A

Project Report on

AGE AND GENDER DETECTION USING DEEP LEARNING

Submitted in partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

G Lakshmi Priya	19AK1A0579
Y Kalavathi	19AK1A0574
Y Pavan Kalyan	20AK5A0506
P Rajashekar	19AK1A05C5

Under the guidance of

Mr. N Venkatramana., MTech.
Assistant Professor

COMPUTER SCIENCE AND ENGINEERING

ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND SCIENCES (AUTONOMOUS)

(Approved by AICTE, New Delhi & Permanent Affiliation to JNTUA, Anantapur. Three B. Tech Programmes (CSE, ECE & CIVIL ENGINEERING) are accredited by NBA, New Delhi, Accredited by NAACwith 'A' Grade , Bangalore. Accredited by Institution of Engineers (India), KOLKATA. A-grade awarded by AP Knowledge Mission. Recognized under sections 2(f) & 12(B) of UGC Act 1956.) Tirupati-517520.

2019-2023

ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND SCIENCES (AUTONOMOUS)

(Approved by AICTE, New Delhi & Permanent Affiliation to JNTUA, Anantapur. Three B. Tech Programmes (CSE, ECE & CIVIL ENGINEERING) are accredited by NBA, New Delhi, Accredited by NAACwith 'A' Grade , Bangalore. Accredited by Institution of Engineers (India), KOLKATA. A-grade awarded by AP Knowledge Mission. Recognized under sections 2(f) & 12(B) of UGC Act 1956.) Tirupati-517520.

2019-2023

COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that this is a bonafide record of the Project Report entitled, ***"Age And Gender Detection using Deep Learning",*** ***done by*** *G Lakshmi Priya(19AK1A0579), Y Kalavathi(19AK1A0574),Y Pavan Kalyan(20K5A0506),P Rajashekar(19AK1A05C5)* ***is being submitted in partial fulfilment of the requirements for the award of the degree of*** ***BACHELOR OF TECHNOLOGY*** *in* ***COMPUTER SCIENCE AND ENGINEERING*** *to the Annamacharya Institute of technology and Sciences, Tirupati, during the academic year* ***2022-2023.***

Signature of the supervisor	Signature of Head of the Department
Mr N. Venkatramana, MTech.	**Mr. B. Ramana Reddy, M.Tech.,(Ph.D)**
Assistant professor,	Associate professor and HOD
Department of CSE,	Department of CSE,
AITS, Tirupati	AITS, Tirupati

DATE:

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND SCIENCES (AUTONOMOUS)

(Approved by AICTE, New Delhi & Permanent Affiliation to JNTUA, Anantapur.
Three B. Tech Programmes (CSE, ECE & CIVIL ENGINEERING) are accredited by NBA, New Delhi, Accredited by NAAC with'A' Grade , Bangalore. Accredited by Institution of Engineers (India), KOLKATA. A-grade awarded by AP Knowledge Mission. Recognized under sections 2(f) & 12(B) of UGC Act 1956.) Tirupati- 517520.

2019-2023

COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the project titled **"Age And Gender Detection using Deep Learning"** is a genuine project work carried out by us, in B.Tech (Computer Science and Engineering) course in Annamacharya Institute of Technology And Sciences and has not been submitted to any other course or university for the award of our degree by us.

G Lakshmi Priya	**19AK1A0579**
Y Kalavathi	**19AK1A0574**
Y Pavan Kalyan	**20AK5A0506**
P Rajashekar	**19AK1A05C5**

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We avail this opportunity to express our deep sense of gratitude and hearty thanks to **Dr.C. GANGI REDDY,** Hon'ble Secretary of AITS-Tirupati, for providing congenial atmosphere and encouragement.

We show gratitude to **Dr. C. NADHAMUNI REDDY**, **Principal** for having provided all the facilities and support.

We would like to thank **Mr. B. RAMANA REDDY., M.Tech(Ph.D), Associate professor & HOD**, **Computer Science and Engineering** for encouragement at various levels of our Project.

We thankful to our project coordinator **Mr. V.SAMBASIVA, Assistant Professor, CSE,** for his sustained inspiring guidance and cooperation throughout the process of this project.

We would like to express our sincere gratitude to our supervisor **Mr. N. Venkatramana, Assistant Professor**, Dept. of CSE, AITS, for his constant help, kind cooperation and encouragement in completing the work successfully.

We express our deep sense of gratitude and thanks to all the **Teaching** and **Non-Teaching Staff** of our college who stood with us during the project and helped us to make it a successful venture.

We place highest regards to our **Parents**, **Friends** and **Well wishers** who helped a lot in making the report of this project.

G Lakshmi Priya	**19AK1A0579**
Y Kalavathi	**19AK1A0574**
Y Pavan Kalyan	**20AK5A0506**
P Rajashekar	**19AK1A05C5**

CONTENTS

APPENDIX A

S.NO	PUBLICATIONS
1.	International Conference Certificates
2.	Journal Paper
3.	Journal Certificates

ABSTRACT

A Human face shows various emotions, expressions, and many more. Here we are going to determine the age of a human person through his/her face. This age estimation from the face is a challenging problem because of many internal factors, such as gender and race, and external factors, such as environments and lifestyles. Here we are going to build a gender and age detector that can guess the gender and age of the person(face) in real-time using a webcam by Deep learning methodologies like CNN and Open-CV.

Automatic prediction of age and gender from face images has drawn a lot of attention recently, due it is wide applications in various facial analysis problems. However, due to the large intraclass 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. Our model is trained on a popular face age and gender dataset, and achieved promising results. Through visualization of the attention maps of the train model, we show that our model has learned to become sensitive to the right regions of the face In this project,we are going to use Deep Learning to accurately identify the gender and age of a personfrom the image of a face.

The predicted gender may be one of ‘Male’ and ‘female’ and the predicted age is in the range of (0 – 3), (4 – 7), (8 – 12), (13 – 18), (19 -24), (25 – 32), (33 – 43), (44 – 53), (54 – 60), (61 – 100). Here we will use the Audience dataset, It has a total of 26,580 photos of 2,284 subjects in eight age ranges as Mentioned above. It is difficult to accurately guess an exact age from a single image because of factors like makeup, lightning, obstructions, and facial-expression.

LIST OF FIGURES

LIST OF TABLES

LIST OF SCREENS

LIST OF ABBRERVIATIONS

Short Form	Abbreviation
AI	Artificial intelligence
RELU	Rectified linear activation function
LBPH	Local binary pattern histograms
ANN	Artificial Neural Network
CNN	Convolutional Neural Networks
DBN	Deep belief network
DNN	Deep neural network
DL	Deep learning
PCA	Principal component analysis
LDA	Linear Discriminant Analysis
HOG	Histograms of oriented gradients
OpenCV	Open Source Computer Vision Library
CSV	Comma-separated values
k-NN	Kth neural network

1. INTRODUCTION

1.1 Introduction

Age and gender information are especially important for various real-world applications, such as social understanding, biometrics, identity verification, video surveillance, human computer interaction, electronic customer, crowd behavior analysis, online advertisement, item recommendation, and many more. Despite their huge applications, being able to automatically predicting age and gender from face images is an extremely hard problem, dueto 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 based on hand-crafted features extracted facial images followed by a classifier. But with the remarkable success of deep learning models in various computer vision problems in the past decade [1]– [5], the more recent works on age and gender predictions are mostly shifted toward deep neural networks-based models. In this work, we propose a deep learningframework to jointly predict the age and gender from face images. Given the intuition that some local regions of the face have more clear 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. Figure 1 provide three sample images, and the corresponding attention map outputs of two different layers of our model for these images. 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-tasklearning 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, improvesthe model performance. To further improve the prediction accuracy of our model, wecombine the prediction of attentional network with the residual network, and use their ensemble model as the final predictor.

1.2 Existing System

Face detection is used in biometrics, frequently as a part of (or together with) a facial recognition structure. Some current digital cameras use face detection for autofocus. Face detection is also beneficial for choosing areas of interest in photo. Face detection is in advance the interest of marketers. A webcam can be integrated into a television and detect any face that walks by. The system then computes the race, gender, and age range of theface. Once the data is composed, a series of announcements can be played that is specific toward the detected race/gender/age. This paper shows prototype or partial application ofthis type of work. Face detection is also being studied in energy conservation. Procedure for face recognition based on information theory method of coding and decoding the face imageis discussed in [Sarala A. Dabade & Mrunal S. Bewoor, 2012][4]. Proposed methodology is connection of two stages – Face detection using Haar Based Cascade classifier and recognition using Principal Component analysis. Various face detection and recognition methods have been evaluated [Faizan Ahmad et al., 2013] and solution for image detection and recognition is proposed as an initial step for video surveillance. 5 Implementation of face recognition using principal component analysis using 4 distance classifiers is proposed in [Hussein Rady, 2011]. 2. Lanitis et al. [5] proposed the first approach applying AAM to age estimation, which extracts craniofacial growth and skin aging during childhood and adulthood.

1) Age-specific estimation, which assumes that the aging process is identical for everyone; and

2) Appearance-specific estimation, which follows the assumption that people who look similar tend to have similar aging processes.

Zhang et al. [6] formulated the inference of each person's age as a warped Gaussian process (WGP) estimation problem, and developed a multi-task extension of WGP to solve the problem. Since different individuals have different aging

1.3 Disadvantages of Existing System

Existing systems for age and gender detection using deep learning have made significant progress in recent years. However, there are still some drawbacks that need to be addressed. Here are some of the Co-Existing systems for age and gender detection using deep learning

have made significant progress in recent years. However, there are still some drawbacks that need to be addressed. Here are some of the common drawbacks:

1. Limited Diversity in the Datasets: Most of the available datasets for age and gender detection are biased towards a particular ethnicity or gender, which can lead to less accurate results for other groups.
2. Challenges in Age Estimation: Age estimation is a difficult task since it is a subjective and complex concept. It can be affected by various factors such as race, gender, and lifestyle, making it challenging to predict accurately.
3. Overfitting: Overfitting occurs when the model is trained too well on the training data and does not generalize well to new data. This can lead to false positives or negatives when detecting age and gender.
4. Dependence on High-Quality Images: The accuracy of the age and gender detection system heavily depends on the quality of the images. Poor quality images can lead to inaccurate predictions.
5. Inability to Capture Facial Expressions: Facial expressions can convey a lot of information about age and gender. However, most of the existing systems do not consider facial expressions, which can lead to less accurate results.

These are some of the common drawbacks of existing systems for age and gender detection using deep learning. However, with continued research and improvements in thetechnology, these challenges can be addressed, and more accurate and reliable systems can be developed.

1.4 Proposed System

Our approach will be based on Convolutional Neural Networks (CNNs), which are a class of deep neural networks that have shown remarkable success in various computer vision tasks. We will use the transfer learning technique, which involves reusing a pre-trained CNN model and fine-tuning it on our dataset.

We will start by using a pre-trained CNN model such as VGG16, ResNet50, or InceptionV3, which are popular choices for image classification tasks. We will remove the last fully connected layer of the pre-trained model and add a new fully connected layer for age and

gender classification. We will freeze the weights of the pre-trained layers and train only the newly added layers on our dataset.

For age classification, we will use a regression approach, where the output of the age classification layer will be a single value representing the predicted age. For gender classification, we will use a binary classification approach, where the output of the gender classification layer will be a probability score indicating the likelihood of the input image belonging to the male or female class.

1.5 Advantages of Proposed System

There are several advantages of using a deep learning-based system for age and gender detection, including:

1. High accuracy: Deep learning algorithms are capable of learning complex patterns and features from images, resulting in highly accurate age and gender predictions.
2. Scalability: Deep learning models can be easily scaled to handle large datasets and can be trained on multiple GPUs or in parallel on multiple machines.
3. Robustness: Deep learning algorithms are robust to variations in lighting, pose, and facial expression, making them suitable for use in a wide range of applications.
4. Speed: Deep learning-based age and gender detection systems can process images in real-time, making them ideal for use in applications that require fast processing, such as security systems or real-time video analysis.
5. Adaptability: Deep learning models can be easily adapted to new datasets and scenarios, allowing them to be customized for specific applications or domains.

Overall, using deep learning for age and gender detection offers significant advantages over traditional methods, resulting in more accurate, efficient, and reliable systems.

2. ANALYSIS

2.1 Introduction

Due to the rise of social platforms and social media nowadays, there is also an increase in the number of applications that want automatic age and gender classification. As we know, age and Gender are two key facial attributes that play an especially significant role in social interactions. So, in this deep learning project. we will be creating real-time gender and age detection using Deep Learning, and in this, we will be using pre-trained models that classify the gender and age of the person. So, the model will predict the gender as 'Male' and 'Female,' and the predicted age will be in one of the following ranges- (0-2), (4-6), (8-12), (15-20), (2532), (38-43), (48-53), (60-100) (so there are 8 nodes in the final output layer or say SoftMax layer). It is exceedingly difficult to predict the exact age of the person due to many reasons like (makeup, light, facial expressions, etc.) so that is why we have consideredthis as a classification problem instead of a regression problem.

