<Title of the Project>

A Project Report submitted to Chaitanya Deemed to be University in partial fulfillment of minimum academic requirements for IV Year II Semester

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

Submitted By

NAME1 (HTNO)
NAME2 (HTNO)
NAME2 (HTNO)

Under the guidance of

Prof. K SATHISH KUMAR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

CHAITANYA DEEMED TO BE UNIVERSITY

HANAMKONDA, WARANGAL -506001

(2024-2025)

FACULTY OF ENGINEERING

HANAMKONDA, WARANGAL-506001

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the major project report entitled **"Title of the project"** is being submitted by NAME1 (HTNO), NAME2 (HTNO) in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in "**Computer Science and Engineering**" at the **Chaitanya deemed to be University** during the academic year 2024-2025.

Name and Signature of the Guide
(Prof. K. Sathish Kumar)

Name and Signature of the HOD
(Dr. E. Aravind, Assoc. Prof)

Dean, Faculty of Engineering
(Prof. G. SHANKARLINGAM)

Signature of the External examiner with Date

ACKNOWLEDGEMENT

The success accomplished in this project would not have been possible, by timely help and guidance rendered by many people, we wish to express our sincere and heartfelt gratitude to all those who have helped and guided us for the completion ofthe project.

We sincerely extend thanks to our project guide **Prof. K Sathish Kumar** Department of Computer Science and Engineering for giving us moral support, kind attention and valuable guidance to us throughout this training work. We extent our heartfelt thanks to **Dr. E. Aravind, Head,** the Department of Computer Science and Engineering for his encouragement during the progress of this major project work. We would like to express our deep sense of gratitude to **Prof. G. SHANKARLINGAM**, **Dean**, Faculty of engineering, for providing the required facilities in the college campus.

We would also thank all the staff of department of electronics and communication engineering who has helped us directly or indirectly for the successful completion of the project.

Finally, we express our sincere thanks & gratitude to our family members & friends for their constant encouragement and moral support, which made the project successful.

NAME1 (HTNO)

NAME2 (HTNO)

DECLARATION

We here submit that the project report entitled **"TITLE OF THE PROJECT"** is an original work done at the **Faculty of Engineering, Hanamkonda** under the valuable guidance of **Prof. K SATHISH KUMAR, Department of Computer Science and Engineering,** in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering. We hereby declare that this project report bears no resemblance to any other reports submitted at the Faculty of Engineering**,** Chaitanya deemed to be a University for the award of the degree.

Name1 (HTNO)

Name2 (HTNO)

ABSTRACT

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

Now a days, social media apps are getting famous day by day. Age and gender play fundamental roles in social interactions. There are some interesting features like predicting gender and age in camera option. Despite the simple roles those attributes play in our everyday lives, the capacity to robotically estimate them appropriately and reliably from face snap shots continues to be far from assembly the desires of industrial applications. This is especially complicated whilst thinking about current claims to super-human talents within side the associated assignment of face recognition. The DNN's CNN algorithm helps to predict gender and age from the input face of the social media app. CNN helps to get a good accuracy to predict the age and gender classification.

1.1 Rationale

Now a day, the technology is increasing day by day people often using social media apps to get entertainment. People are expecting more features from social media apps. Adding gender and age prediction makes really cool. This project has a scope of using it at allowing people to movie theatres only who are above eighteen and also predicting male or female in their respective hostel etc.

1.2 Goal

In this study, we focus on mainly three things:

- Recognizing face by using open cv face recognition.
- Predicting gender and age using trained caffe model with the help of CNN's layers.
- Generating the result and providing it on the command prompt.

1.3 Objective

The objective of our work is to:

- Providing an easy interactive interface.
- Detecting the face from the image.

- Giving a good accuracy by predicting.
- Predicting the exact age and gender from its classification list.

1.4 Methodology

Machine learning

The space of Machine Learning manages the plan of projects that can take in rules from information, adjust to changes, and improve execution with experience. As well as being one of the underlying dreams of Computer Science, Machine Learning has gotten vital as PCs are relied upon to take care of progressively complex issues and become more coordinated into our everyday lives. Composing a PC program is a bit like recording directions for a very strict kid who incidentally turns out to be a great many occasions quicker than you.

It is one of the quickest developing spaces of software engineering, with extensive applications. The point of this reading material is to present ML, and the algorithmic ideal models it offers, in a principled way. The book gives a broad hypothetical record of the central thoughts hidden ML and the numerical determinations that change these standards into viable calculations. Following a show of the nuts and bolts of the field, the book covers a wide cluster of focal subjects that have not been tended to by past reading material. These incorporate a conversation of the computational intricacy of learning and the ideas of convexity and security; significant algorithmic ideal models including stochastic inclination plunge, neural organizations, and organized yield learning.

Anyway an enormous number of the issues we presently need PCs to address are no longer tasks we understand how to explicitly prompt a PC how to do. These fuse perceiving faces in imagines, independent driving in the desert, finding pertinent reports in an informational index (or throwing out pointless ones, for instance, spam email), finding plans in tremendous volumes of legitimate data, and changing internal limits of structures to redesign execution. That is, we may ourselves be satisfactory at recognizing people in photographs, yet we don't have even the remotest clue how to directly prompt a PC how to do it. Taking everything into account, procedures that take named getting ready data (pictures named by who is in them, or email messages set apart by whether they are spam) and a while later take in fitting standards from the data, have all the earmarks of being the most ideal approaches to manage handling these issues. Besides, we need structures that can change in accordance with developing conditions, that can

be not difficult to use by acclimating to necessities of their individual customers, and that can improve execution as time goes on.

Machine Learning Theory is both a principal hypothesis with numerous fundamental and convincing primary inquiries, and a subject of reasonable significance that assists with propelling the state of the workmanship in programming by giving numerical systems to planning new machine learning calculations. It is an energizing time for the field, as associations with numerous different regions are being found and investigated, and as new ML applications bring new inquiries to be displayed and considered. It is protected to say that the capability of Machine Learning furthermore, its hypothesis lie past the boondocks of our creative mind.

1.4.1 Data Collection

The dataset we used for this project is Adience dataset. This dataset is available in Kaggle.com. This dataset is more helpful when dealing with face related projects mainly. This dataset contains 19,487 images of 2,284 subjects with 8 age groups: 0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53 and 60+.Most age bunches have around one to 2,000 pictures with the exception of two senior gatherings (just around 800 pictures each), and the 25-32 gathering (around 5,000 pictures).

1.4.2 Data Preprocessing

Our data is in image format, which is not understandable by our system. So we need to process our data before sending it to the model. As our system can just understand numeric format, we should change our data from image to numeric values.

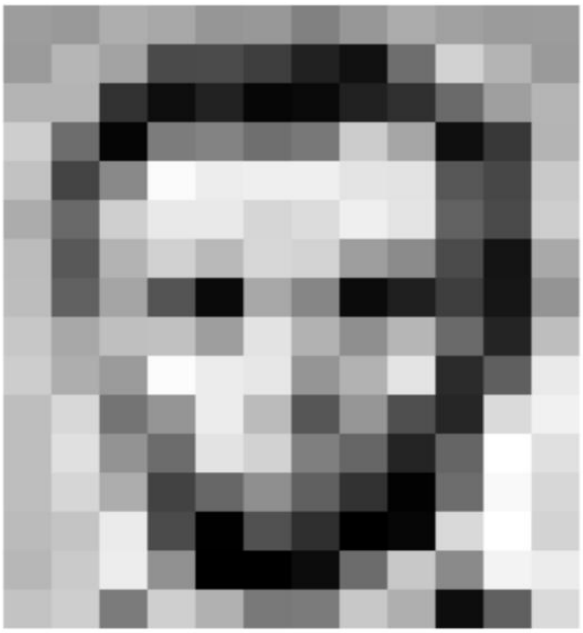
Initially, we start with cleaning the dataset images from the audience dataset and store them in a format that can be used by Caffe model. Here caffe is a train model. Weuse a Python script that will handle both image pre-processing and storage.we choose a CNN architecture and we define its parameters in a configuration file with extension .prototxt. The solver is responsible for model optimization. We define the solver parameters in a configuration file with extension .prototxt. We train the model by executing one Caffe command from the terminal. After training the model, we will get the trained model in a file with extension caffe model.

