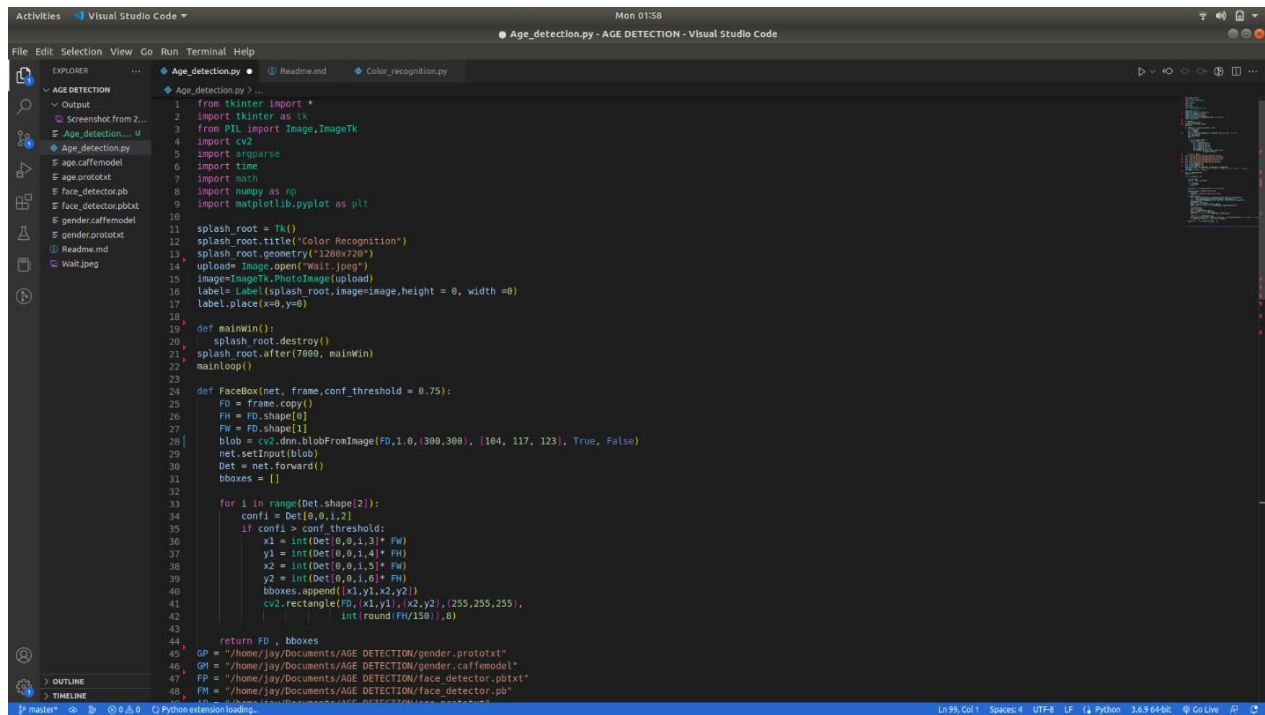
1. A. Kumar and F. Shaik, "Importance of Image Processing", 2016.
2. Chenjing Yan, Congyan Lang, Tao Wang, Xuetao Du, and Chen Zhang, "Age Estimation Based on Convolutional Neural Network", 2014 Springer International Publishing Switzerland.
3. Hang Qi and Liqing Zhang, "Age Classification System with ICA Based Local Facial Features", 2009 Springer-Verlag Berlin Heidelberg.
4. Eran Eidinger, Roee Enbar, and Tal Hassner, "Age and Gender Estimation of Unfiltered Faces", 2014 IEEE.
5. Aditya K. Saxena, Shweta Sharma and Vijay K. Chaurasiya, "Neural Network based Human Age-group Estimation in Curvelet Domain", 2015 The Authors Published by Elsevier.
6. J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao. Wld: "A robust local image descriptor", Trans.Pattern Anal. Mach. Intell.,
7. S. E. Choi, Y. J. Lee, S. J. Lee, K. R. Park, and J. Kim. "Age estimation using a hierarchical classifier based on global and local facial features", Pattern Recognition, 44(6):1262–1281, 2011
8. T. F. Cootes, G. J. Edwards, and C. J. Taylor. "Active appearance models", In European Conf. Comput. Vision, pages 484–498. Springer, 1998.