2.1.1: Literature Survey Summary

Sr.no	Title of the Paper	Author	Publication	Technique Used
01	Age and Gender Estimation of Unfiltered Faces	Eran Eidinger, Roee	IEEE TRANSACTION DEC. 2014	Robust face alignment technique, SVM
02	Automated Estimation of Human Age, Gender, and Expression	Yaoyu Tao	Stanford, CA 94305, USA taoyaoyu@stanford.edu	LBP & Gabor filter LDA algorithm
03	Comparison of Recent Machine Learning Techniques for Gender Recognition from Facial Images	Joseph Lemley Sami Abdul-Wahid Dipayan Banik	Central Washington University Ellensburg, WA, USA MAICS 2016	Feature extraction techniques: PCA & HOG. Gender classification methods
04	Partial Face Recognition: Alignment-Free Approach	Shengcai Liao, Anil K. Jain, Fellow, IEEE, and Stan Z. Li	IEEE transactions on pattern analysis	PCA + LDA & LBP Canny edge detector
05	Age Group Estimation using Face Features	Ranjan Jana, Debaleena Datta, Rituparna	(IJEIT) Volume 3, Issue 2, August 2013	K-means clustering algorithm. PCA, LDA.
06	Gender Recognition and Age-group Prediction: A Survey	Mr. Brajesh Patel. Mr. Raghvendra	ISSN:2319-7242 Volume 3 Dec.2014	Algorithm: SVM Adab Oost
07	Face Recognition G on Improved SIFT Algorithm	Ehsan sadeghipo Nasrollah sahragard	(IJACSA) Vol. 7, No. 1, 2016	Improved SIFT descriptor using Gabor

Table 2.1: Literature Survey Summary

2.2 Requirements Specification

2.2.1 Software Requirements

- OpenCV: We used OpenCV 3 dependency for python 3. OpenCV is library where there are lots of image processing functions are available. This is very useful library for image processing. Even one can get expected outcome without writing a single code. The library is cross-platform and free for use under the open-source BSD license. Example of some supported functions are given bellow:
    - Derivation: Gradient/Laplacian computing, contours delimitation
    - Hough transforms: lines, segments, circles, and geometrical shapesdetection
    - Histograms: computing, equalization, and object localization withback projection algorithm
    - Segmentation-thresholding, distance transform, foreground/background detection, watershed segmentation
    - Filtering: linear and nonlinear filters, morphological operations
    - Cascade detectors: detection of face, eye, car plates
    - Interest points: detection and matching
    - Video processing: optical flow, background subtraction, camshaft(object tracking)
- Operating System (Windows)
- IDE (Visual Studio)

2.2.3: Hardware Requirements:

- Processor – i3
- Storage Device(1TB)
- Memory – 8GB RAM

2.3: Flowchart

Flowchart is a type a diagram that represents a workflow or a process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach for solving a problem. The flowchart shows the steps as boxes with arrows. The diagrammatic

representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting, or managing a process or program in various fields.

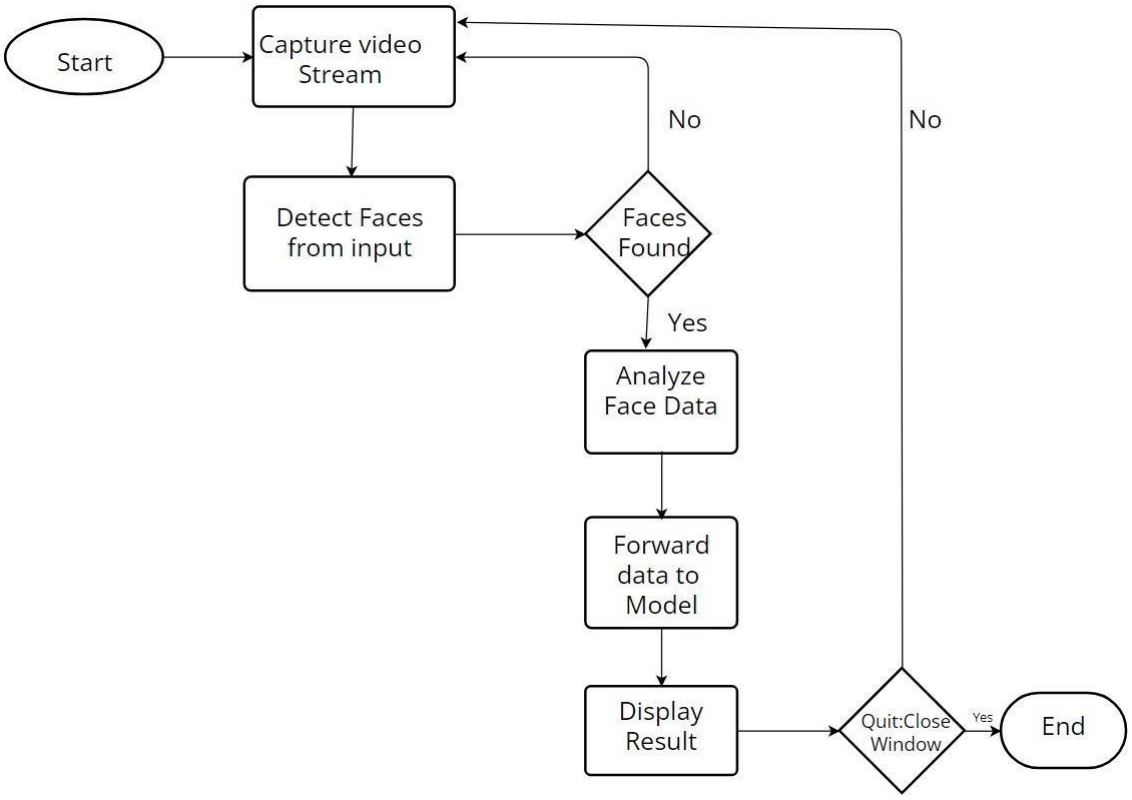


Fig 2.1: Flow Chart

3.DESIGN

3.1 Introduction

The network architecture used for age approximation in based on the paper of [2]G. Levi and T. Hassner. This network is intended to be shallow to prevent over-fitting. All the three colors i.e., Red, Green, Blue are processed directly. The images are scaled to 256 x 256 and cropped to 227 x 227. The network consists of 3 convolution layers followed by 3 fully connected layers.

The Adience Dataset: The benchmark for this problem, introduced by Eran Eidinger et al. [3], uses the Adience dataset which is composed of images scraped from Flickr.com albums that were labelled for age and gender. The benchmark uses 8 classes for age groups (02, 46, 813, 1520, 2532, 3843, 4853, 60+), and therefore we treated both gender prediction and age prediction as a classification problem. The Audience dataset is small (containing 34,795 images), so we also used the IMDB + Wiki dataset which is the largest dataset publicly available for age and gender (containing 523,051 face images).

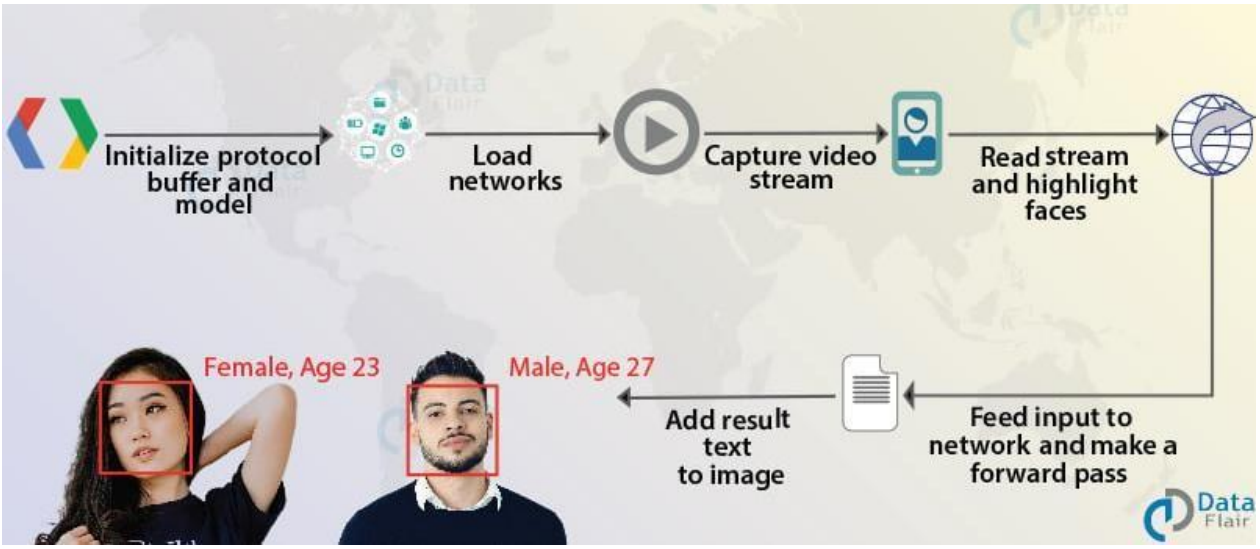


Fig 3.1: System Design

3.2 DFD/UML Diagrams

3.2.1 DFD Diagra ms

A Data Flow Diagram (DFD) is a graphical representation of how data flows through a system. In the case of age and gender detection using deep learning, a DFD can be used to represent how the input data (i.e., an image) is processed and how the results are generated.

Here are two possible levels of DFDs for age and gender detection using deep learning:

Level 0 DFD:

This level of DFD represents the overall flow of data in the system.

- Inputs: An image of a person's face
- Outputs: Age and gender estimates

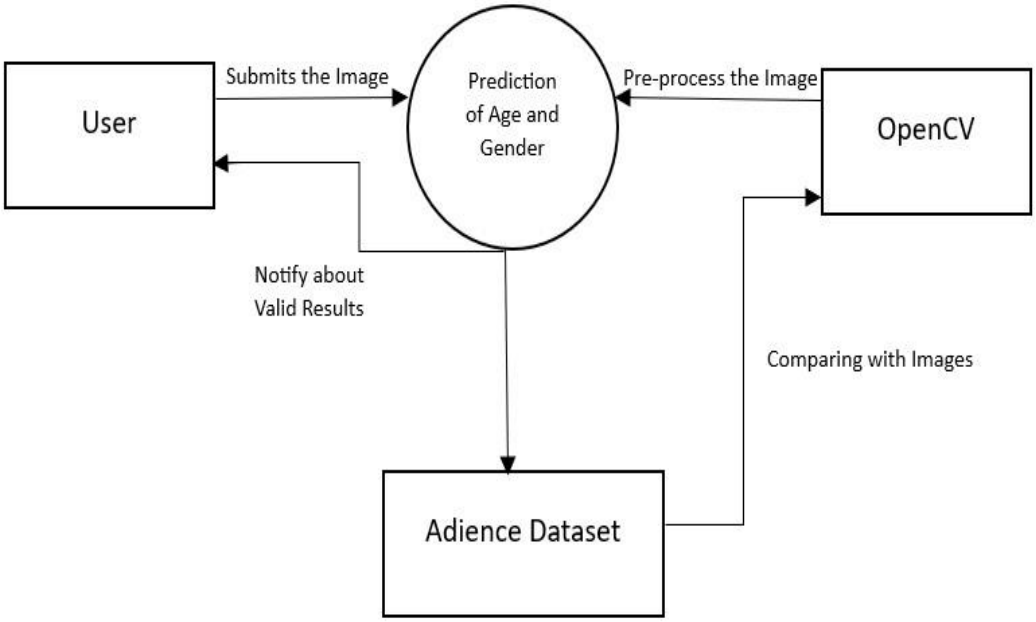


Fig 3.2: Level 0 DFD

Level 1 DFD:

This level of DFD breaks down the overall flow into more detailed steps.

- Inputs: An image of a person's face
- Process 1: Pre-processing (e.g., resizing, normalization, augmentation)
- Process 2: Feature extraction using a deep learning model

- Process 3: Age and gender estimation using a separate deep learning model for each task
- Outputs: Age and gender estimates

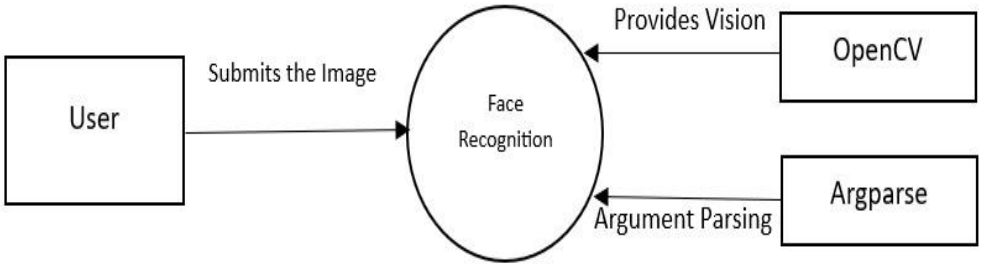


Fig 3.3: Level 1 DFD

3.2.2 UML Diagrams

UML is a modelling language used in object-oriented software design. UML provides capabilities to specify and visualize the components that make up a software system. UML diagrams mainly represent the structural view and the behavioural view of a system. Structural view of the system is represented using diagrams like class diagrams, composite structure diagrams, etc. Dynamic view of the system is represented using diagrams such as sequence diagrams, activity diagrams, etc. UML version 2.2 includes fourteen diagrams, which includes seven diagrams for representing the structural view and other seven representing the behavioural view. Among the seven behavioural diagrams, four diagrams can be used to represent interactions with the system. There are tools that can be used for UML modelling such as IBM Rational Rose.