1.4.3 Algorithms

In this project we used DNN's CNN Algorithm. CNN's algorithm contains some important layers that to predict gender and age.

1.4.4 Computer Vision

Computer vision is a field of man-made reasoning that trains computers to decipher and comprehend the visual world. Utilizing advanced pictures from cameras and recordings and profound learning models, machines can precisely recognize and characterize objects — and afterward respond to what they "see". xImages are separated into pixels, which are viewed as the components of the image or the littlest unit of data that make up the image. Computer vision isn't just about changing over an image into pixels and afterward attempting to sort out what's in the image through those pixels. You need to comprehend the master plan of how to remove data from those pixels and decipher what they address.



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Deep Neural Network

Deep Learning is a field inside ML and Artificial insight (A.I.) that arrangements with calculations motivated from a human mind to help machines with knowledge without unequivocal programming. A worked on adaptation of Deep Neural Network is addressed as a various leveled

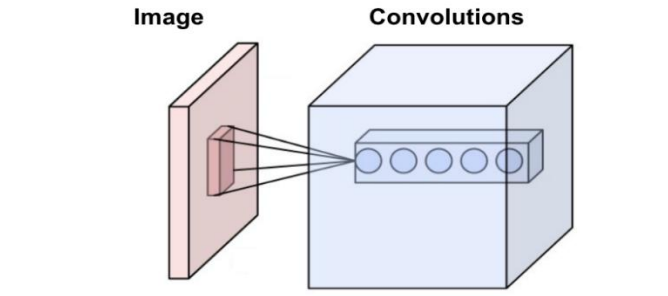
(layered) association of neurons (like the neurons in the mind) with associations with different neurons. These neurons pass a message or sign to different neurons dependent on the got info and structure a perplexing organization that learns with some criticism component.

Convulution Neural Network

There are some Convolutional neural networks are a special type of feed-forward networks. These models are designed to emulate the behaviour of a visual cortex. CNNs perform very well on visual recognition tasks. CNN's have special layers called Rectified Linear Unit and convolutional layers and pooling layers that allow the network to encode certain images properties.

Convulution Layer

This layer consists of a set of learnable filters that we slide over the image spatially, computing dot products between the entries of the filter and the input image. The filters should extend to the full depth of the input image. For example, if we want to apply a filter of size 5x5 to a colored image of size 32x32, then the filter should have depth 3 (5x5x3) to cover all 3 color channels (Red, Green, Blue) of the image. These filters will activate when they see same specific structure in the images.



Neurons of a convolutional layer, connected to their receptive field
(source: Wikipedia)

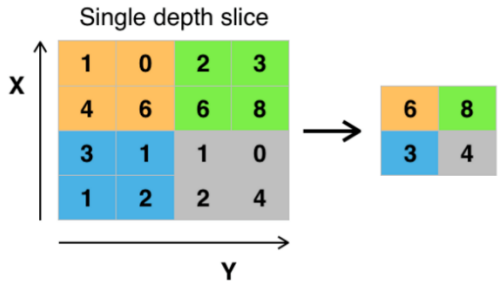
ReLu Layer

The ReLU layer applies the function f(x) = max(0, x) to all of the values in the input volume. In

basic terms, this layer just changes all the negative activations to 0. This layer increases the nonlinear properties of the model and the overall network without affecting the receptive fields of the conv layer.

Pooling Layer

Pooling is a form of non-linear down-sampling. The goal of the pooling layer is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. There are several functions to implement pooling among which max pooling is the most common one. Pooling is often applied with filters of size 2x2 applied with a stride of 2 at every depth slice. A pooling layer of size 2x2 with stride of 2 shrinks the input image to a 1/4 of its original size.



Max pooling with a 2x2 filter and stride = 2
(source: Wikipedia)

CNN's Architecture

The convolutional neural network for this python project has 3 convolutional layers:

Convolutional layer; 96 nodes, kernel size 7

Convolutional layer; 384 nodes, kernel size 3

Convolutional layer; 256 nodes, kernel size 5

It has 2 fully connected layers, each with 512 nodes, and a final output layer of softmax type.

To go about the python project, we'll:

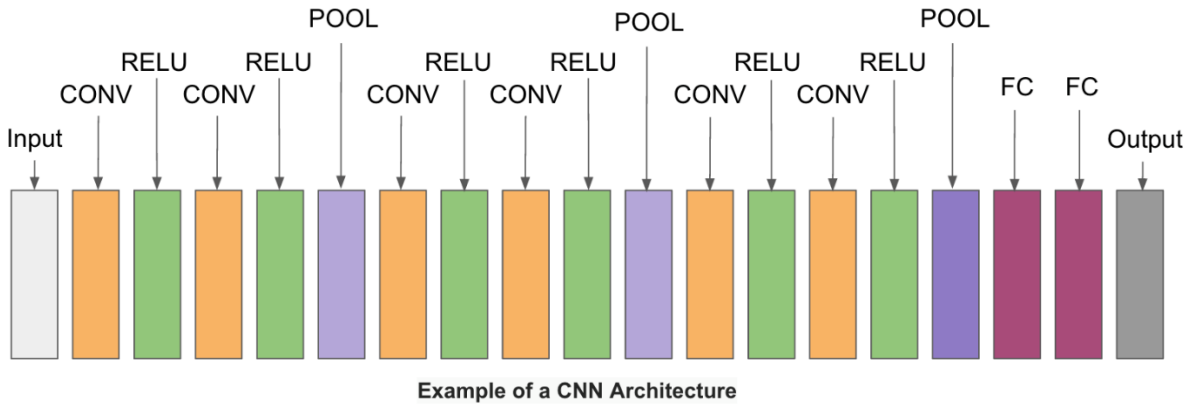
- Detect faces
- Classify into Male/Female

- Classify into one of the 8 age ranges
- Put the results on the image and display it

The easiest engineering of a convolutional neural organizations begins with an information layer (pictures) trailed by a succession of convolutional layers and pooling layers, and finishes with completely associated layers. The convolutional layers are normally trailed by one layer of ReLU initiation capacities.

The convolutional, pooling and ReLU layers act as learnable features extractors, while fully connected layers acts as a machine learning classifier. Anyhow, the early layers of the network encode generic patterns of the images, while later layers encode the details patterns of the images.

Note that only the convolutional layers and fully-connected layers have weights. These weights are learned in the training phase.



Example of a CNN Architecture

Open-CV

OpenCV is short for Open Source Computer Vision. Intuitively by the name, it is an open-source Computer Vision and Machine Learning library. This library is capable of processing real-time image and video while also boasting analytical capabilities. It supports the Deep Learning frameworks tensorflow, Caffe, and PyTorch.

Python

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its

language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Methodology

We used caffe model which Is the trained model that to predict gender and age for the given input image. caffe model contains CNN's layers which is the part of machine learning. For detecting the face from the image used open CV. Open CV provides a real-time optimized Computer Vision library, tools, and hardware. It also supports model execution for Machine Learning.

Initially we start with the initiating the caffe model and then we input the image. After completion of input then highlighting the face from the image. Predicting the gender and age from the highlighting face.