Appendix

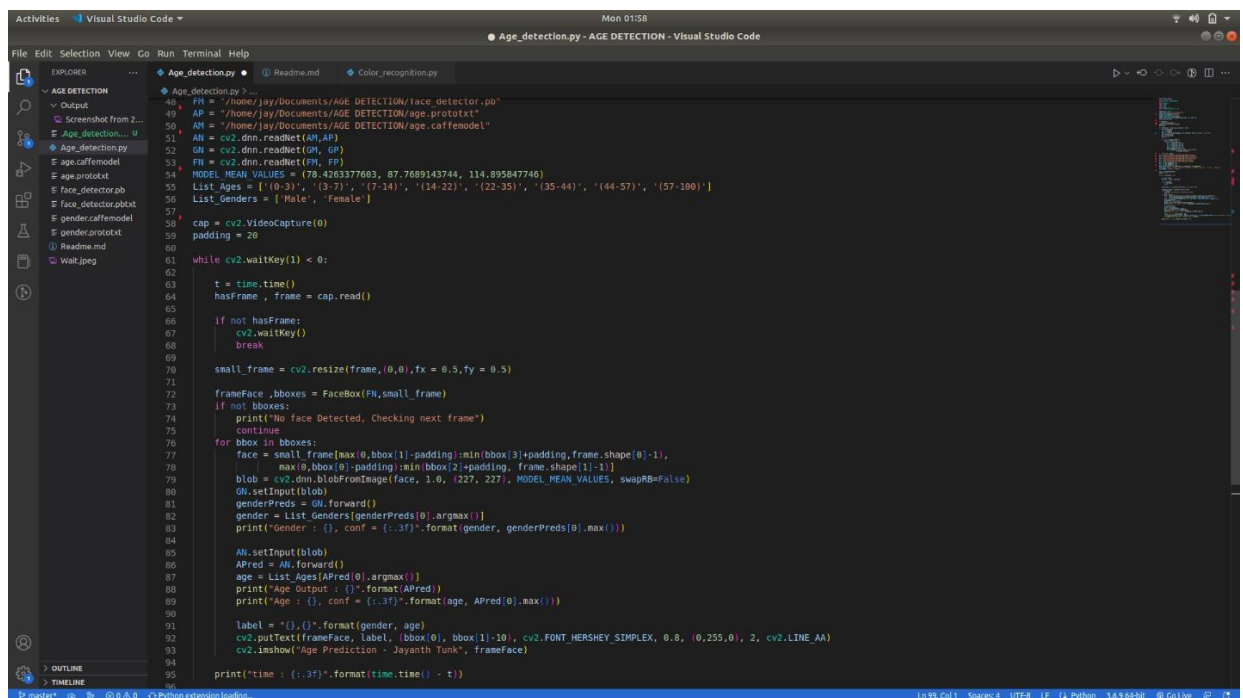
Source Code:

Screenshots:

Python File, Prototxt file and pbtxt files



```
1 from tkinter import *
2 import tkinter as tk
3 from PIL import Image, ImageTk
4 import cv2
5 import argparse
6 import time
7 import math
8 import numpy as np
9 import matplotlib.pyplot as plt
10
11 splash_root = Tk()
12 splash_root.title("Color Recognition")
13 splash_root.geometry("1280x720")
14 upload = Image.open("wait.jpeg")
15 image = ImageTk.PhotoImage(upload)
16 label = Label(splash_root, image=image, height = 0, width = 0)
17 label.place(x=0, y=0)
18
19 def mainwin():
20     splash_root.destroy()
21     splash_root.after(7000, mainwin)
22     mainloop()
23
24 def FaceBox(net, frame_conf_threshold = 0.75):
25     FD = frame.copy()
26     FH = FD.shape[0]
27     FW = FD.shape[1]
28     blob = cv2.dnn.blobFromImage(FD, 1.0, (300, 300), [104, 117, 123], True, False)
29     net.setInput(blob)
30     Det = net.forward()
31     bboxes = []
32
33     for i in range(Det.shape[2]):
34         conf1 = Det[0,0,i,2]
35         if conf1 > conf_threshold:
36             x1 = int(Det[0,0,i,3]* FW)
37             y1 = int(Det[0,0,i,4]* FH)
38             x2 = int(Det[0,0,i,5]* FW)
39             y2 = int(Det[0,0,i,6]* FH)
40             bboxes.append([x1,y1,x2,y2])
41             cv2.rectangle(FD, (x1,y1), (x2,y2), (255,255,255),
42                           int(round(FH/150)), 0)
43
44     return FD, bboxes
45
46 GP = "/home/jay/Documents/AGE DETECTION/gender.prototxt"
47 GM = "/home/jay/Documents/AGE DETECTION/gender.caffemodel"
48 FP = "/home/jay/Documents/AGE DETECTION/face_detector.prototxt"
49 FM = "/home/jay/Documents/AGE DETECTION/face_detector.pb"
```



```
49 FM = "/home/jay/Documents/AGE DETECTION/face_detector.pb"
50 AP = "/home/jay/Documents/AGE DETECTION/age.prototxt"
51 AM = "/home/jay/Documents/AGE DETECTION/age.caffemodel"
52 AN = cv2.dnn.readNet(GP, GM)
53 FN = cv2.dnn.readNet(FP, FM)
54 MODEL_MEAN_VALUES = (78.4263377603, 76.769143744, 114.895847746)
55 List_Ages = ['(0-3)', '(3-7)', '(7-14)', '(14-22)', '(22-35)', '(35-44)', '(44-57)', '(57-100)']
56 List_Genders = ['Male', 'Female']
57
58 cap = cv2.VideoCapture(0)
59 padding = 20
60
61 while cv2.waitKey(1) < 0:
62
63     t = time.time()
64     hasFrame, frame = cap.read()
65
66     if not hasFrame:
67         cv2.waitKey()
68         break
69
70     small_frame = cv2.resize(frame, (0,0), fx = 0.5, fy = 0.5)
71
72     frameFace, bboxes = FaceBox(FN, small_frame)
73     if not bboxes:
74         print("No face Detected, Checking next frame")
75         continue
76     for bbox in bboxes:
77         face = small_frame[max(0, bbox[1]-padding):min(bbox[3]+padding, frame.shape[0]-1),
78                             max(0, bbox[0]-padding):min(bbox[2]+padding, frame.shape[1]-1)]
79         blob = cv2.dnn.blobFromImage(face, 1.0, (227, 227), MODEL_MEAN_VALUES, swapRB=False)
80         GN.setInput(blob)
81         genderPreds = GN.forward()
82         gender = List_Genders[genderPreds[0].argmax()]
83         print("gender : {}, conf = {:.3f}".format(gender, genderPreds[0].max()))
84
85         AN.setInput(blob)
86         APred = AN.forward()
87         age = List_Ages[APred[0].argmax()]
88         print("Age Output : {}".format(APred))
89         print("Age : {}, conf = {:.3f}".format(age, APred[0].max()))
90
91         label = "{}({})".format(gender, age)
92         cv2.putText(frameFace, label, (bbox[0], bbox[1]-10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,0), 2, cv2.LINE_AA)
93         cv2.imshow("Age Prediction - Jayanth Tunk", frameFace)
94
95     print("time : {:.3f}".format(time.time() - t))
```