In the uml diagrams we are going through the following diagrams:

- Use Case Diagram
- Sequence Diagram
- Class Diagram
- Activity Diagram
- Context Diagram

3.2.2.1 Use Case Diagram

In UML, use-case diagrams model the behaviour of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally. Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system. You can model a complex system with a single use-case diagram, or create many use-case diagrams to model the components of the system. You would typically develop use-case diagrams in the early phases of a project and refer to them throughout the development process. Use cases are represented with a labelled oval shape. Stick figures represent actors in the process, and the actor's participation in the system is modelled with a line between the actor and use case. To depict the system boundary, draw a box around the use case itself.
UML Use case diagrams are ideal for:

- Representing the goals of system-user interactions.
- Defining and organizing functional requirements in a system.
- Specifying the context and requirements of a system.
- Modelling basic flow of events in a use case.

An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems.
- Goals that your system or application helps those entities (known as actors) achieve.
- The scope of your system.

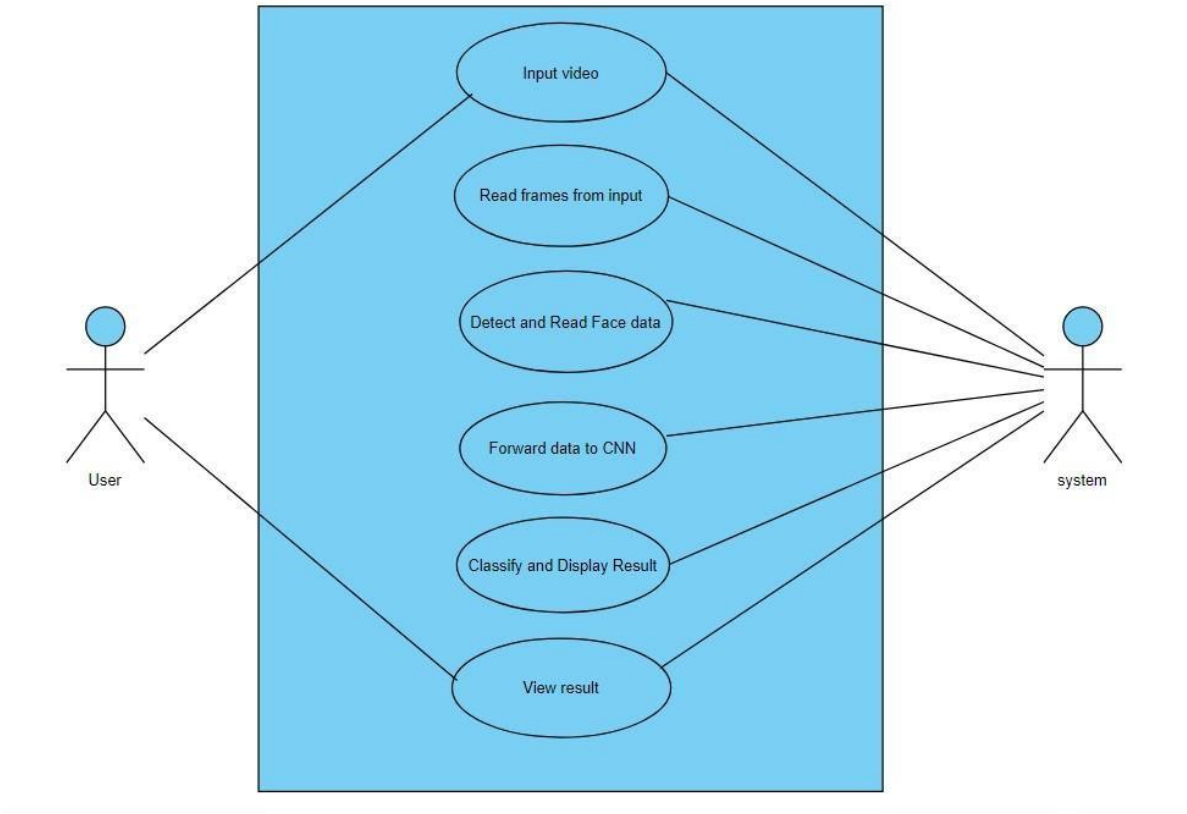


Fig 3.4: Use Case Diagram

3.2.2.2 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios. Sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- See how objects and components interact with each other to complete a process.
- Plan and understand the detailed functionality of an existing or future scenario.

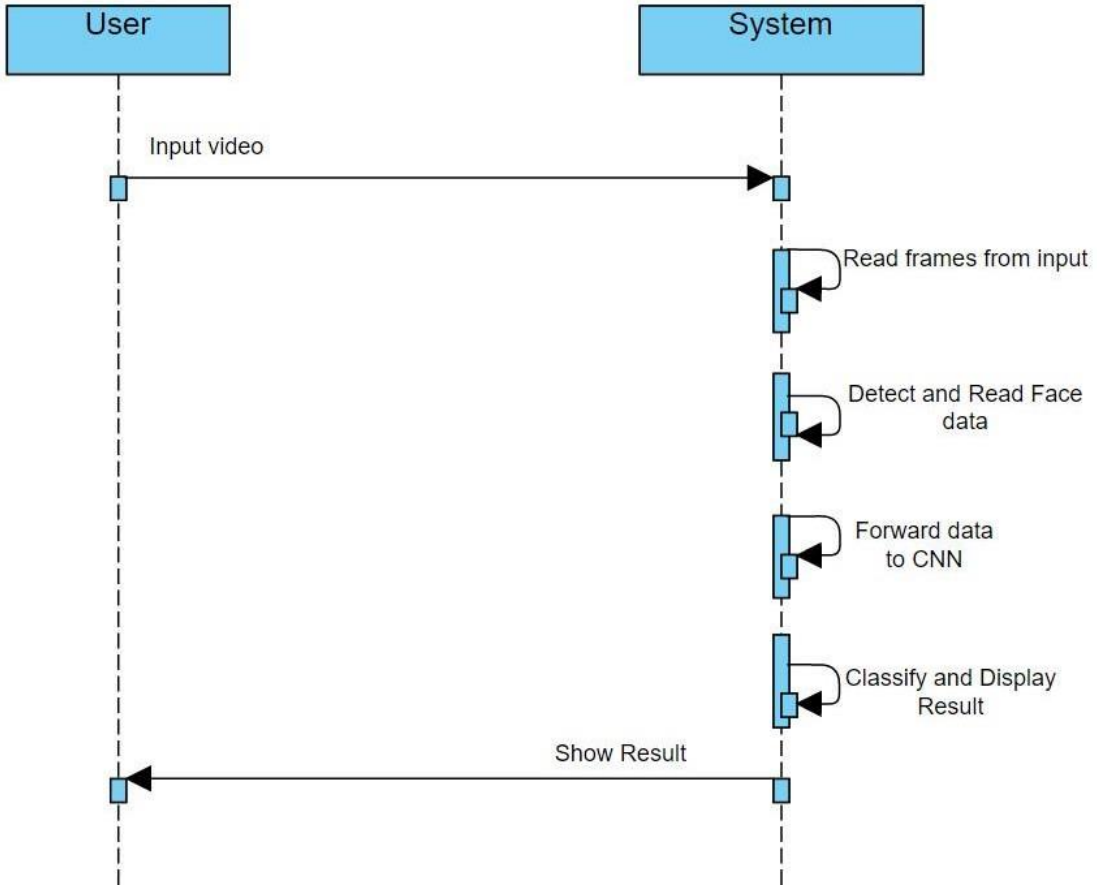


Fig 3.5: Sequence diagram

3.2.2.3 Class Diagram

The class diagram depicts the static view of an application. It is used to visualize describe, document various aspects of the system. The class diagram constitutes class name, attributes and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaboration, and constraints, it is termed as a structural diagram. Purpose of class diagram:

1) It analyses and design a static view of an application

2) It describes the major responsibilities of a system 11 Benefits of class diagram:

1)It can represent the object model for complex system

2) It reduces the maintenance time by providing of an overview of how an application is structured before coding.

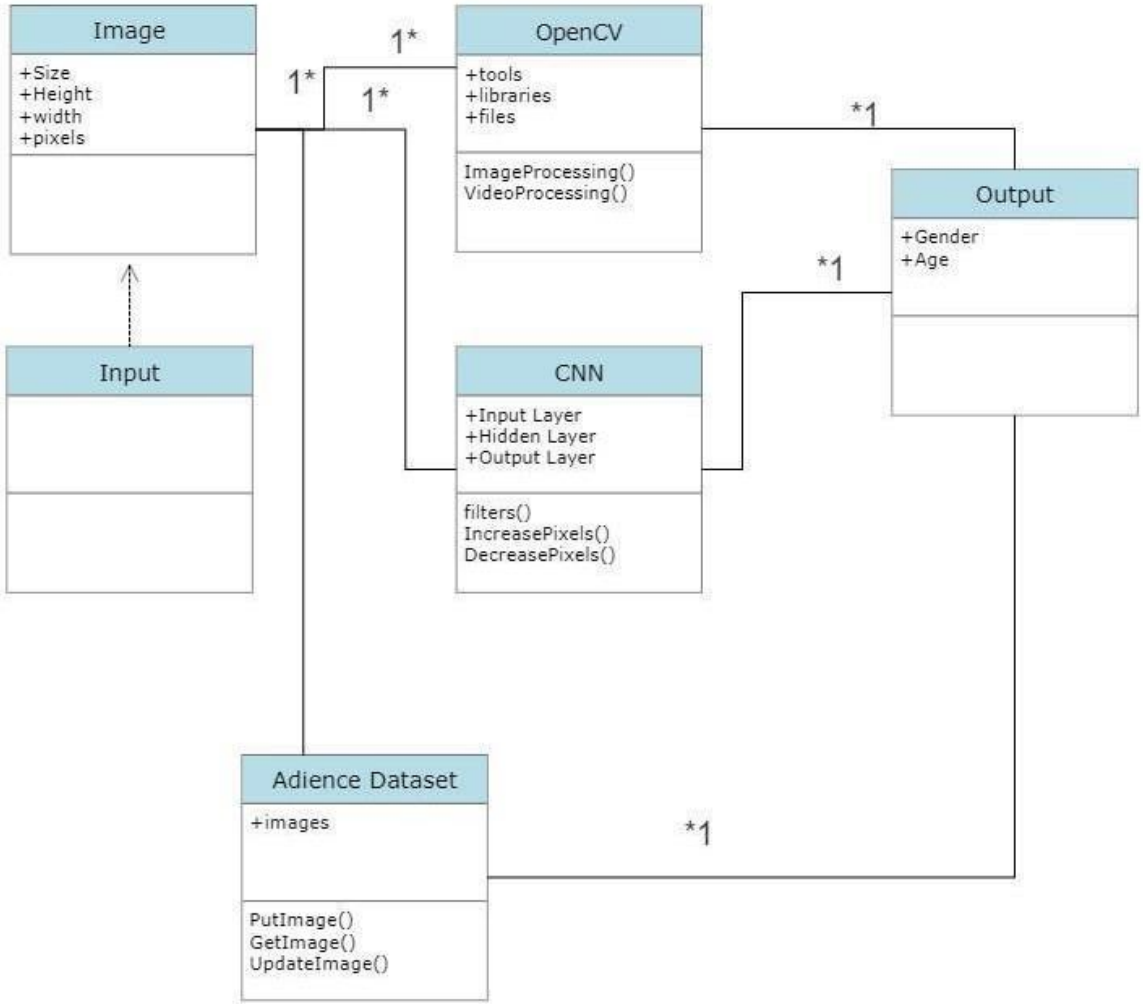


Fig 3.6: Class Diagram

3.2.2.4 Activity Diagram

Activity diagram is a behavioural diagram and it represents the behaviour using finite state transitions state chart diagram also known as state machine diagram. Simplify. a state chart diagram is used to model the dynamic behaviour of a class in response to time and changing external stimuli. purpose of state chart diagram:

1) To model the dynamic aspect of system.

2) To model the lifetime of a reactive of the system.

3) To describe the different states of an object during its life time.

4) To model the dynamic aspect of system.

5) To model the lifetime of a reactive of the system.

6) To describe the different states of an object during its life time.