1.5 Roles and Responsibilities

Role	Name	Responsibilities
Data processing and ML	Name1/Name2/Name3	• Preparing train model • Face detection • Machine Learning • Age model
ML and Data processing	Name1/Name2/Name3	• Gender model • Machine Learning • Testing • Documentation
Tester	Name1/Name2/Name3	• UML Diagrams • Testing • Documentation • Face detection

1.6 Contribution of Project

Market potential

The probability of using smart phones by people is high in number. Many people are using social media apps for their own purpose. Having a feature like predicting people gender and their respective age is an interesting thing for them. people who use social media apps like Instagram, snap chat are actively using this feature.

Improving more updates on this feature would increases the usage of it. As we know there are many existing models on detecting age and gender but those are not upto the mark. Many scientists agrees CNN has more accuracy to detect age and gender.

Innovativeness

The idea behind implementing this project that many existing models are available that to predict age and gender. But those are not upto the mark to detect or predict. CNN algorithm is a algorithm that predicts gender and age more accurately.

Usefulness

This project is used to predict the age and gender of a person based on the image that user inputs. This project has a scope of using it at allowing people to movie theatres only who are above eighteen and also predicting male or female in their respective hostel etc. According to a prediction, in this growing technology, it has proven that usage of social media apps increases and as well as increases updates on existing features and non-existing features. So this project is like a base for advanced generation**.**

1.7 Report Organization

The remaining section of the report is structured as follows:

- **Chapter 2** Provides detailed business and technical requirements
- **Chapter 3** Provides analysis and design of this project
- **Chapter 4** Provides Construction, implementation details of this project
- **Chapter 5** Provides Conclusion and future scope

CHAPTER 2
SYSTEM STUDY

2.1 Existing System

Overview: Existing systems for Kidney prediction typically utilize traditional statistical methods and basic machine learning algorithms. These systems rely on clinical data collected from patients to assess the risk of developing Kidney and monitor disease progression.

Key Features:

- **Data Utilization:** Most existing systems use a limited set of clinical parameters, such as serum creatinine levels, blood pressure, age, and urine analysis, to predict Kidney.
- **Predictive Models:** Traditional models like Logistic Regression, Decision Trees, and basic Neural Networks are commonly employed. These models may not effectively capture the complex relationships within the data.
- **Diagnostic Tools:** Many existing systems primarily function as diagnostic tools to identify Kidney at advanced stages rather than providing early predictions.

Strengths:

- **Simplicity:** Existing systems are often straightforward and easy to implement, making them accessible for healthcare providers with limited technical expertise.
- **Cost-Effective:** Basic predictive models require less computational power and can be deployed without significant investment in advanced technology.

Limitations:

- **Limited Accuracy:** Existing models may have lower predictive accuracy due to their reliance on a small number of features and inability to capture non-linear relationships.
- **Lack of Personalization:** Traditional systems often do not account for individual patient characteristics, limiting their effectiveness in diverse populations.

Static Models: Most existing systems do not adapt to new data, leading to outdated predictions as medical knowledge evolves.

2.2 Proposed System

Overview: The proposed system aims to enhance CKD prediction by leveraging advanced machine learning algorithms and integrating a more comprehensive set of patient data. This system seeks to improve predictive accuracy, provide personalized insights, and facilitate timely interventions.

Key Features:

- **Comprehensive Data Input:** The proposed system will utilize a wide range of features, including demographic data, laboratory results, medical history, lifestyle factors, and genetic information.
- **Advanced Machine Learning Algorithms:** Algorithms such as Random Forest, Support Vector Machines (SVM), Gradient Boosting Machines (GBM), and Deep Learning models (e.g., Neural Networks) will be employed to improve predictive performance.
- **Real-Time Monitoring and Feedback:** The system will allow for real-time data input and continuous monitoring of patient conditions, enabling timely updates to predictions and interventions.
- **User-Friendly Interface:** A web or mobile application interface will be developed to make the system accessible to healthcare providers and patients, with visualization tools for better understanding.

Strengths:

- **Increased Accuracy:** By utilizing advanced algorithms and a wider range of features, the proposed system aims to achieve higher accuracy in CKD prediction compared to existing systems.
- **Personalized Insights:** The system will consider individual patient characteristics, improving the relevance of predictions and recommendations.
- **Adaptability:** The proposed system can be updated with new data and algorithms, ensuring that it remains current with the latest medical research and guidelines.

Limitations:

- **Complexity:** The implementation of advanced ML algorithms may require more technical expertise and resources than traditional systems.
- **Data Privacy Concerns:** The integration of sensitive patient data necessitates robust security measures to protect patient privacy and comply with regulations.

Dependency on Data Quality: The effectiveness of the proposed system heavily relies on the quality and completeness of the data collected.

CHAPTER-3
REQUIREMENT ENGINEERING

3.1 Functional Requirement

The functional requirements describe the core functionality of the application. It tells about the input and output behaviors of application or system.

Interface Requirement:

- Command Prompt page 1
- User gives image name as input in command prompt
- A dialog box displayed as output page 2
- And displays results on Command prompt

3.2 Non Functional Requirement

Non functional requirements are those requirements of the system which are not directly concerned with specific functionality delivered by the system. They may be related to emergent properties such as reliability, usability etc.

- To provide maximum accuracy.
- Provide visualized analysis.
- Ease of use.
- Availability
- Reliability
- Maintainability

CHAPTER-4

ANALYSIS AND DESIGN

4.1 Use case diagram

Use case diagram shows a relationship between actors and use cases. Here actors specify a role played by a user and the use case specifies a function that an actor should be performed. An actor can be a user or a part of the system.