Visual Studio Code - AGE DETECTION - Visual Studio Code

age.prototxt

```
1 name: "CaffeNet"
2 input: "data"
3 input_dim: 1
4 input_dim: 3
5 input_dim: 227
6 input_dim: 227
7 layers {
8   name: "conv1"
9   type: CONVOLUTION
10  bottom: "data"
11  top: "conv1"
12  convolution_param {
13    num_output: 96
14    kernel_size: 7
15    stride: 4
16  }
17 }
18 layers {
19   name: "relu1"
20   type: RELU
21   bottom: "conv1"
22   top: "conv1"
23 }
24 layers {
25   name: "pool1"
26   type: POOLING
27   bottom: "conv1"
28   top: "pool1"
29   pooling_param {
30     pool: MAX
31     kernel_size: 3
32     stride: 2
33   }
34 }
35 layers {
36   name: "norm1"
37   type: LRN
38   bottom: "pool1"
39   top: "norm1"
40   lrn_param {
41     local_size: 5
42     alpha: 0.0001
43     beta: 0.75
44   }
45 }
46 layers {
47   name: "conv2"
48   type: CONVOLUTION
```

Visual Studio Code - AGE DETECTION - Visual Studio Code

age.prototxt

```
130 type: DROPOUT
131 bottom: "fc6"
132 top: "fc6"
133 dropout_param {
134   dropout_ratio: 0.5
135 }
136 }
137 layers {
138   name: "fc7"
139   type: INNER_PRODUCT
140   bottom: "fc6"
141   top: "fc7"
142   inner_product_param {
143     num_output: 512
144   }
145 }
146 layers {
147   name: "relu7"
148   type: RELU
149   bottom: "fc7"
150   top: "fc7"
151 }
152 layers {
153   name: "drop7"
154   type: DROPOUT
155   bottom: "fc7"
156   top: "fc7"
157   dropout_param {
158     dropout_ratio: 0.5
159   }
160 }
161 layers {
162   name: "fc8"
163   type: INNER_PRODUCT
164   bottom: "fc7"
165   top: "fc8"
166   inner_product_param {
167     num_output: 8
168   }
169 }
170 layers {
171   name: "prob"
172   type: SOFTMAX
173   bottom: "fc8"
174   top: "prob"
175 }
176 }
```