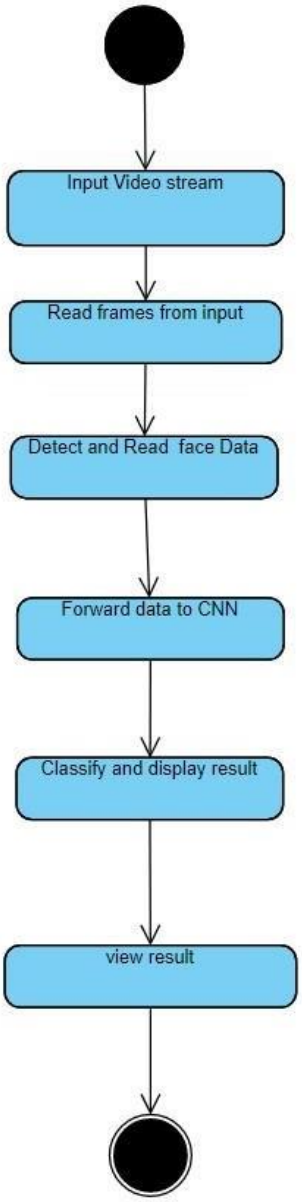


Fig 3.7: Activity Diagram

3.2.2.4 Context Diagram

In the below block diagram, an image is taken as input after the input is taken by the system it starts the preprocessing of an image. In the features extraction phase the image features (like eyes, nose etc.) are extracted by using the convolutional neural networks. Adience dataset is involved which includes trained images and these trained images are also preprocessed to get more accurate results. In NN classifier phase the input images and the

dataset images are compared with the features and then gives the output as age within the 8 ranges and the gender as male or female.

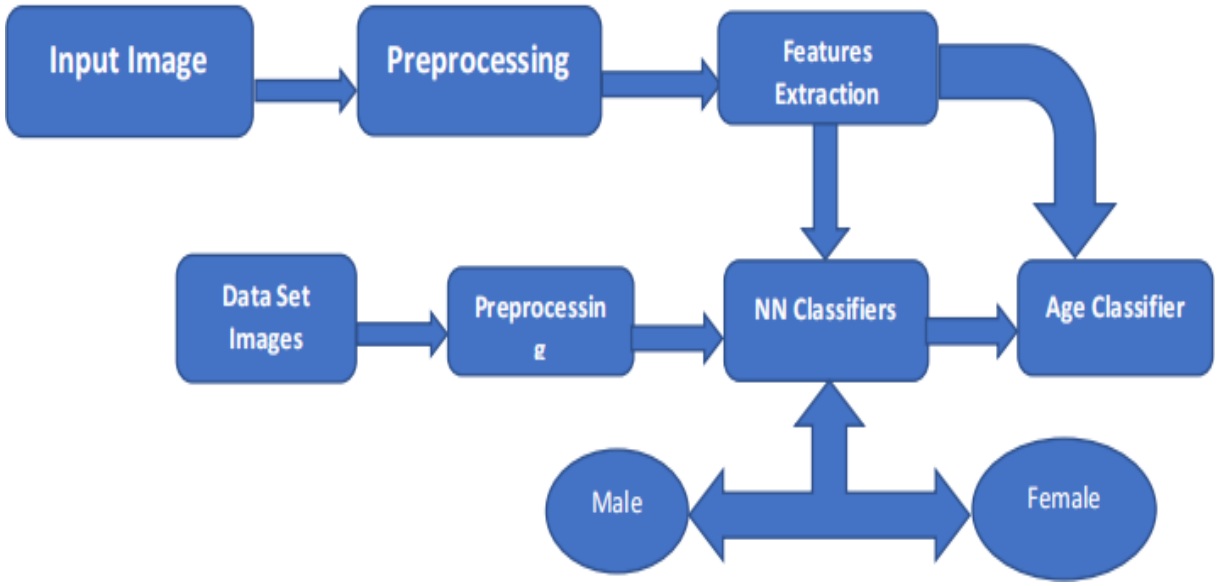


Fig 3.8: Context Diagram

3.3 Module design & Organization

Adience Dataset:The Adience dataset was created to aid the study of age and gender recognition, and can be also used as benchmark dataset for face photos

Gender	0 - 3	4 - 8	9 - 13	14 - 20	21 - 25	26 - 32	33 - 40	41 - 48	48 - 55	55 - 62	62 - 100	Total
Male	700	900	900	700	2300	1200	1300	800	700	750	450	10,700
Female	700	900	1200	1450	2700	1500	1700	800	650	500	400	12,500
Both	1400	1800	2100	2150	5000	2700	3000	1600	1350	1250	850	23,200

Fig 3.9: The Adience Dataset

4.IMPLEMENTATION & DETAILS

4.1 Introduction

Age and gender detection is a deep learning-based model the main concept of the deep learning used in this project is to get the accurate results. The age and gender detection are one of the promising research projects. To implement this project some of the technologies are used those are the Convolutional neural Networks (CNN) for extracting the features from an input image and the other is one of the python libraries called the OpenCV which can process an image. And, dataset is used I.e., Adient dataset which Is trained with some images along with age and gender features.

4.1.1 Model Code

```
import cv2 import
numpy as np
FACE_PROTO = "weights/deploy.prototxt.txt"
FACE_MODEL = "weights/res10_300x300_ssd_iter_140000_fp16.caffemodel"
GENDER_PROTO = 'weights/gender_net.caffemodel'
MODEL_MEAN_VALUES = (78.4263377603, 87.7689143744, 114.895847746)
GENDER_LIST = ['Male', 'Female'] AGE_MODEL = 'weights/deploy_age.prototxt'
AGE_PROTO = 'weights/age_net.caffemodel'
AGE_INTERVALS = ['(0, 2)', '(4, 6)', '(8, 12)', '(15, 20)',
 '(25, 32)', '(38, 43)', '(48, 53)', '(60, 100)']
frame_width = 1280  frame_height = 720
face_net = cv2.dnn.readNetFromCaffe(FACE_PROTO, FACE_MODEL) age_net =
cv2.dnn.readNetFromCaffe(AGE_MODEL, AGE_PROTO) # Load gender
prediction model
gender_net = cv2.dnn.readNetFromCaffe(GENDER_MODEL, GENDER_PROTO) def
get_faces(frame, confidence_threshold=0
blob = cv2.dnn.blobFromImage(frame, 1.0, (300, 300), (104, 177.0, 123.0))
face_net.setInput(blob) output = np.squeeze(face_net.forward()) faces = [] 19
# Loop over the faces detected
for i in range(output.shape[0]):
```

```
confidence = output[i, 2] if
confidence > confidence_threshold:
box = output[i, 3:7] * \
np.array([frame.shape[1], frame.shape[0], frame.shape[1], frame.shape[0]])
# Convert to integers
start_x, start_y, end_x, end_y = box.astype(np.int) # Widen the box a little start_x, start_y, end_x, end_y = start_x - \
10, start_y - 10, end_x + 10, end_y + 10
start_x = 0 if start_x < 0 else start_x
start_y = 0 if start_y < 0 else start_y
end_x = 0 if end_x < 0 else end_x  end_y
= 0 if end_y < 0 else end_y
 # Append to our list
faces.append((start_x, start_y, end_x, end_y))
return faces  def display_img(title, img):
cv2.imshow(title, img) cv2.waitKey(0) cv2.destroyAllWindows() def
image_resize(image, width = None, height = None, inter = cv2.INTER_AREA):
dim = None
(h, w) = image.shape[:2] if width is
None and height is None:
return image 20 if width
is None: r = height /
float(h) dim = (int(w *
r), height)  else:
 r = width / float(w)
dim = (width, int(h * r)
return cv2.resize(image, dim, interpolation = inter)  def
get_gender_predictions(face_img):
blob = cv2.dnn.blobFromImage( image=face_img, scalefactor=1.0,
size=(227, 227), mean=MODEL_MEAN_VALUES,
```

```
swapRB=False, crop=False )  gender_net.setInput(blob)  return
gender_net.forward()  def get_age_predictions(face_img):
blob = cv2.dnn.blobFromImage( image=face_img, scalefactor=1.0,
size= (227, 227),
mean=MODEL_MEAN_VALUES, swapRB=False
)
age_net.setInput(blob) return
age_net.forward() def
predict_age_and_gender(input_path: str):
img = cv2.imread(input_path)
frame = img.copy() if
frame.shape[1] > frame_width:
frame = image_resize(frame, width=frame_width)  faces
= get_faces(frame) for i, (start_x, start_y, end_x, end_y)
in enumerate(faces):
face_img = frame[start_y: end_y, start_x: end_x]
age_preds = get_age_predictions(face_img)
gender_preds = get_gender_predictions(face_img)
i = gender_preds[0].argmax() gender =
GENDER_LIST[i] gender_confidence_score =
gender_preds[0][i]
i = age_preds[0]. argmax() age =
AGE_INTERVALS[i]
age_confidence_score = age_preds[0][i]
label = f"{gender}-{gender_confidence_score*100:.1f} %, {age}-
{age_confidence_score*100:.1f}%"
# label = "{}-{:.2f}%". format(gender, gender_confidence_score*100)
print(label)  yPos = start_y - 15  while yPos < 15:
yPos += 15 box_color = (255, 0, 0) if gender == "Male" else (147, 20, 255)
cv2.rectangle(frame, (start_x, start_y), (end_x, end_y), box_color, 2) font_scale = 0.54
cv2.putText(frame, label, (start_x, yPos),
```

```
cv2.FONT_HERSHEY_SIMPLEX, font_scale, box_color, 2
display_img("Gender Estimator", frame)
cv2.imwrite("output.jpg", frame) cv2.destroyAllWindows()
if __name__== "_main_":
import sys input_path = sys.argv[1]  predict_age_and_gender(input_path)
import cv2
import math
import argparse

def highlightFace(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()
    faceBoxes=[]
    for i 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)
            faceBoxes.append([x1,y1,x2,y2])
            cv2.rectangle(frameOpencvDnn, (x1,y1), (x2,y2), (0,255,0), int(round(frameHeight/150)),
8)
    return frameOpencvDnn,faceBoxes


parser=argparse.ArgumentParser()
parser.add_argument('--image')
```

```
args=parser.parse_args()

faceProto="opencv_face_detector.pbtxt"
faceModel="opencv_face_detector_uint8.pb"
ageProto="age_deploy.prototxt"
ageModel="age_net.caffemodel"
genderProto="gender_deploy.prototxt"
genderModel="gender_net.caffemodel"

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']

faceNet=cv2.dnn.readNet(faceModel,faceProto)
ageNet=cv2.dnn.readNet(ageModel,ageProto)
genderNet=cv2.dnn.readNet(genderModel,genderProto)

video=cv2.VideoCapture(args.image if args.image else 0)
padding=20
while cv2.waitKey(1)<0:
    hasFrame,frame=video.read()
    if not hasFrame:
        cv2.waitKey()
        break

    resultImg,faceBoxes=highlightFace(faceNet,frame)
    if not faceBoxes:
        print("No face detected")

    for faceBox in faceBoxes:
        face=frame[max(0,faceBox[1]-padding):
                   min(faceBox[3]+padding,frame.shape[0]-1),max(0,faceBox[0]-padding)
                   :min(faceBox[2]+padding, frame.shape[1]-1)]
```