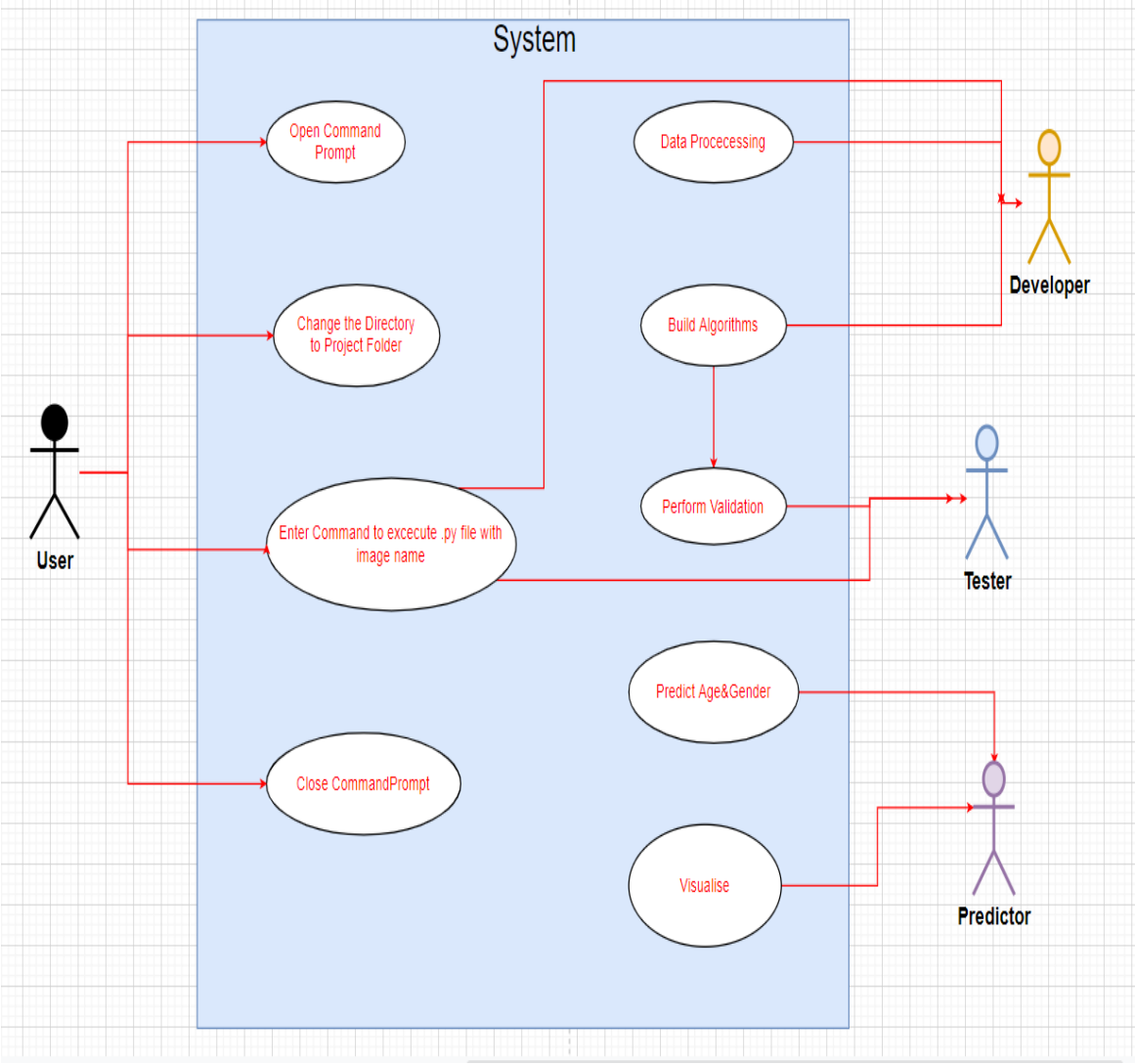


Fig 4.1-Use case diagram

4.2 Sequence diagram:

In the sequence diagram how the object interacts with the other object is shown. There are sequence of events that are represented.It is a period situated perspective on the association between items to achieve a conduct objective of the framework.

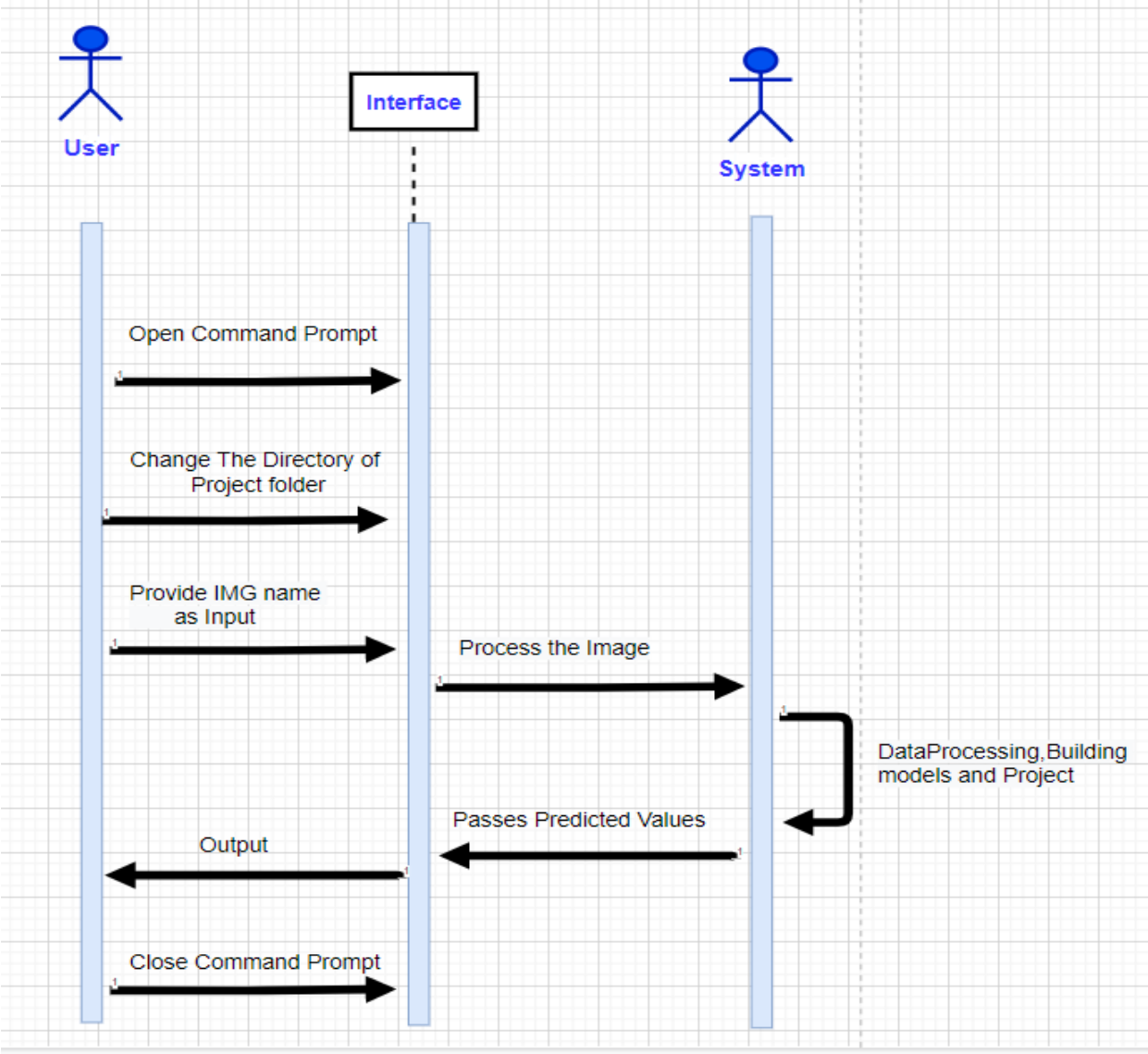


Fig 4.2-Sequencediagram

4.3 Activity diagram

The activity diagram is a graphical representation for representing the flow of interaction within specific scenarios. It is similar to a flowchart in which various activities that can be performed in the system are represented.

