ANALYSIS AND DETECTION OF AGE AND GENDER USING DEEP LEARNING HEURISTIC

A Project report submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

Submitted by

MD HAFSHA FIRDOUS (317126510152)

LALAM NIKHIL (317126510149)

R BHANU PRAVEEN (317126510161)

Under the guidance of

G SATHAR

ASSISTANT PROFESSOR, CSE DEPARTMENT

PRAGNANAM BRAHMA

ANITS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)

Sangivalasa, Bheemili Mandal, Visakhapatnam dist. (A.P)

2020-2021

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **G. SATHAR** Assistant Professor, Department of Computer Science and Engineering, ANITS, for his guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. R. SIVARANJANI**, Head of the Department, Computer Science and Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to Project Coordinator **Dr. V. USHA BALA**, for her Continuous support and encouragement. We thank all **teaching faculty** of Department of CSE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **SK. SHAHJAHAN** of the Department of CSE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

Md. Hafsha Firdous (317126510152),

Lalam Nikhil (317126510149),

R. Bhanu Praveen (317126510161),

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

(***Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade***)

Sangivalasa, Bheemili mandal, Visakhapatnam dist. (A.P)

PRAGNANAM BRAHMA

ANITS

CERTIFICATE

This is to certify that the project report entitled **"ANALYSIS AND DETECTION OF AGE AND GENDER USING DEEP LEARNING HEURISTIC"** submitted by **MD HAFSHA FIRDOUS (317126510152), LALAM NIKHIL (317126510149), R BHANU PRAVEEN (317126510161)** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science Engineering** of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Project Guide	**Head of the Department**
G SATHAR	**Dr. R. SIVARANJANI**
Assistant Professor	Professor & HOD
Department of CSE	Department of CSE
ANITS	ANITS

DECLARATION

We, **MD HAFSHA FIRDOUS (317126510152), LALAM NIKHIL (317126510149), R BHANU PRAVEEN (317126510161),** of final semester B.Tech., in the department of Computer Science and Engineering from ANITS, Visakhapatnam, hereby declare that the project work entitled **ANALYSIS AND DETECTION OF AGE AND GENDER USING DEEP LEARNING HEURISTIC** is carried out by us and submitted in partial fulfillment of the requirements for the award of **Bachelor of Technology in Computer Science Engineering** , under Anil Neerukonda Institute of Technology & Sciences(A) during the academic year 2017-2021 and has not been submitted to any other university for the award of any kind of degree.

MD HAFSHA FIRDOUS	**317126510152**
LALAM NIKHIL	**317126510149**
R BHANU PRAVEEN	**317126510161**

ABSTRACT

When a person is uniquely identified then it is because of the face which is the crucial part. With the help of a face, different people are classified and also besides these, a large number of applications can be implemented like for security purposes at banks, various organizations and also in the areas where there is a large public gathering. As the raise in usage of social media and social platforms reached up in the air, age and gender detection became prominent. The attribute information such as age and gender improves the performance of face recognition. This project proposes age and gender detection method from face images using Deep-convolutional neural network(CNN). In this study, face images of persons are trained using CNN. Training of deep models shows exceptional performance with large datasets, but they are not suitable for learning from few samples. The input faces are compared with the images in the data set and will be recognized. There are many methods which have been proposed in the literature for age estimation and gender classification. However, all of them still have a disadvantage such as partial reflection about face structure and face texture. This technique applies to both face alignment and recognition and significantly improves these two aspects. To this end, we propose a simple convolutional network architecture that can be used even when the amount of learning data is limited.

Keywords: face recognition, attribute information, Deep-Convolutional neural networks, gender-classification, age-classification.

LIST OF FIGURES

LIST OF ABBREVATIONS

2-D	2-dimensional
3-D	3-dimensional
AI	Artificial intelligence
RELU	Rectified linear activation function
NN	Neural Network
ANN	Artificial Neural Network
CNN	Convolutional Neural Networks
DL	Deep learning
OpenCV	Open Source Computer Vision Library
CV	Computer Vision
SVM	Support Vector Machine
ML	Machine Learning
UML	Unified Modelling Language
DFD	Data Flow Diagram

TABLE OF CONTENTS

1. INTRODUCTION

Facial analysis has gained much recognition in the computer vision community in the recent past. Age and gender, two of the key facial attributes, play a very foundational role in social interactions, making age and gender estimation from a single face image an important task in intelligent applications, such as access control, human-computer interaction, law enforcement, etc. We formulate the age and gender classifications task as a classification problem in which the CNN model learns to predict the age and gender from a face image. We need to propose a model that uses CNN architecture to predict the age group and gender of human's faces from unfiltered real-world environments. The CNN approach addresses the age and gender labels as a set of discrete annotations and train the classifiers that predict the human's age group and gender. Then we design a quality and robust image preprocessing algorithm that prepares and preprocesses the unfiltered images for the CNN model and this greatly has a very strong impact on the performance accuracy of our age and gender classifiers. We demonstrate that pertaining on large-scale datasets allows an effective training of our age and gender CNN model which enable the classifiers to generalize on the test images and then avoid overfitting. Finally, UTKFace dataset is used to evaluate the performance of the CNN model, and despite the very challenging nature of the images in the dataset, the approach produces significant improvements in age group and gender classification accuracy. Face recognition techniques described in the last few years have shown that tremendous progress can be made by the use of deep convolutional neural networks (CNN). We demonstrate similar gains with a simple network architecture, designed by considering the rather limited availability of accurate age and gender labels in existing face data sets.