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

        ageNet.setInput(blob)
        agePreds=ageNet.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"
}
```

4.2 Explanation of Key Features

The Key functions that are to be noticed, processed, and implemented are:

➢ Detect faces

➢ Classify into Male/Female

➢ Classify into one of the 8 age ranges (0 – 2), (4 – 6), (8 – 12), (15 – 20), (25 – 32), (38 – 43), (48 – 53), (60 – 100).

➢ Put the results on the image and display it

4.3 Method Of Implementation

To implement this system, we should use Convolutional Neural Network(CNN) algorithm. A CNN (Convolution Neural Network) uses a system like a multilayer perceptron that has been designed to process the requirements faster. The CNN layer consist of an input layer, an output layer and a hidden layer that includes multiple convolution layers, pooling layers, fully connected layers, and normalization layers. The removal of limitations and increase in efficiency for image processing results in a system that is far more effective, simpler to trains limited for image processing and natural language processing.

4.3.1 Convolutional Neural Network

Convolutional neural networks are one of the most common types of neural networks used in computer vision to recognize objects and patterns in images. One of their defining traits is the use of filters within convolutional layers. Neural networks are artificial systems that were inspired by biological neural networks. These systems learn to perform tasks by being exposed to various datasets and examples without any task specific rules. The idea is that the system generates identifying characteristics from the data they have been passed without being programmed with a pre-programmed understanding of these datasets. Neural networksare based on computational models for threshold logic. Threshold logic is a combination of algorithms and mathematics. Neural networks are based either on the study of the brain or on the application of neural networks to artificial intelligence. The work has led to improvements in finite automata theory. Components of a typical neural network involve neurons, connections, weights, biases, propagation function, and a learning rule. Neurons will receive an input from predecessor neurons that have an activation, threshold, an activation

function f, and an output function. Connections consist of connections, weights and biases which rules how neuron transfers output to neuron. Propagation computes the input and outputs the output and sums the predecessor neurons function with the weight.The learning rule modifies the weights and thresholds of the variables in the network.

Within a convolutional layer, the input is transformed before being passed to the next layer.A CNN transforms the data by using filters.

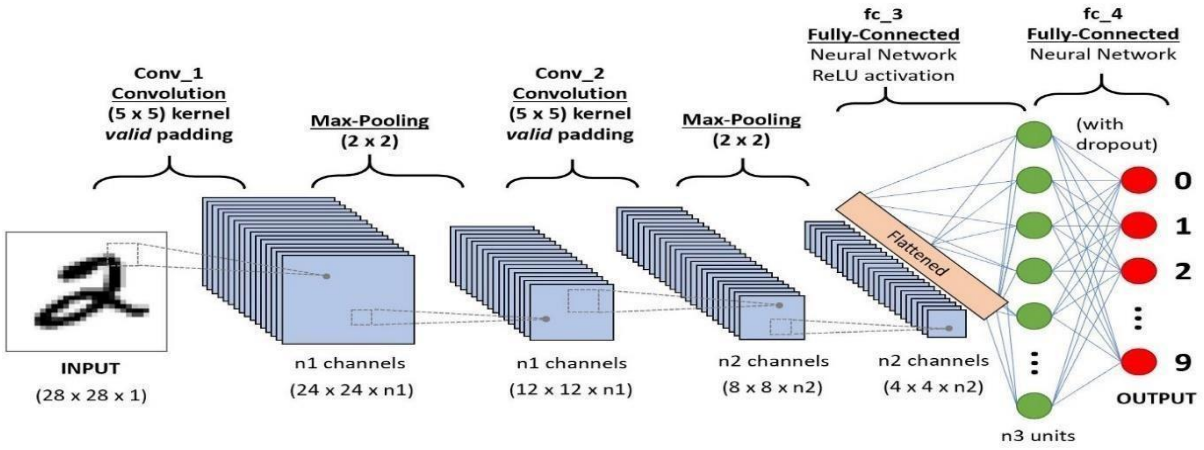


Fig 4.1: Convolutional Neural Network

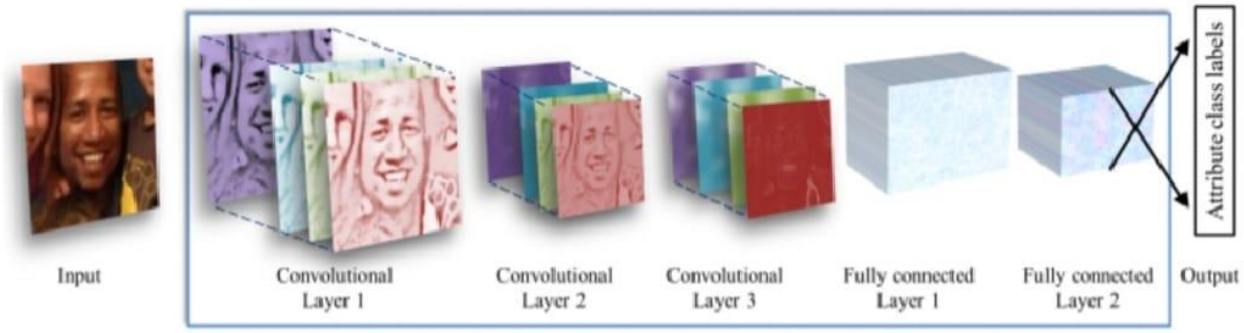


Fig 4.2: Illustration of neural network architecture used in classification method

4.3.1.1 Evolution of Neural Networks

Hebbian learning deals with neural plasticity. Hebbian learning is unsupervised and deals with long term potentiation. Hebbian learning deals with pattern recognition and exclusive- or circuits; deals with if-then rule. Back propagation solved the exclusive-or issue that Hebbian learning could not handle. This also allowed for multi-layer networks to be feasible and efficient. If an error was found, the error was solved at each layer by modifying the weights at each node. This led to the development of support vector machines, linearclassifiers, and max-pooling. The vanishing gradient problem affects feedforward networks that use back propagation and recurrent neural network. This is known as deep-learning. Hardware-based designs are used for biophysical simulation and neurotrophic computing. They have large scale component analysis and convolution creates new class of neural computing with analog. This also solved backpropagation for many-layered feedforward neural networks. Convolutional networks are used for alternating between convolutional layers and maxpooling layers with connected layers (fully or sparsely connected) with a final classification layer. The learning is done without unsupervised pre-training. Each filter is equivalent to a weights vector that has to be trained. The shift variance has to beguaranteed to dealing with small and large neural networks. This is being resolved in Development Networks.

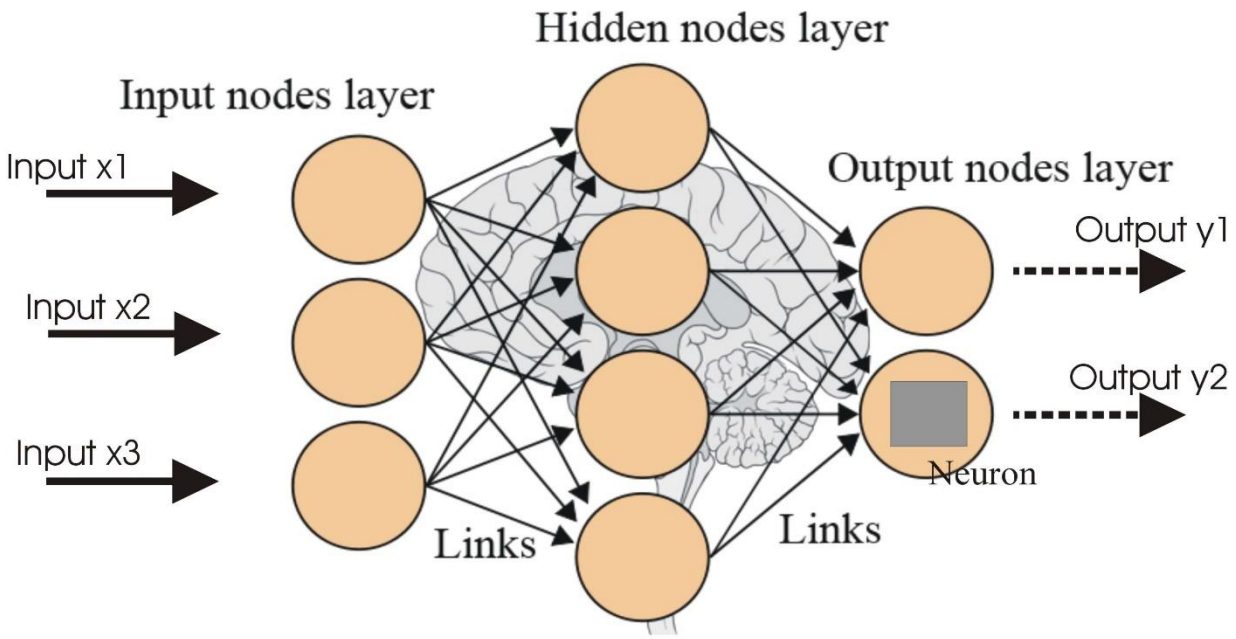


Fig 4.3: Basic layered Architecture of Convolutional Neural Network

4.3.1.2 Supervised vs Unsupervised Learning

Neural networks learn via supervised learning; Supervised machine learning involves an input variable x and output variable y. The algorithm learns from a training dataset. With each correct answers, algorithms iteratively make predictions on the data. The learning stops when the algorithm reaches an acceptable level of performance. Unsupervised machine learning has input data X and no corresponding output variables. The goal is to model the underlying structure of the data for understanding more about the data. The keywords for supervised machine learning are classification and regression. For unsupervised machine learning, the keywords are clustering and association

4.1.3.2 Supervised vs Unsupervised Learning

Neural networks learn via supervised learning; Supervised machine learning involves an input variable x and output variable y. The algorithm learns from a training dataset. With each correct answers, algorithms iteratively make predictions on the data. The learning stopswhen the algorithm reaches an acceptable level of performance. Unsupervised machine learning has input data X and no corresponding output variables. The goal is to model the underlying structure of the data for understanding more about the data. The keywords for supervised machine learning are classification and regression. For unsupervised machine learning, the keywords are clustering and association.