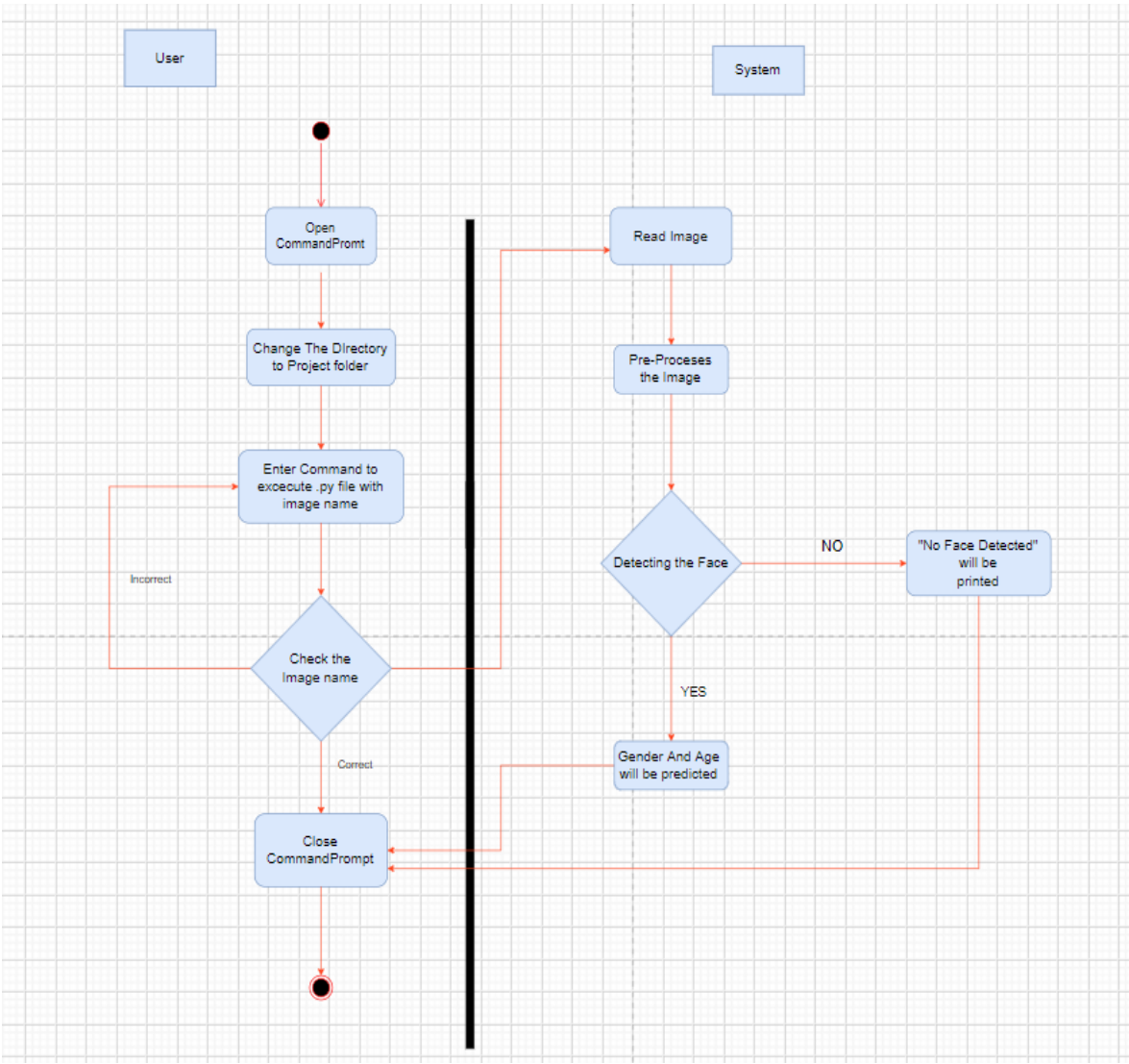


Fig 4.3- Activity diagram

4.4 System architecture

The system architectural design is the design process for identifying the subsystems making up the system and framework for subsystem control and communication. The goal of the architectural design is to establish the overall structure of the software system.

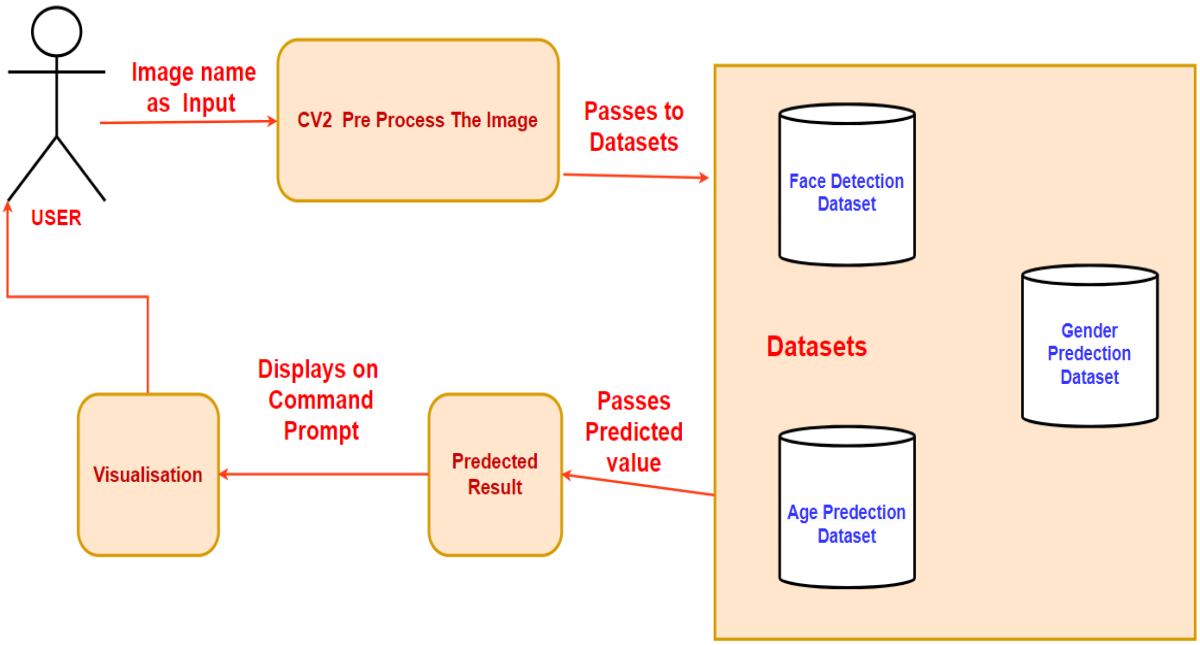


Fig 4.4- System Architecture

CHAPTER-5

CONSTRUCTION

5.1 System Requirements

5.1.1 Software Details

- Microsoft visual studio code
- Python 3.0 or above
- Open-CV
- Argparse

5.1.2 Hardware Details

- Operating system: Windows 7 or newer, or Linux.
- System architecture: 64-bit x86, 32-bit x86 with Windows or Linux.
- CPU: Intel Core 2 Quad CPU Q6600 @ 2.40GHz or greater.
- Ram above 2gb
- Rom above 500mb

CHAPTER 6
IMPLEMENTATION

6.1 Building Application

We have divided our project into some modules. In each module we perform some tasks and after completing all modules we integrate all into one.

- **Importing libraries**

  Importing required libraries and packages which are cv2,math,argparseetc important to call different methods.

- **Tools**

  Install python 3.8

  Install argparse

  Install open cv

- **Importing Dataset**

  We use audience dataset for this project. It contains 26580 photos of 2284 subjects.

- **Building Caffe model**

  A module is built for detecting gender and age.

6.1.1 Sample Code

detect.py

```
import cv2
import math
importargparse
defhighlightFace(net, frame, conf_threshold=0.7):
frameOpencvDnn=frame.copy()
frameHeight=frameOpencvDnn.shape[0]
frameWidth=frameOpencvDnn.shape[1]
blob=cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)
net.setInput(blob)
detections=net.forward()
fBoxes=[]
fori in range(detections.shape[2]):
confidence=detections[0,0,i,2]
if confidence>conf_threshold:
x1=int(detections[0,0,i,3]*frameWidth)
```

```
        y1=int(detections[0,0,i,4]*frameHeight)
x2=int(detections[0,0,i,5]*frameWidth)
        y2=int(detections[0,0,i,6]*frameHeight)
fBoxes.append([x1,y1,x2,y2])
cv2.rectangle(frameOpencvDnn, (x1,y1), (x2,y2), (0,255,0), int(round(frameHeight/150)), 8)
returnframeOpencvDnn,fBoxes


parser=argparse.ArgumentParser()
parser.add_argument('--picture')
args=parser.parse_args()
agePrototxt="age_deploy.prototxt"
ageModel1="age_net.caffemodel"
genderPrototxt="gender_deploy.prototxt"
genderModel1="gender_net.caffemodel"
facePrototxt="opencv_face_detector.pbtxt"
faceModel1="opencv_face_detector_uint8.pb"
MODEL_MEAN_VALUES=(78.4263377603, 87.7689143744, 114.895847746)
ageList=['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)', '(38-43)', '(48-53)', '(60-100)']
genderList=['Male','Female']
face_Net=cv2.dnn.readNet(faceModel1,facePrototxt)
age_Net=cv2.dnn.readNet(ageModel1,agePrototxt)
gender_Net=cv2.dnn.readNet(genderModel1,genderPrototxt)
vid=cv2.VideoCapture(args.image if args.image else 0)
padding0=20
while cv2.waitKey(1)<0 :
Framing,frame=vid.read()
if not Framing:
cv2.waitKey()
break
resultImg,fBoxes=highlightFace(face_Net,frame)
if not fBoxes:
print("No face detected")
forfaceBox in fBoxes:
face=frame[max(0,faceBox[1]-padding0):
            min(faceBox[3]+padding0,frame.shape[0]-1),max(0,faceBox[0]-padding0)
:min(faceBox[2]+padding0, frame.shape[1]-1)]

blob=cv2.dnn.blobFromImage(face,1.0,(227,227),MODEL_MEAN_VALUES, swapRB=False)
gender_Net.setInput(blob)
genderPreds=gender_Net.forward()
gender=genderList[genderPreds[0].argmax()]
print(f'Gender: {gender}')
age_Net.setInput(blob)
agePreds=age_Net.forward()
age=ageList[agePreds[0].argmax()]
```

```
print(f'Age: {age[1:-1]} years')
cv2.putText(resultImg, f'{gender}, {age}', (faceBox[0], faceBox[1]-10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,255), 2, cv2.LINE_AA)
cv2.imshow("Detecting age and gender", resultImg)
```