Advantages of CNN:

- Processing speed.
- Flexible and Robust
- Versatile in nature / Dynamic Behavior.

Applications of CNN:

- Decoding Facial Recognition.
- Analyzing Documents.
- Understanding Climate

1.1 NEURAL NETWORKS

Neural Network (or Artificial Neural Network) has the ability to learn by examples. ANN is an information processing model inspired by the biological neuron system. ANN biologically inspired simulations that are performed on the computer to do a certain specific set of tasks like clustering, classification, pattern recognition etc. It is composed of a large number of highly interconnected processing elements known as the neuron to solve problems. It follows the non-linear path and process information in parallel throughout the nodes. A neural network is a complex adaptive system. Adaptive means it has the ability to change its internal structure by adjusting weights of inputs.

Artificial Neural Networks can be best viewed as weighted directed graphs, where the nodes are formed by the artificial neurons and the connection between the neuron outputs and neuron inputs can be represented by the directed edges with weights. The ANN receives the input signal from the external world in the form of a pattern and image in the form of a vector. These inputs are then mathematically designated by the notations x(n) for every n number of inputs. Each of the input is then multiplied by its corresponding weights (these weights are the details used by the artificial neural networks to solve a certain problem). These weights typically represent the strength of the interconnection amongst neurons inside the artificial neural network.
All the weighted inputs are summed up inside the computing unit (yet another artificial neuron).

If the weighted sum equates to zero, a bias is added to make the output non-zero or else to scale up to the system's response. Bias has the weight and the input to it is always equal to 1. Here the sum of weighted inputs can be in the range of 0 to positive infinity. To keep the response in the limits of the desired values, a certain threshold value is benchmarked. And then the sum of weighted inputs is passed through the activation function. The activation function is the set of transfer functions used to get the desired output of it. There are various flavours of the activation function, but mainly either linear or non-linear set of functions. Some of the most commonly used set of activation functions are the Binary, Sigmoid (linear) and Tan hyperbolic sigmoidal (non-linear) activation functions.

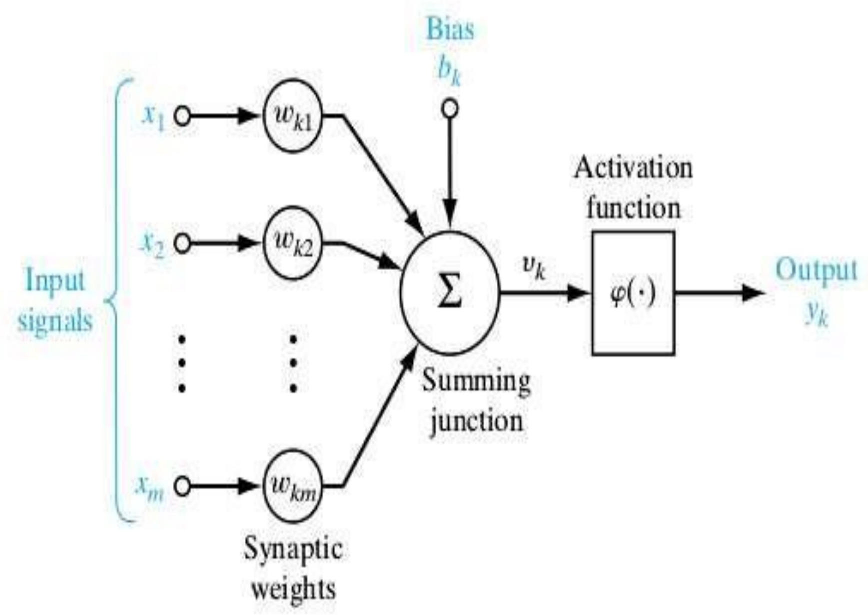


Fig 1: Basic Neural Network

The Artificial Neural Network contains three layers:

1. **Input Layer:** The input layers contain those artificial neurons (termed as units) which are to receive input from the outside world. This is where the actual learning on the network happens or corresponding happens else it will process.