4.1.3.3.Model Confusion Matrix

To provide a more detailed analysis of the proposed model's performance, we also present the confusion matrix of this model for each of the two tasks. Figure 13 and Figure 14 show the confusion matrix for age range and gender classification respectively. It can clearly be seen that a vast majority of the cases are predicted as the true label of their class, seen on the main diagonal of the matrix. It is worth noting the largest false positive percentage in the age range classification model, corresponds to the images belonging to the 30-40 age range which are mistaken for 20-30.

Fig 4.4: Confusion matrix for age range

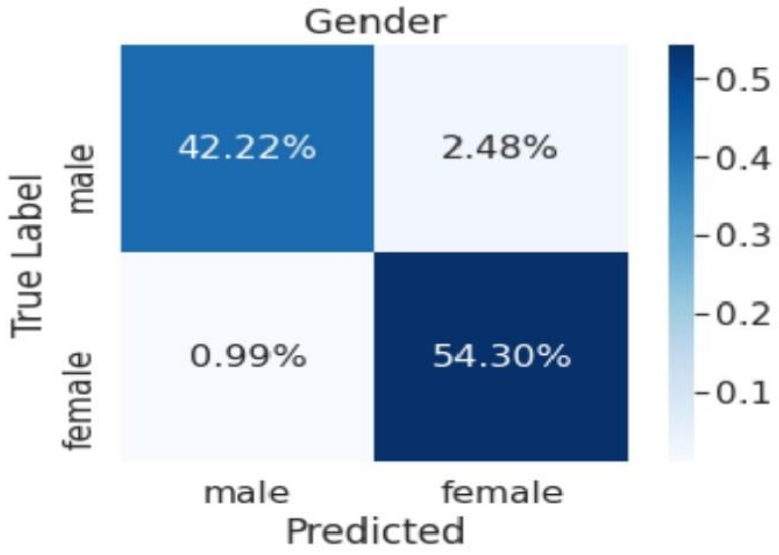


Fig 4.5 Confusion matrix for gender outputs

4.3.1.3 What are Filters in Convolutional Neural Networks?

A filter in a CNN is simply a matrix of randomized number values like in the diagram below.

0.230	**0.380**	**0.971**
0.402	**0.119**	**0.886**
0.693	**0.563**	**0.771**

Fig 4.6: Sample 3 x 3 filter

The number of rows and columns in the filter can vary and is dependent on the use case and data being processed. Within a convolutional layer, there are a number of filters that move through an image. This process is referred to as convolving. The filter convolves the pixels of the image, changing their values before passing the data on to the next layer in the CNN.

4.3.1.5 How do Filters Work?

To understand how filters transform data, let's take a look at how we can train a CNN to recognize handwritten digits. Below is an enlarged version of a 28x28 pixel image of the number seven from the MNIST dataset.

As the filter convolves through the image, the matrix of values in the filter line up with the pixel values of the image and the dot product of those values is taken. The filter moves, or convolves, through each 3 x 3 matrix of pixels until all the pixels have been covered. The dot product of each calculation is then used as input for the next layer. Initially, the values in the filter are randomized.

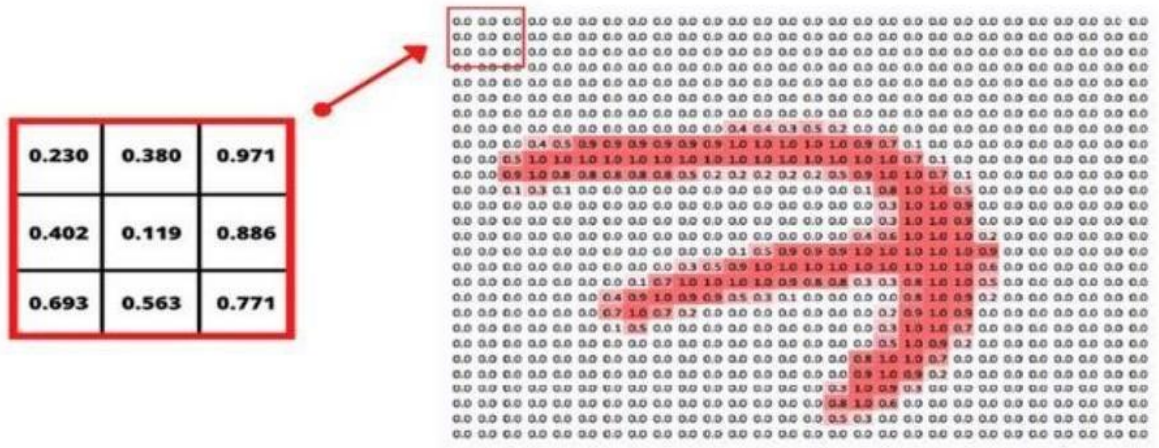


Fig 4.7: Applying the filter

As a result, the first passes or convolutions act as a training phase and the initial output isn't very useful. After each iteration, the CNN adjusts these values automatically using a loss function. As the training progresses, the CNN continuously adjusts the filters. By adjusting these filters, it is able to distinguish edges, curves, textures, and more patterns and features of the image. While this is an amazing feat, in order to implement loss functions, a CNN needs to be given examples of correct output in the form of labelled training data.

4.3.1.4 CNN-Architecture

The training of deep networks provides with more accurate results, but sometimes it is not satisfied with the availability of few samples and may be in need of more training data. In this work, CNN makes the availability of possible training samples and finally they are aggregated in the training set to create an enlarged training set.

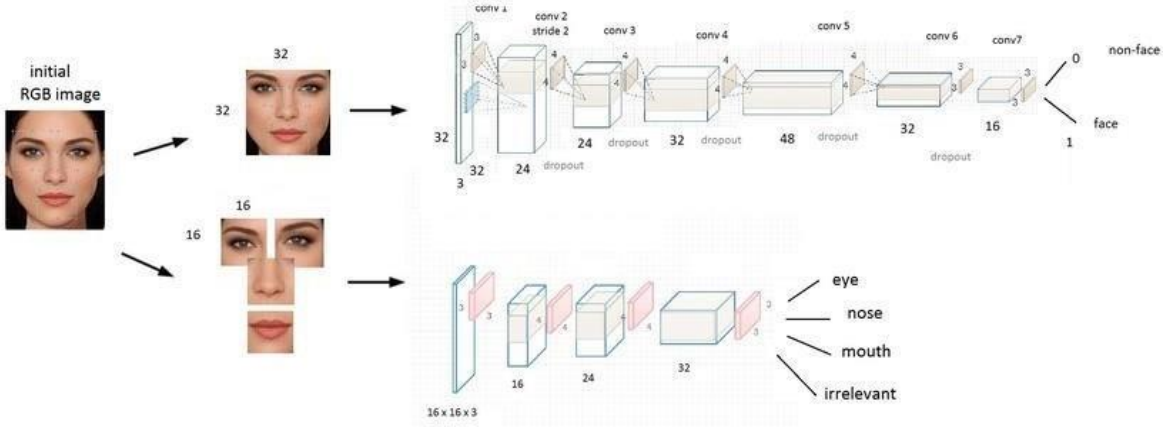


Fig 4.8: CNN Architecture

1)Face detection using CNN

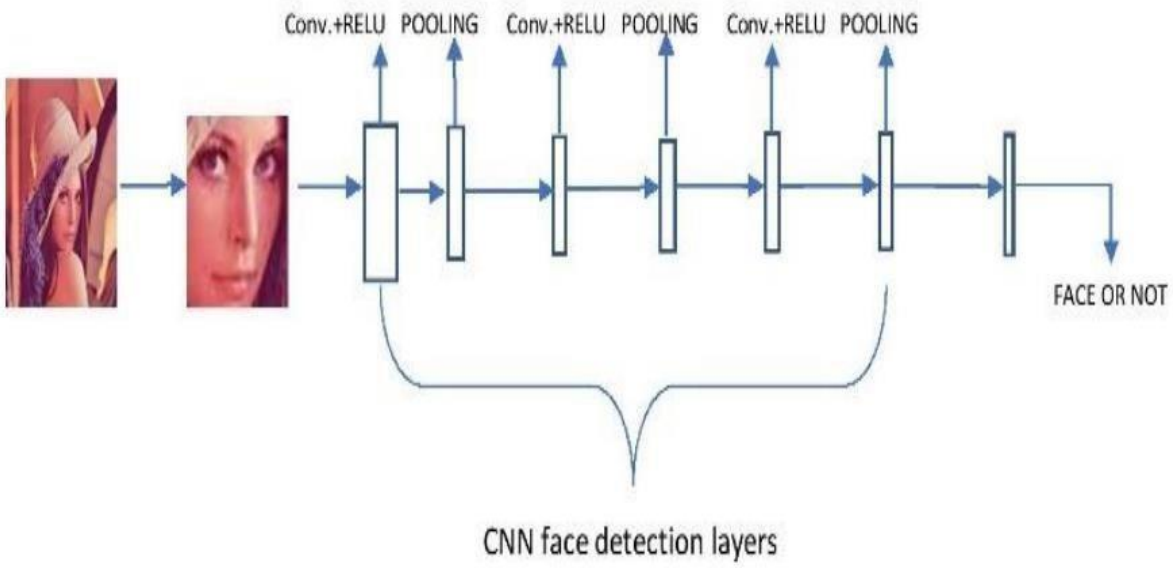


Fig 4.9: Face Detection using CNN

The main aim is to locate and extract the features from the pictures to be used facial recognition algorithm. Whenever an input image is given, it then matches with all the pictures present in the database and gets ready to by extracting the features of the image.

2)Face recognition using CNN

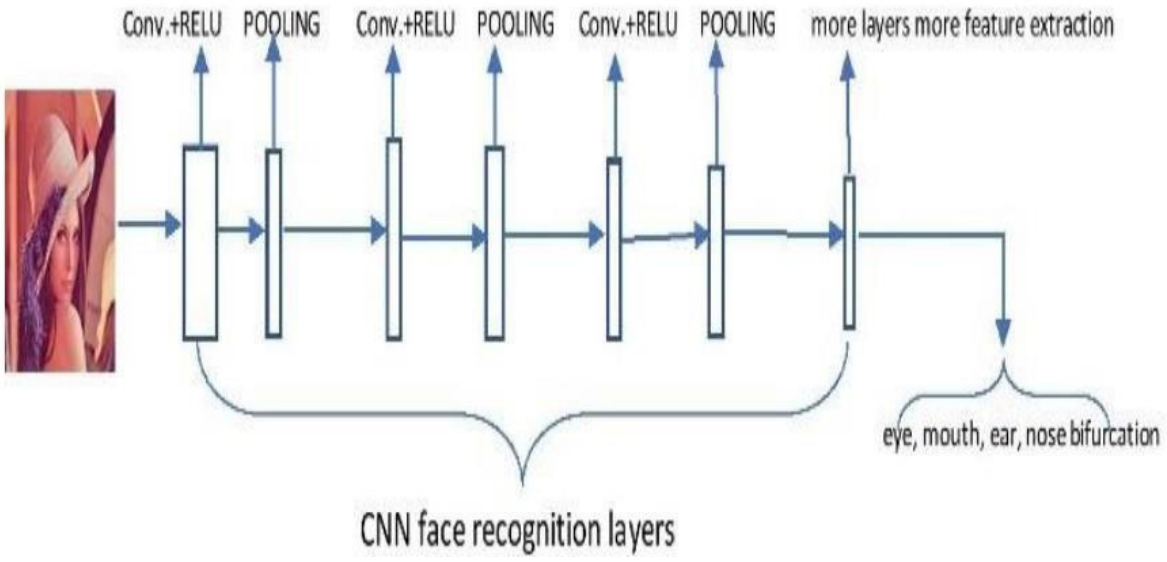


Fig 4.10: Face Recognition using CNN

4.3.1.5 Defining The CNN Layers

4.3.1.5.1 Input layer

4.3.1.5.2 Convo layer (Convo + RELU)

4.3.1.5.3 Pooling layer

4.3.1.5.4 Fully connected(FC) layer

4.3.1.5.5 Softmax/Logistic layer

4.3.1.5.6 Output layer

- Input Layer:

The input layer in CNN should contain image data. Image data is represented by a 3dimensional matrix, as we saw earlier. It would be best if you reshaped it into a single column. For example, suppose you have a picture of dimension 28x28 =784; you need to change it into 784 x 1 before feeding it into an input. If you have "m" training examples, then the input dimension will be (784, m).

- Convolution Layer:

The Convo layer is sometimes called the feature extractor layer because the image features are extracted within this layer. First of all, a part of an image is connected to theConvo layer to perform convolution operation as we saw earlier and calculate the dot product between the filter and the receptive field (it is a global part of the input image with the same size as that of filter). The result of the operation is a single integer of the output volume. Then we slide the filter over the following receptive field of the same

input image by a Stride, and we repeat the operation again and again. We will continue the same process again and again until we get through the complete picture. Then, the output will be conducted to the input for the next layer. Convo layer contains Rectified Linear Unit activation to make all negative results to zero.

- Pooling Layer:

The pooling layer is used to reduce the spatial volume of the input image after convolution. It is used between two convolution layers. If we apply F.C. after the Convo layer without applying max pooling or pooling, the calculation part will be costly, and we don't want it. So, max pooling is the only way to reduce the spatial volume of an input image. In the above example, we have applied max-pooling with a Stride of 2 in a single depth slice. As a result, we can observe the 4X4 dimension input is decreased to 2X2 dimensions. There is no parameter in the pooling layer, but it consists of two primary hyper parameters