age_deploy.prototxt

```
name: "CaffeNet"
input: "data"
input_dim: 1
input_dim: 3
input_dim: 227
input_dim: 227
layers {
name: "conv1"
type: CONVOLUTION
bottom: "data"
top: "conv1"
convolution_param {
num_output: 96
kernel_size: 7
stride: 4
  }
}
layers {
name: "relu1"
type: RELU
bottom: "conv1"
top: "conv1"
}
layers {
name: "pool1"
type: POOLING
bottom: "conv1"
top: "pool1"
pooling_param {
pool: MAX
kernel_size: 3
stride: 2
  }
}
layers {
name: "norm1"
type: LRN
bottom: "pool1"
top: "norm1"
```

```
lrn_param {
local_size: 5
alpha: 0.0001
beta: 0.75
  }
}
layers {
name: "conv2"
type: CONVOLUTION
bottom: "norm1"
top: "conv2"
convolution_param {
num_output: 256
pad: 2
kernel_size: 5
  }
}
layers {
name: "relu2"
type: RELU
bottom: "conv2"
top: "conv2"
}
layers {
name: "pool2"
type: POOLING
bottom: "conv2"
top: "pool2"
pooling_param {
pool: MAX
kernel_size: 3
stride: 2
  }
}
layers {
name: "norm2"
type: LRN
bottom: "pool2"
top: "norm2"
lrn_param {
local_size: 5
alpha: 0.0001
beta: 0.75
  }
}
layers {
```

```
name: "conv3"
type: CONVOLUTION
bottom: "norm2"
top: "conv3"
convolution_param {
num_output: 384
pad: 1
kernel_size: 3
  }
}
layers{
name: "relu3"
type: RELU
bottom: "conv3"
top: "conv3"
}
layers {
name: "pool5"
type: POOLING
bottom: "conv3"
top: "pool5"
pooling_param {
pool: MAX
kernel_size: 3
stride: 2
  }
}
layers {
name: "fc6"
type: INNER_PRODUCT
bottom: "pool5"
top: "fc6"
inner_product_param {
num_output: 512
  }
}
layers {
name: "relu6"
type: RELU
bottom: "fc6"
top: "fc6"
}
layers {
name: "drop6"
type: DROPOUT
bottom: "fc6"
```

```
top: "fc6"
dropout_param {
dropout_ratio: 0.5
  }
}
layers {
name: "fc7"
type: INNER_PRODUCT
bottom: "fc6"
top: "fc7"
inner_product_param {
num_output: 512
  }
}
layers {
name: "relu7"
type: RELU
bottom: "fc7"
top: "fc7"
}
layers {
name: "drop7"
type: DROPOUT
bottom: "fc7"
top: "fc7"
dropout_param {
dropout_ratio: 0.5
  }
}
layers {
name: "fc8"
type: INNER_PRODUCT
bottom: "fc7"
top: "fc8"
inner_product_param {
num_output: 8
  }
}
layers {
name: "prob"
type: SOFTMAX
bottom: "fc8"
top: "prob"
```

gender_deploy.prototxt

```
name: "CaffeNet"
```

```
input: "data"
input_dim: 10
input_dim: 3
input_dim: 227
input_dim: 227
layers {
name: "conv1"
type: CONVOLUTION
bottom: "data"
top: "conv1"
convolution_param {
num_output: 96
kernel_size: 7
stride: 4
  }
}
layers {
name: "relu1"
type: RELU
bottom: "conv1"
top: "conv1"
}
layers {
name: "pool1"
type: POOLING
bottom: "conv1"
top: "pool1"
pooling_param {
pool: MAX
kernel_size: 3
```

```
stride: 2
  }
}
layers {
name: "norm1"
type: LRN
bottom: "pool1"
top: "norm1"
lrn_param {
local_size: 5
alpha: 0.0001
beta: 0.75
  }
}
layers {
name: "conv2"
type: CONVOLUTION
bottom: "norm1"
top: "conv2"
convolution_param {
num_output: 256
pad: 2
kernel_size: 5
  }
}
layers {
name: "relu2"
type: RELU
bottom: "conv2"
top: "conv2"
```

```
}
layers {
name: "pool2"
type: POOLING
bottom: "conv2"
top: "pool2"
pooling_param {
pool: MAX
kernel_size: 3
stride: 2
  }
}
layers {
name: "norm2"
type: LRN
bottom: "pool2"
top: "norm2"
lrn_param {
local_size: 5
alpha: 0.0001
beta: 0.75
  }
}
layers {
name: "conv3"
type: CONVOLUTION
bottom: "norm2"
top: "conv3"
convolution_param {
num_output: 384
```

```
pad: 1
kernel_size: 3
  }
}
layers{
name: "relu3"
type: RELU
bottom: "conv3"
top: "conv3"
}
layers {
name: "pool5"
type: POOLING
bottom: "conv3"
top: "pool5"
pooling_param {
pool: MAX
kernel_size: 3
stride: 2
  }
}
layers {
name: "fc6"
type: INNER_PRODUCT
bottom: "pool5"
top: "fc6"
inner_product_param {
num_output: 512
  }
}
```

```
layers {
name: "relu6"
type: RELU
bottom: "fc6"
top: "fc6"
}
layers {
name: "drop6"
type: DROPOUT
bottom: "fc6"
top: "fc6"
dropout_param {
dropout_ratio: 0.5
  }
}
layers {
name: "fc7"
type: INNER_PRODUCT
bottom: "fc6"
top: "fc7"
inner_product_param {
num_output: 512
  }}
layers {
name: "relu7"
type: RELU
bottom: "fc7"
top: "fc7"
}
layers {
```

name: "drop7"

type: DROPOUT

bottom: "fc7"

top: "fc7"

dropout_param {

dropout_ratio: 0.5

}}

layers {

name: "fc8"

type: INNER_PRODUCT

bottom: "fc7"

top: "fc8"

inner_product_param {

num_output: 2

}}

layers {

name: "prob"