2. **Hidden Layer:** The hidden layers are mentioned hidden in between input and the output layers. The only job of a hidden layer is to transform the input into something meaningful that the output layer/unit can use in some way. Most of the artificial neural networks are all interconnected, which means that each of the hidden layers is individually connected to the neurons in its input layer and also to its output layer leaving nothing to hang in the air. This makes it possible for a complete learning process and also learning occurs to the maximum when the weights inside the artificial neural network get updated after each iteration.

3. **Output Layer:** The output layers contain units that respond to the information that is fed into the system and also whether it learned any task or not.

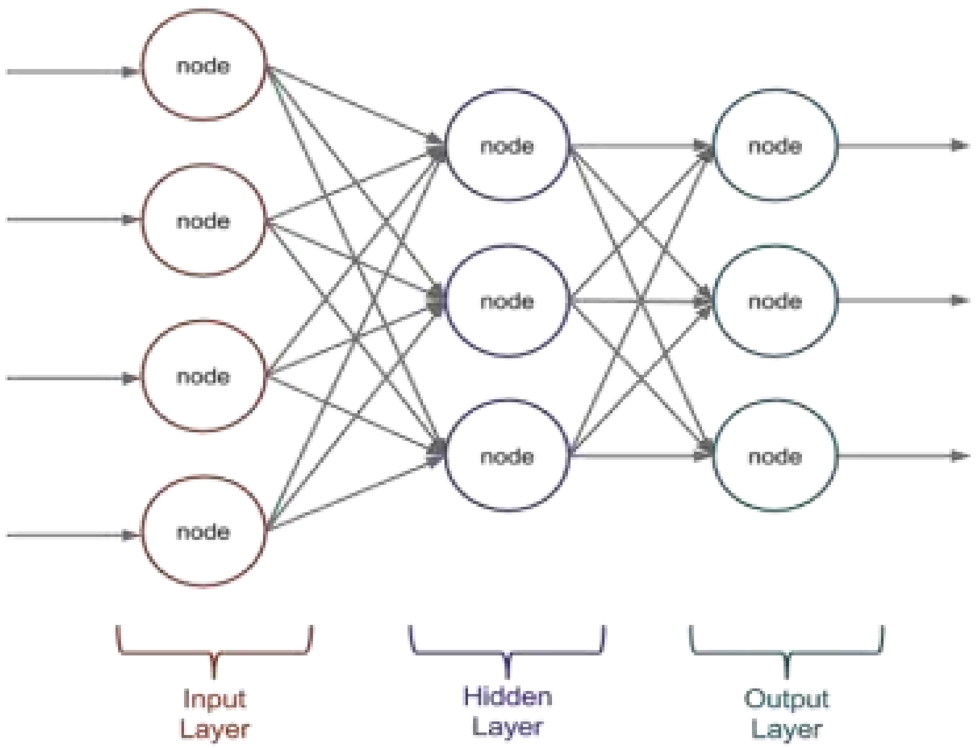


Fig 2: Layers of Neural Network

LEARNING PROCESS OF A NEURAL MODEL:

1. Start with values for the network parameters (wij weights and bj biases).
2. Take a set of examples of input data and pass them through the network to obtain their prediction.
3. Compare these predictions obtained with the values of expected labels and calculate the loss with them.
4. Perform the backpropagation in order to propagate this loss to each and every one of the parameters that make up the model of the neural network.
5. Use this propagated information to update the parameters of the neural network with the gradient descent in a way that the total loss is reduced, and a better model is obtained.
6. Continue iterating in the previous steps until we consider that we have a good model.

ARCHITECTURES OF NEURAL NETWORKS:

Artificial Neural Networks (ANNs) make up an integral part of the Deep Learning process. They are inspired by the neurological structure of the human brain. ANNs are "complex computer code written with the number of simple, highly interconnected processing elements which is inspired by human biological brain structure for simulating human brain working & processing data (Information) models."

Feed Forward Networks: Feed-forward ANNs allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down. to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down.

Feedback/Recurrent Networks: Feedback networks can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic. They change continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organization.