— Stride(S) and Filter(F). In general, if we have input dimension Width1XHeight1XDepth1, then Width2=(Width1−Filter)/Stride+1 Height2 = (Height1−Filter)/Stride+1 Depth2 = Depth1 Where Height2, Width2, and Depth2 are the height, width, and depth of output.

- RELU correction layer :

RELU (Rectified Linear Units) refers to the real non-linear function defined by RELU(x)=max(0,x). Visually, it looks like the following: Fig 5.12- RELU Function

Fully Connected Layer:

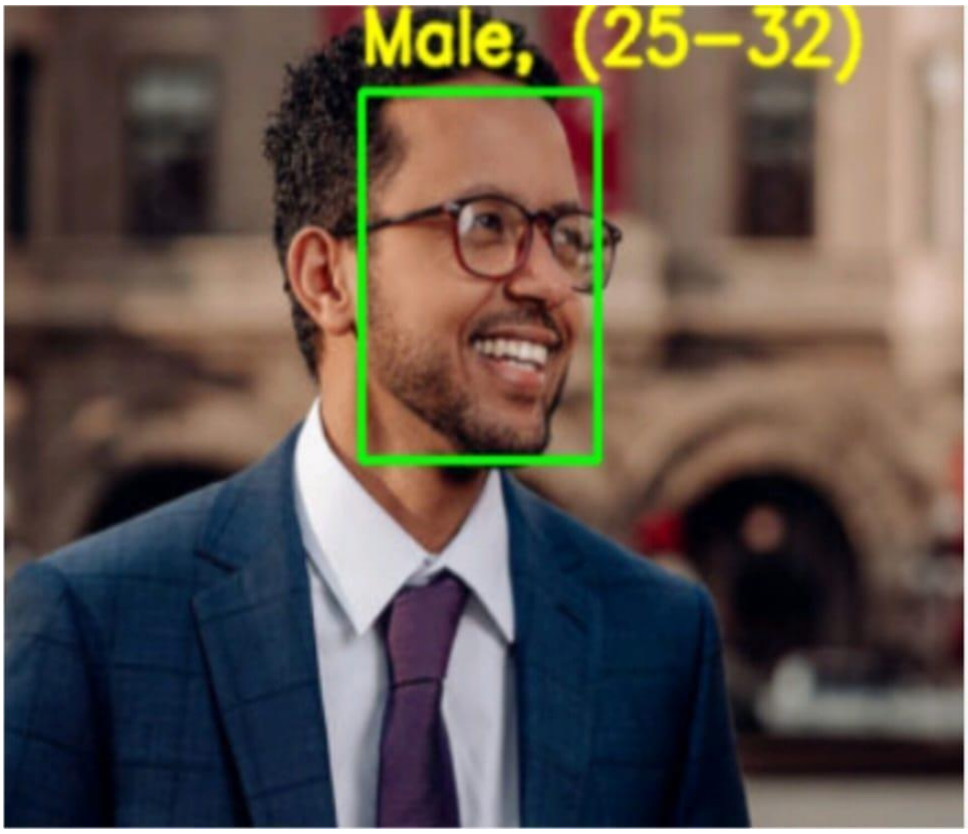
The fully-connected layer is always the last layer of a neural network, convolutional or not — so it is not characteristic of a CNN. This type of layer receives an input vector and produces a new output vector. To do this, it applies a linear combination and then possibly an activation function to the input values received. The last fully-connected layer classifies the image as an input to the network: it returns a vector of size N, where N is the number of classes in our image classification problem. Each element of the vector indicates the probability for the input image to belong to a class. To calculate the probabilities, the fully-connected layer, therefore, multiplies each input element byweight, makes the sum, and then applies an activation function (logistic if N=2, softmax if N>2). This is equivalent to multiplying the input vector by the matrix containing the weights. The fact that each input value is connected with all output values explains the term fully-connected. The convolutional neural network learns weight values in the

same way as it learns the convolution layer filters: during the training phase, by backpropagation of the gradient. The fully connected layer determines the relationship between the position of features in the image and a class. Indeed, the input table being the result of the previous layer, it corresponds to a feature map for a given feature: the high values indicate the location (more or less precise depending on the pooling) of this feature in the image. If the location of a feature at a certain point in the image is characteristic of a certain class, then the corresponding value in the table is given significant weight.

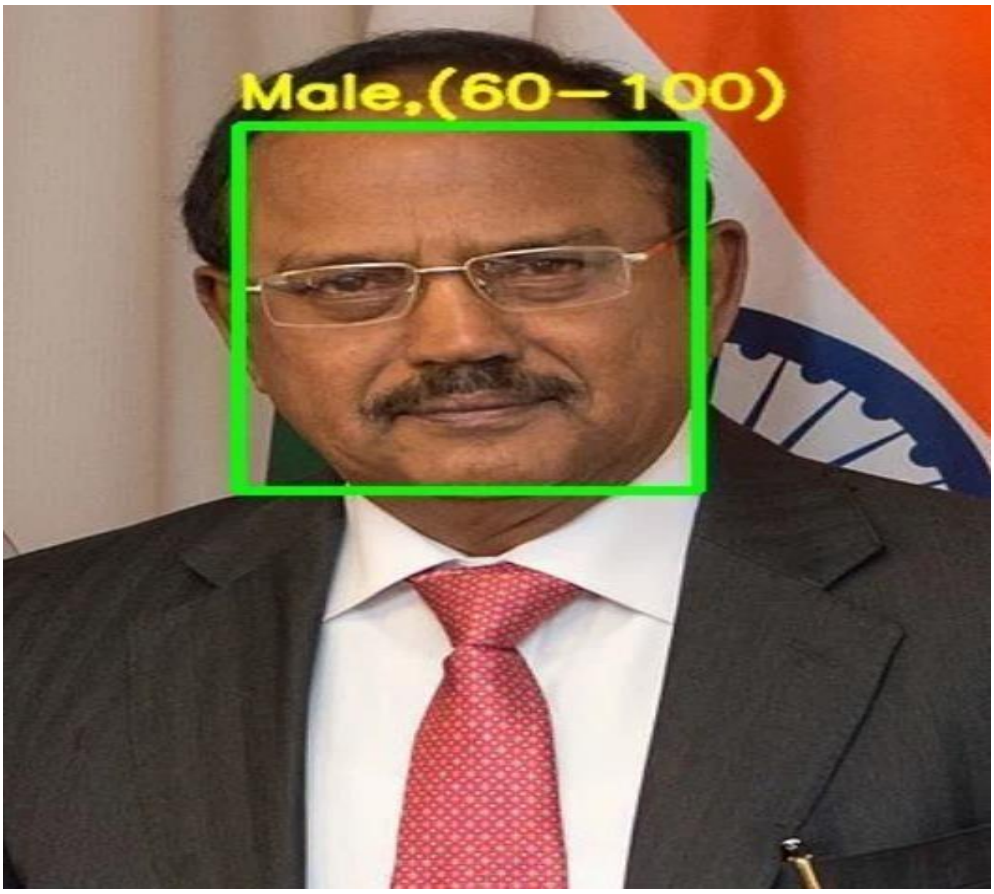
- Output Layer:

  The output layer contains the flag or label, which is of the form one-hot encoded.

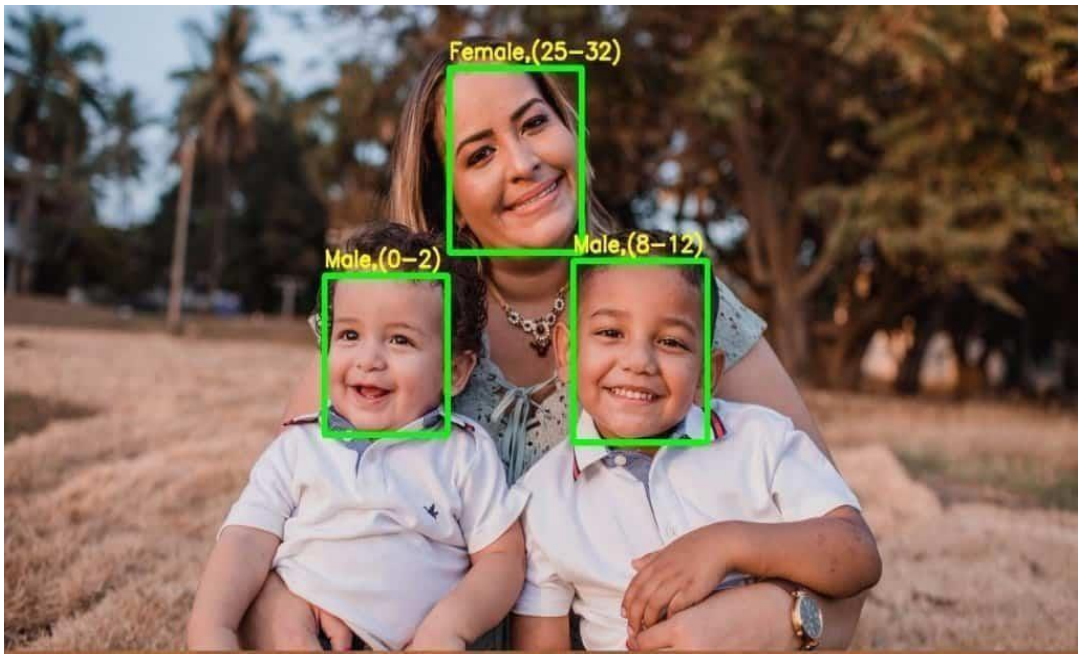
4.3.2 Output Screens



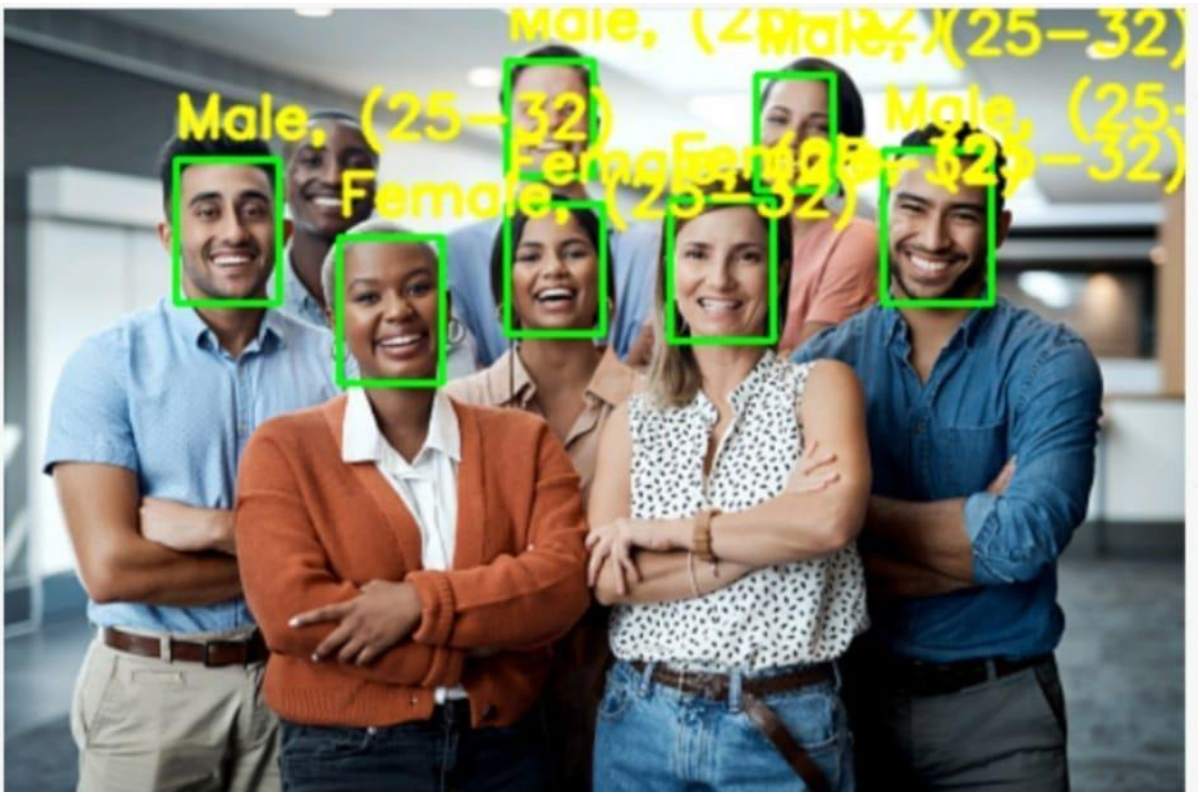
Screen 4.1: Single Person Image1



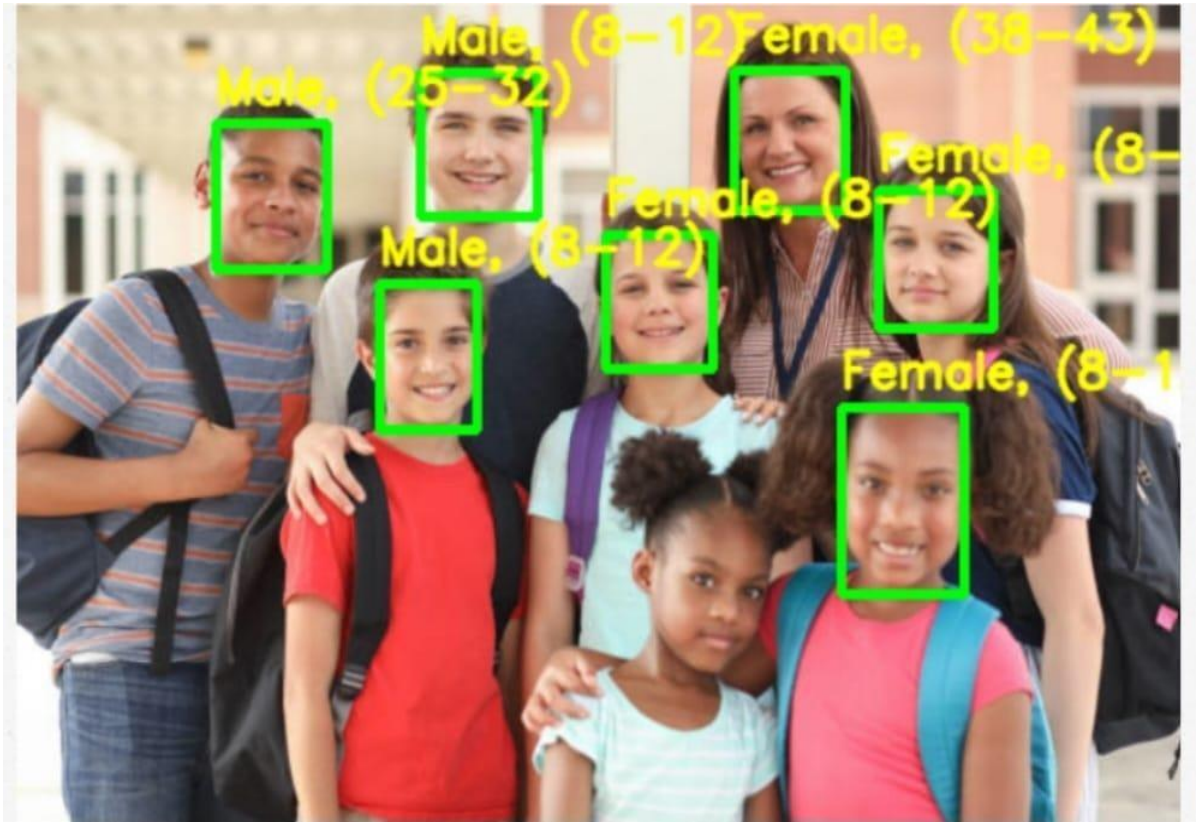
Screen 4.2: Single Person Image2



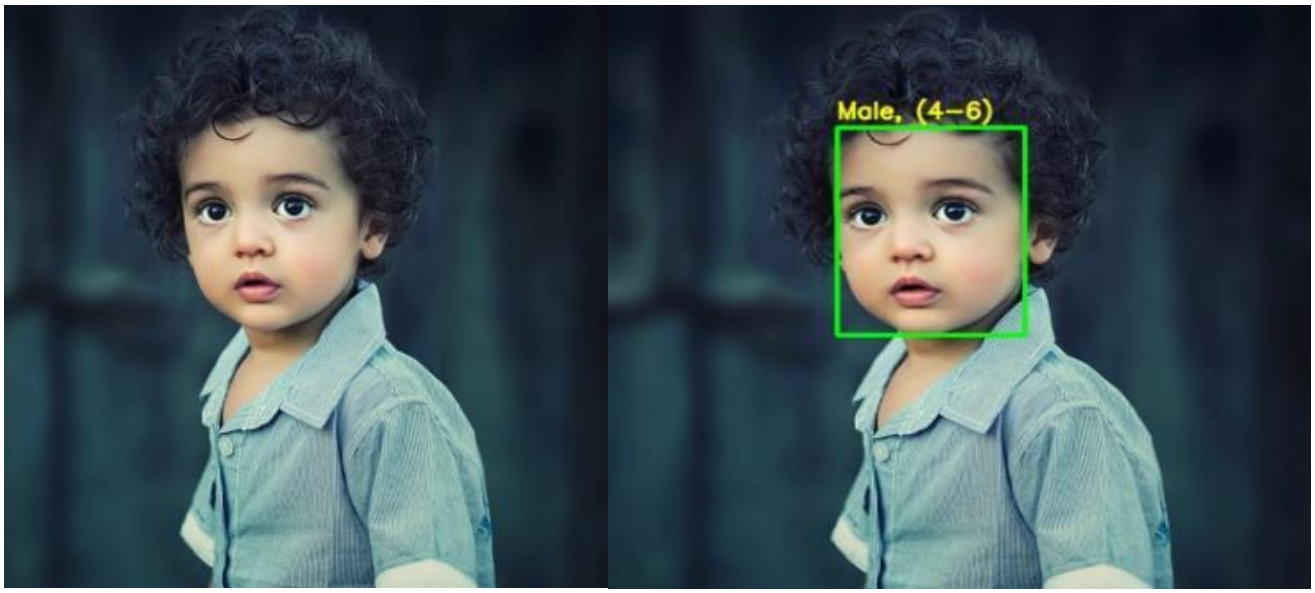
Screen 4.3: Group Image1



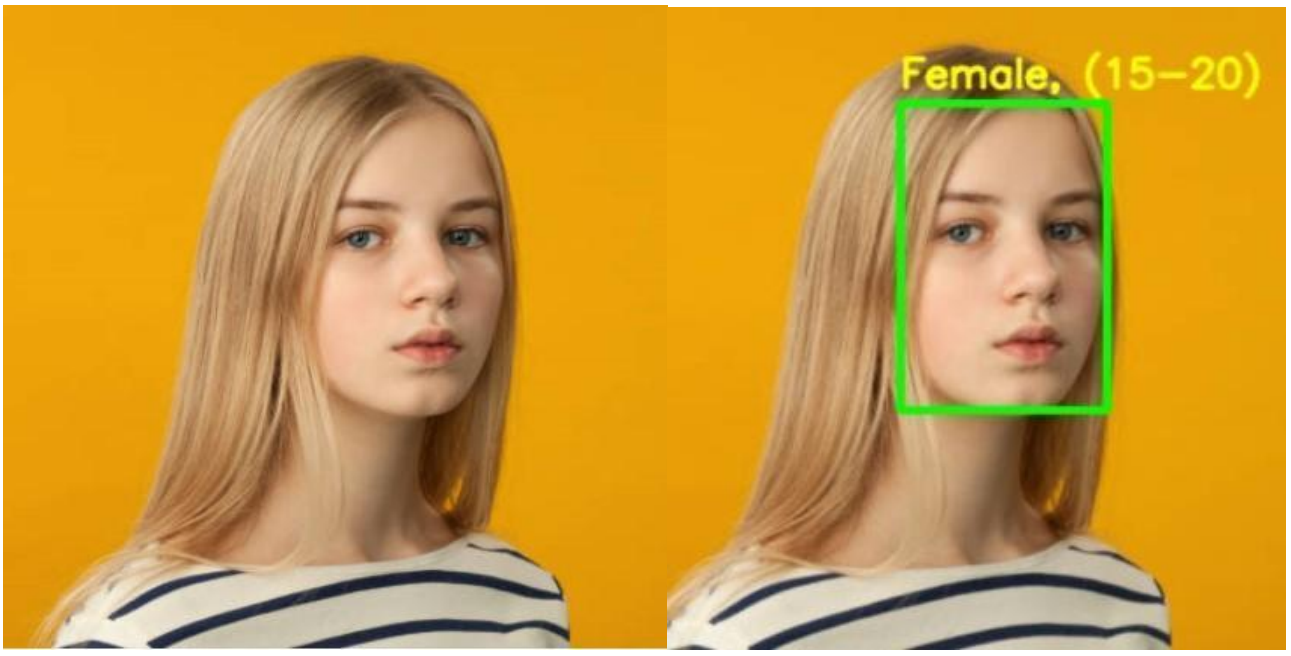
Screen 4.4: Group Image2



Screen 4.5: Group Image3



Screen 4.6: Child Age Image

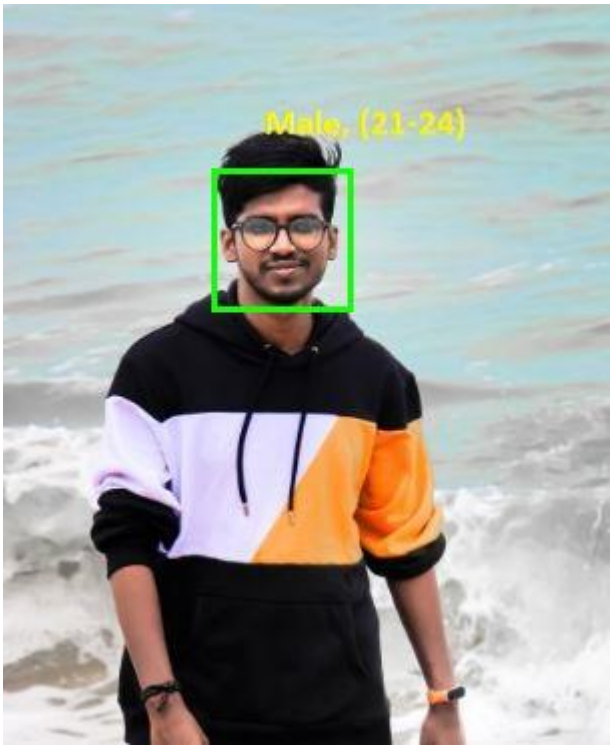


Screen 4.7: Teen Age Image

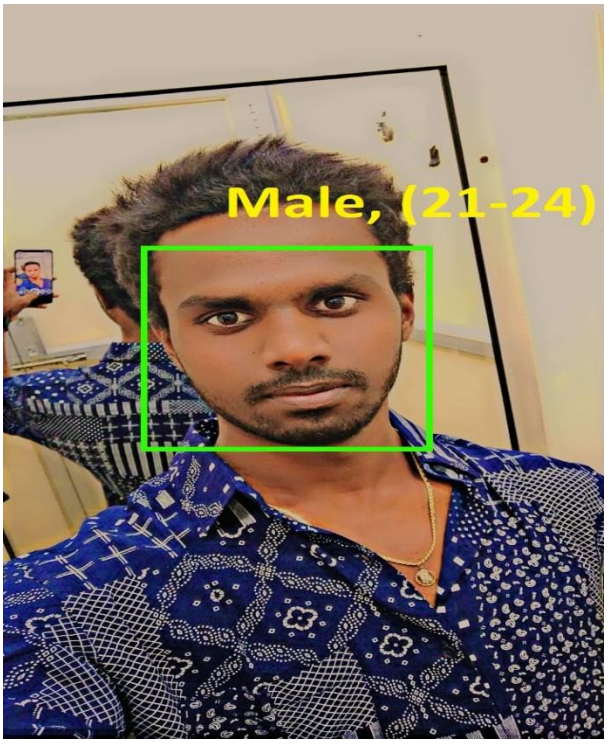
Screen 4.8: Middle Age Image

Screen 4.9: Old Age Image

Screen 4.10: Image With Object



Screen 4.11: Young Age Image-1



Screen 4.12: Young Age Image-2

5.TESTING & VALIDATION

5.1 Introduction

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. It is the major quality measure employed during software development. Testing is the exposure of the system to trial input to see whether it produces correct output. It is a process, which reveals errors in theprogram. During testing, the program is executed with a set of test cases and the output ofthe program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

5.1.1 Testing Phases:

Software testing phases include the following:

- Test activities are determined and test data selected.
- The test is conducted and test results are compared with the expected results. There are various types of Testing:

Unit Testing:

Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the module/program.

Integration Testing:

All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules.

System Testing:

It is mainly used if the software meets its requirements. The reference document for this process is the requirement document.

Acceptance Testing:

It is performed with realistic data of the client to demonstrate that the software is working satisfactorily.

5.1.2 Testing Methods :

Testing is a process of executing a program to find out errors. If testing is conducted 23 successfully, it will uncover all the errors in the software. Any testing can be done basing on two ways:

- White Box Testing:

  It is a test case design method that uses the control structures of the procedural design to derive test cases. In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. Using this testing a Software Engineer can derive the following test cases: Exercise all the logical decisions on either true or false sides. Execute all loops at their boundaries and within their operational boundaries. Exercise the internal data structures to assure their validity. White Box testing attempts to find errors in the following categories:

  - Guarantee that all independent paths have been executed.
  - Execute all logical decisions on their true and false Sides.
  - Execute all loops at their boundaries and within their operational bounds
  - Execute internal data structures to ensure their validity.

- Black Box Testing:

  It is a test case design method used on the functional requirements of the software. In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program It will help a software engineer to derive sets of input conditions that will exercise all the functional requirements of the program. Black Box testing attempts to find errors in the following categories:

  - Incorrect or missing functions
  - Interface errors
  - Errors in data structure or external database access
  - Performance errors
  - Initialization and termination errors

5.1.3 Testing Approach:

Testing can be done in two ways:

• Bottom-up approach

• Top-down approach

Bottom-up Approach:

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded with in the larger system. When bottom level modules are tested attention turns tothose on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