type: SOFTMAX

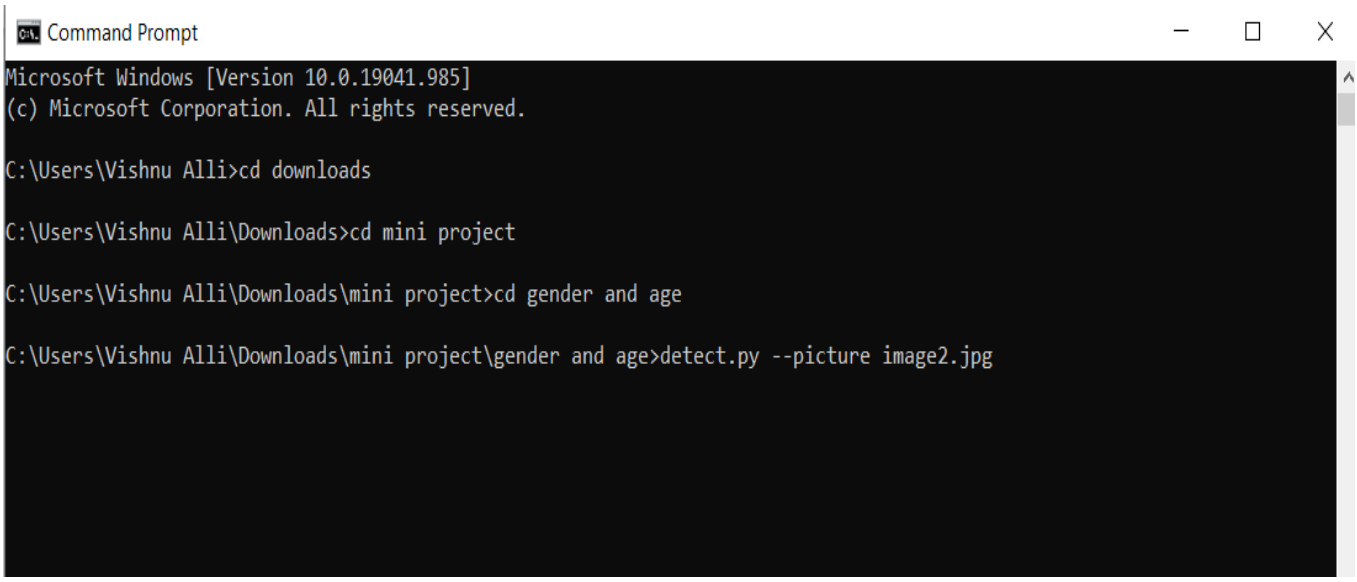
bottom: "fc8"

top: "prob"

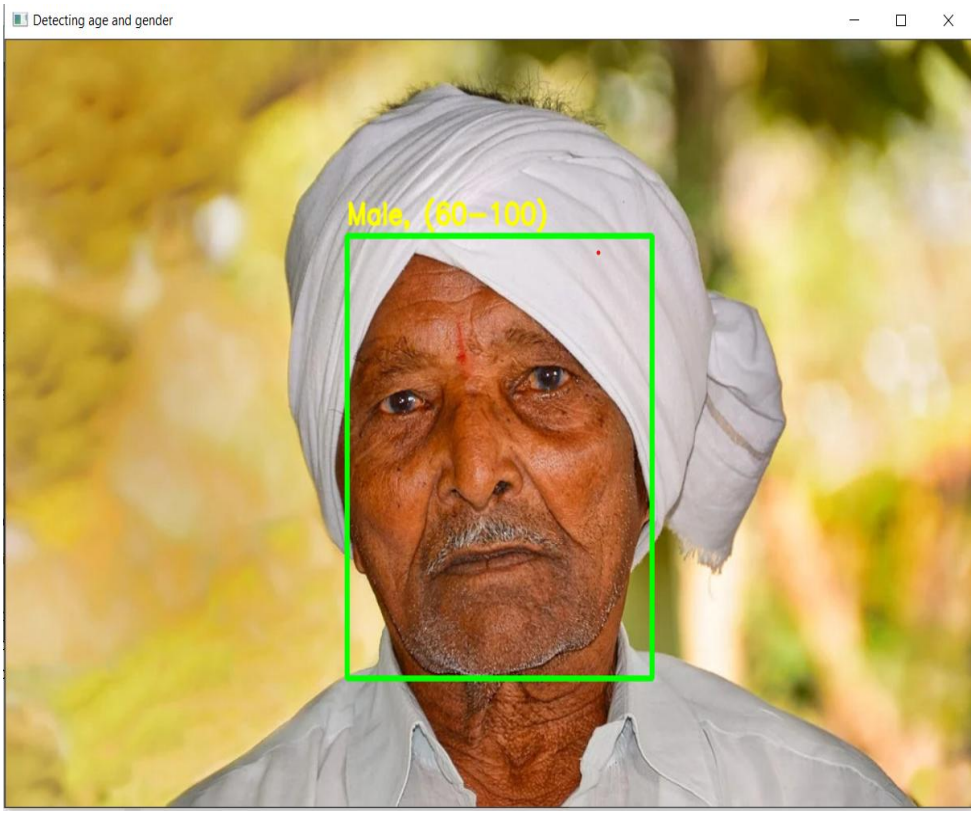
}

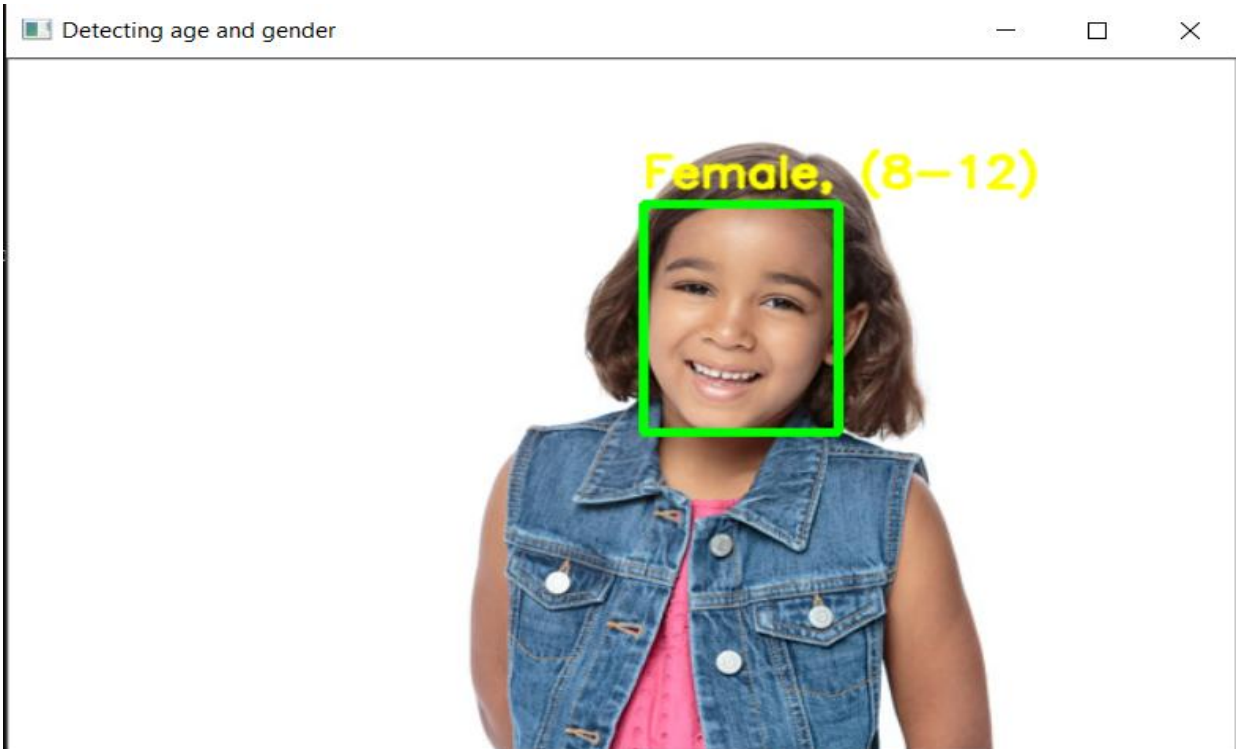
6.1.2 Screenshots
Source of the project:

1. Command Prompt

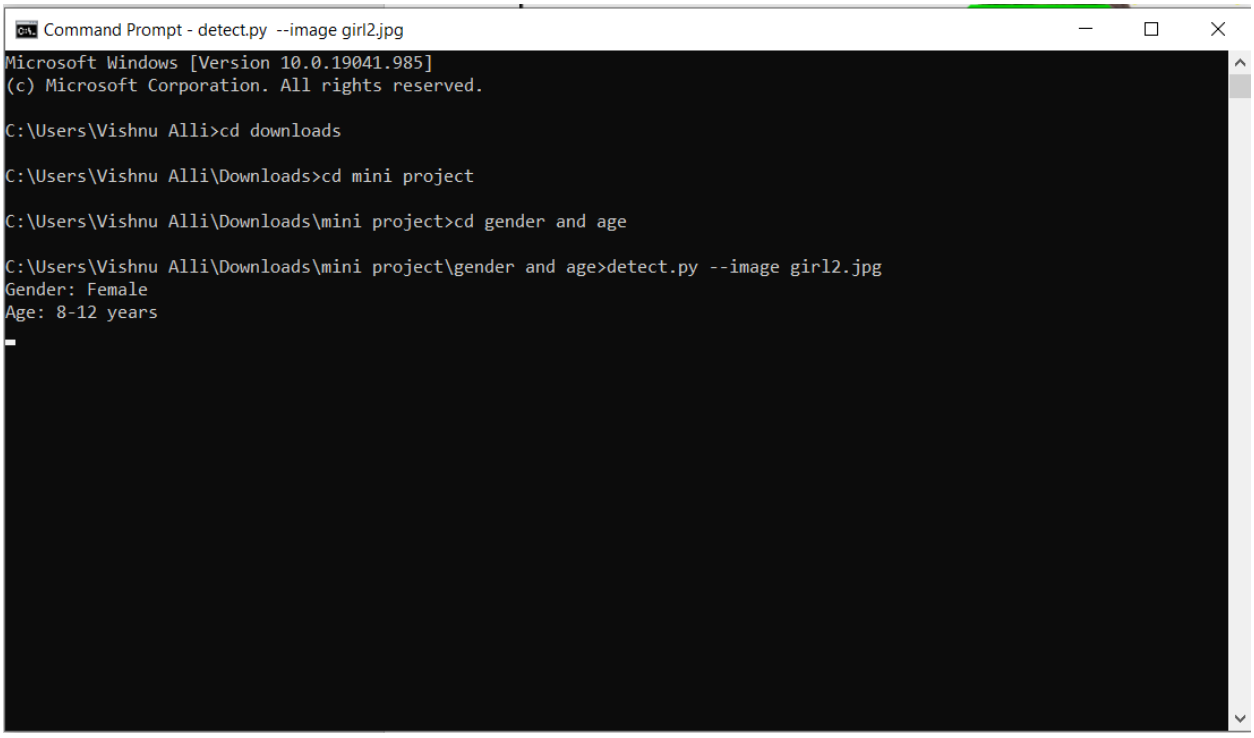


2. Displays pop up window





3. Displaying results on command prompt



6.2 Testing

Testing is meant to discover errors. Testing is the method of trying to find some possible defect or flaw in a product of work. This is a means to test the functionality of modules, subassemblies, assemblies and/or a finished product It is the software evaluation phase in order to ensure that the software system meets the specifications and user expectations and does not fail in an inappropriate manner. There are specific test forms. Every type of test satisfies a particular test requirement.

6.2.1 Types of Testing

White Box Testing

White Box Testing is a test in which the software tester has information or at least its intent about the inner workings, structure and language of the program. It is deliberate. It is used for testing areas not reachable from a black box point.

Black Box Testing

Black Box Research checks the program without any understanding of the module's inner workings, configuration or language. Black box examinations, as in most other examinations, Specifications or requirements documents, such as specifications or requirements documents, must be written from a reliable source document. This is a test in which the device under test is viewed as a black box within which you can't "touch" this. The test provides inputs and responds to outputs without considering how the software works.

6.2.2 Test cases and result

S.NO	Action	Excepted Output	Output
1	Open Command Prompt	Command Prompt window will be Open	Command Prompt window will be Open
2	Change Directory to Project Folder	Directory Changes into Project folder	Directory Changes into Project Folder
3	Detect.py –image <imageName> should be given	It takes 2-3 sec to Open a pop window For Result	It takes 2-3 sec to Open a pop window For Result
4	If face is recognized	Gender and Age will be Displayed on Command Prompt	Gender and Age will be Displayed on Command Prompt
5	If face is not recognized	“No Face Detected” will be printed on Command Prompt	“No Face Detected” will be printed on Command Prompt

CHAPTER-7

EXPERIMENTS AND CONCLUSION

7.1 Experiments and Results

Our method is implemented using the Caffe open-source framework. Training was performed on an Amazon GPU machine with 1,536 CUDA cores and 4GB of video memory. Training each network required about four hours, predicting age or gender on a single image using our network requires about 200ms. Prediction running times can conceivably be substantially improved by running the network on image batches. We test the accuracy of our CNN design using the recently released Adience benchmark, designed for age and gender classification. The entire Adience collection includes roughly 26K images of 2,284 subjects.

	0-2	**4-6**	**8-13**	**15-20**	**25-32**	**38-43**	**48-53**	**60+**	**Total**
Male	**745**	**928**	**934**	**734**	**2308**	**1294**	**392**	**442**	**8192**
Female	**682**	**1234**	**1360**	**919**	**2589**	**1056**	**433**	**427**	**9411**
Both	**1427**	**2162**	**2294**	**1653**	**4897**	**2350**	**825**	**869**	**19487**

Table 7.1. The Adiencefaces benchmark

The tables-7.2, 7.3 presents our results for gender and age classification respectively. For age classification, we measure and compare both the accuracy when the algorithm gives the exact age-group classification and when the algorithm is off by one adjacent age-group (i.e., the subject belongs to the group immediately older or immediately younger than the predicted group). This follows others who have done so in the past, and reflects the uncertainty inherent to the task – facial features often change very little between oldest faces in one age class and the youngest faces of the subsequent class.

Method	**Accuracy**
Best from E. Eidineger and T.hassner	77.8 (+ or -) 1.3
Best from T.hassner and S.Harel	79.3 (+ or -) 0.0
Proposed using single crop	85.9 (+ or -) 1.4
Proposed using over-sample	86.8 (+ or -) 1.4

Table 7.2. Gender estimation results on Adience Benchmark

Ethod	Exact	1-off
Best from T.hassner and E. Eidinger	45.1 (+ or -) 2.6	79.5 (+ or -) 1.4
Proposed using single crop	49.5 (+ or -) 4.4	84.6 (+ or -) 1.7
Proposed using over sample	50.7 (+ or -) 5.1	84.7 (+ or -) 2.2

Table 7.3 Age estimation results on benchmark

7.2 Conclusion

In spite of the fact that numerous past techniques have tended to the issues old enough and sexual orientation arrangement, up to this point, quite a bit of this work has zeroed in on obliged pictures taken in lab settings. Such settings don't sufficiently reflect appearance varieties normal to this present reality pictures in friendly sites and online stores. Web pictures, nonetheless, are not just really testing: they are likewise bountiful. The simple accessibility of enormous picture assortments furnishes present day AI based frameworks with adequately perpetual preparing information, however this information isn't generally reasonably marked for directed learning.

CNN can be used to provide improved age and gender classification results. It extracts the feature from the image accurately. CNN improves the performance of gender and age classification.This project’s accuracy is not good when group of people are in the image.Two significant ends can be produced using our outcomes. To begin with, CNN can be utilized to give improved age and gender orientation grouping results, in any event, considering the a lot more modest size of contemporary unconstrained picture sets marked for age and sex. Second, the straightforwardness of our model suggests that more intricate frameworks utilizing seriously preparing information likely could be able to do significantly improving outcomes past those announced here.

CHAPTER 8

REFERENCES

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition Trans. Pattern Anal. Mach. Intell., 28(12):2037–2041, 2006.

[2] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. Trans. on Inform. Forensics and Security, 9(12), 2014.

[3] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. Proc. Conf. Comput. Vision Pattern Recognition, 2015.

[4] S. Baluja and H. A. Rowley. Boosting sex identification performance. Int. J. Comput. Vision, 71(1):111–119, 2007.

[5] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. arXiv preprint arXiv:1405.3531, 2014