This screenshot shows the first part of a TensorFlow Lite graph in the file `face_detector.pbtxt`. The graph starts with a `data` node, followed by a `FusedBatchNorm` operation, then a `Mul` operation, a `BiasAdd` operation, and finally a `SpaceToBatchND` operation. The `SpaceToBatchND` node has attributes for `value`, `dtype` (`DT_INT32`), and `tensor_shape` with dimensions `[1, 1, 1, 1]`.

```
1 node {
2   name: "data"
3   op: "Placeholder"
4   attr {
5     key: "dtype"
6     value {
7       type: DT_FLOAT
8     }
9   }
10 }
11 node {
12   name: "data_bn/FusedBatchNorm"
13   op: "FusedBatchNorm"
14   input: "data:0"
15   input: "data_bn/gamma"
16   input: "data_bn/beta"
17   input: "data_bn/mean"
18   input: "data_bn/std"
19   attr {
20     key: "epsilon"
21     value {
22       f: 1.000999996416e-05
23     }
24   }
25 }
26 node {
27   name: "data_scale/Mul"
28   op: "Mul"
29   input: "data_bn/FusedBatchNorm"
30   input: "data_scale/mul"
31 }
32 node {
33   name: "data_scale/BiasAdd"
34   op: "BiasAdd"
35   input: "data_scale/Mul"
36   input: "data_scale/add"
37 }
38 node {
39   name: "SpaceToBatchND/block_shape"
40   op: "Const"
41   attr {
42     key: "value"
43     value {
44       tensor {
45         dtype: DT_INT32
46         tensor_shape {
47           dim {
48             size: 1
49           }
50         }
51       }
52     }
53   }
54 }
```

This screenshot shows the second part of the TensorFlow Lite graph in `face_detector.pbtxt`. It continues with a `Const` node for `max_threshold`, followed by a `Const` node for `num_classes`, a `Const` node for `share_location`, a `Const` node for `top_k`, and a `Const` node for `reshape_before_softmax`. The `reshape_before_softmax` node has attributes for `value`, `dtype` (`DT_INT32`), `tensor_shape` with dimensions `[1, 1, 1, 1]`, and `int_val` values `0`, `-1`, and `2`.

```
2317   key: "max_threshold"
2318   value {
2319     f: 0.449999988079
2320   }
2321 }
2322 attr {
2323   key: "num_classes"
2324   value {
2325     i: 2
2326   }
2327 }
2328 attr {
2329   key: "share_location"
2330   value {
2331     b: true
2332   }
2333 }
2334 attr {
2335   key: "top_k"
2336   value {
2337     i: 400
2338   }
2339 }
2340 node {
2341   name: "reshape_before_softmax"
2342   op: "Const"
2343   attr {
2344     key: "value"
2345     value {
2346       tensor {
2347         dtype: DT_INT32
2348         tensor_shape {
2349           dim {
2350             size: 3
2351           }
2352         }
2353         int_val: 0
2354         int_val: -1
2355         int_val: 2
2356       }
2357     }
2358   }
2359 }
2360 }
2361 library {
2362 }
2363 }
```

```
1 name: "CaffeNet"
2 input: "data"
3 input_dim: 18
4 input_dim: 3
5 input_dim: 227
6 input_dim: 227
7 layers {
8   name: "conv1"
9   type: CONVOLUTION
10  bottom: "data"
11  top: "conv1"
12  convolution_param {
13    num_output: 96
14    kernel_size: 7
15    stride: 4
16  }
17 }
18 layers {
19   name: "relu1"
20   type: RELU
21   bottom: "conv1"
22   top: "conv1"
23 }
24 layers {
25   name: "pool1"
26   type: POOLING
27   bottom: "conv1"
28   top: "pool1"
29   pooling_param {
30     pool: MAX
31     kernel_size: 3
32     stride: 2
33   }
34 }
35 layers {
36   name: "norm1"
37   type: LRN
38   bottom: "pool1"
39   top: "norm1"
40   lrn_param {
41     local_size: 5
42     alpha: 0.0001
43     beta: 0.75
44   }
45 }
46 layers {
47   name: "conv2"
```

```
130 type: DROPOUT
131 bottom: "fc6"
132 top: "fc6"
133 dropout_param {
134   dropout_ratio: 0.5
135 }
136 }
137 layers {
138   name: "fc7"
139   type: INNER_PRODUCT
140   bottom: "fc6"
141   top: "fc7"
142   inner_product_param {
143     num_output: 512
144   }
145 }
146 layers {
147   name: "relu7"
148   type: RELU
149   bottom: "fc7"
150   top: "fc7"
151 }
152 layers {
153   name: "drop7"
154   type: DROPOUT
155   bottom: "fc7"
156   top: "fc7"
157   dropout_param {
158     dropout_ratio: 0.5
159   }
160 }
161 layers {
162   name: "fc8"
163   type: INNER_PRODUCT
164   bottom: "fc7"
165   top: "fc8"
166   inner_product_param {
167     num_output: 2
168   }
169 }
170 layers {
171   name: "prob"
172   type: SOFTMAX
173   bottom: "fc8"
174   top: "prob"
175 }
176 }
```