Top-down approach:

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module 17 shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower level module.

5.2 Design of Test Cases and Scenarios

Input	Actual Age and Gender	Output		Result
	Gender: Male Age: 38 year	Gender: Male	Age Group: (34-48)	Both Age and Gender are correct.
	Gender: Female Age: 32 years	Gender: Female	Age Group: (22-34)	Both Age and Gender are correct.
	Gender: Female Age: 3 years	Gender: Female	Age Group: (4-6)	Gender is correct whereas original age doesn't lie in the predicted agr group.
	Gender: Male Age: 28 years	Gender: Male	Age Group: (22-34)	Both Age and Gender are correct.
	Gender: Male Age: 88 years	Gender: Male	Age Group: (60-100)	Both Age and Gender are correct.
	No human face in this frame	Gender: Female	Age Group: (8-12)	Detects a human face like object as a legit human face.

Table 5.1: Test Cases

5.3 Validation

We experimented with two methods of using the network to produce age and gender predictions for novel faces:

- Center Crop: Feeding the network with the face image, cropped to 227 × 227 around the face center.
- Over-sampling: We extract five 227 × 227-pixel crop regions, four from the corners of the 256 × 256 face image, and an additional crop region from the center of the face.

The network is presented with all five images, along with their horizontal reflections. Its final prediction is taken to be the average prediction value across all these variations. We have found that small misalignments in the Adience images, caused by the many challenges of these images (conclusions, motion blur, etc.) can have a noticeable impact on the quality of our results. This second, over-sampling method, is designed to compensate for these small misalignments, bypassing the need for improving alignment quality, but directlyfeeding the network with multiple translated versions of the same face.

6.CONCLUSION & FUTURE ENHANCEMENT

6.1 Conclusion

Human Age and gender classification" are two of the many valuable information gathering resource from and individual. Human faces provide enough data which may be used for many purposes. To reach the correct audience human age and gender classification is very essential.

Here we tried to do the same process but with general equipment. The efficiency of the algorithm depends on several factor but the main motif of this project is being easy andfaster while also being as accurate as possible. Work is being done to the improve the efficiency of the algorithm. Some future improvements include discarding the face like non- human objects, more datasets for people belonging to different ethnic groups and more granular control over the workflow of the algorithm.

The task of recognizing age and gender, nonetheless, is an innately troublesome issue, more so than numerous other PC vision undertakings. The fundamental justification for thistrouble hole lies in the information needed to prepare these kinds of frameworks. While general article discovery errands can regularly approach many thousands or even large numbers of pictures for preparing, datasets with age and gender names are extensively more modest, as a rule in the large numbers or, best case scenario, several thousand.

6.2 Future Enhancement

There are several potential enhancements that can be made to age and gender detection using deep learning. Here are some possibilities:

1. Multi-task learning: Rather than training separate models for age and gender detection, a multi-task learning approach can be used to train a single model that can perform both tasks simultaneously. This can lead to better performance and faster inference times.
2. Data augmentation: Augmenting the training data can help improve the robustness of the model. For example, synthetic images can be generated with different lighting conditions, facial expressions, and camera angles.
3. Transfer learning: Transfer learning can be used to improve the performance of the model. For example, a pre-trained model can be fine-tuned on a smaller dataset of age and gender-labeled images.

4. Ensemble methods: An ensemble of models can be used to improve the overall accuracy of age and gender detection. This can involve training multiple models with different architectures, hyperparameters, or training data, and combining their predictions.
5. Attention mechanisms: Attention mechanisms can be used to help the model focus on the most relevant features for age and gender detection. This can improve the model's performance and also help with interpretability.
6. Explainability: Techniques for explaining the model's predictions can be developed to help users understand how the model arrived at its age and gender predictions. This can improve trust in the model and enable users to better understand its limitations.

7.BIBILOGRAPHY

7.1 References

[1] Vincenzo Carletti, Sai Greco, Gennaro Priyanka Cannella, and Mario Vento, "Age from Faces in the Deep Learning Revolution", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 42, NO. 9, 20 pages, SEPTEMBER 2020.

[2] [2] Y. Sun, M. Zhang, Z. Sun, and T. Tan, "Demographic analysis from biometric data: Achievements, challenges, and new frontiers," IEEE Trans. Pattern Anal. Mach.Intell., vol. 30, no. 2, pp. 332–351, Feb. 2018.

[3] [3] Y. Fu, G. Guo, and T. S. Huang, "Age synthesis and estimation via faces: A survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 11, pp. 1955–1976, Nov. 2010.

[4] [4] ARWA S. AL-SHANNAQ AND LAMIAA A. ELREFAEI, "Comprehensive Analysis of the Literature for Age Estimation from Facial Images", ACCESS.2019.2927825, vol 7, July 2019.

[5] [5] KAZEMI, Vahid; SULLIVAN, Josephine. One Millisecond Face Alignment with an Ensemble of Regression Trees. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition.

[6] [6] ROTHE, Rasmus; TIMOFTE, Radu; GOOL, Luc Van. DEX: Deep Expectation of apparent age from a Single Image.