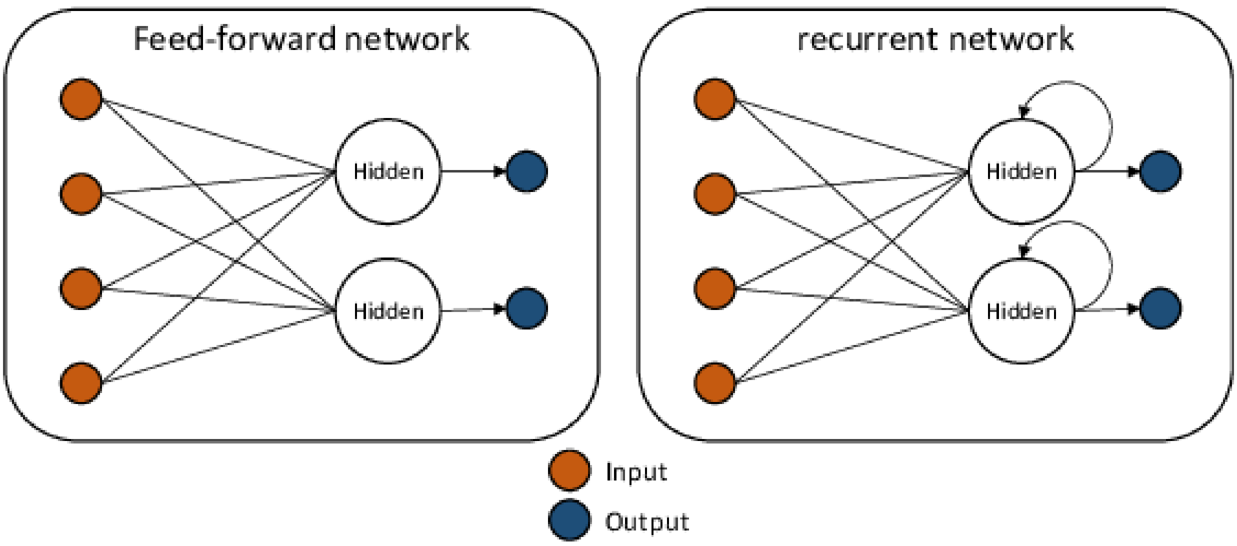


Fig 3: Types of Network Architecture

NEURAL NETWORKS VERSUS CONVENTIONAL COMPUTERS:

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem-solving capability of conventional computers to problems that we already understand and know how to solve. Neural networks process information in an equivalent way the human brain does. The network is composed of many highly-interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task.

On the other hand, conventional computers use a cognitive approach to problem solving; the way the problem is to solved must be known and stated in small unambiguous instructions. These instructions are then converted to a high-level language program and then into machine code that the computer can understand. These machines are totally predictable; if anything goes wrong is due to a software or hardware fault. Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks, require systems that use a combination of the two approaches in order to perform at maximum efficiency.

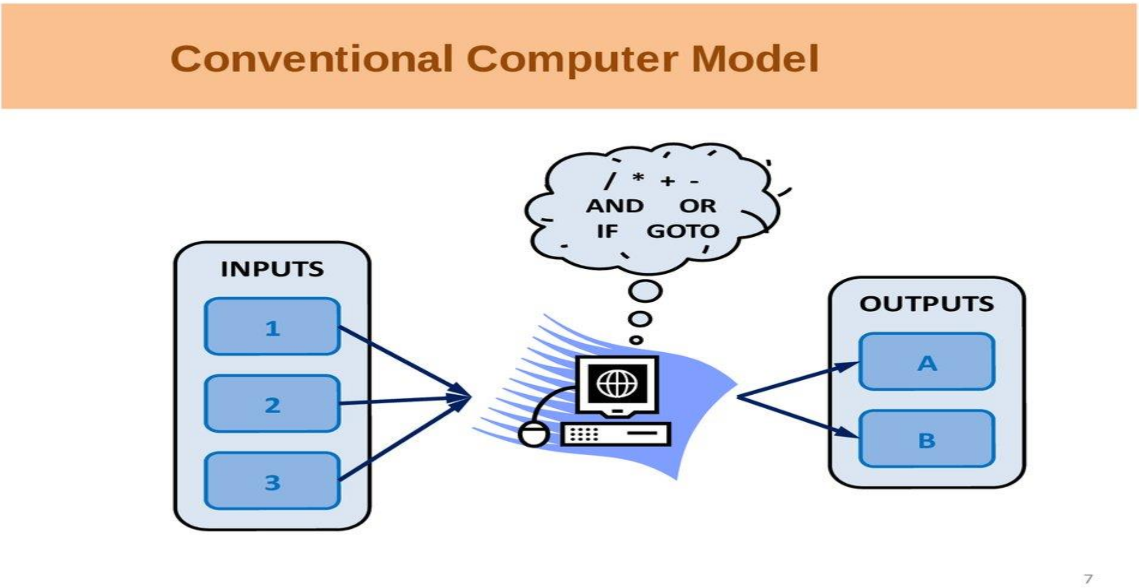


Fig 4: Conventional Computer Model

1.2 DEEP LEARNING

Deep learning is a branch of machine learning which is completely based on artificial neural networks. Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabelled. It has a greater number of hidden layers and known as deep neural learning or deep neural network.

Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, ecommerce platforms, and online cinemas, among others. However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unravelling this wealth of information and are increasingly adapting to AI systems for automated support.

Deep learning learns from vast amounts of unstructured data that would normally take humans decades to understand and process. Deep learning and utilizes a hierarchical level of artificial neural networks to carry out the process of machine learning. The artificial neural networks are built like the human brain, with neuron nodes connected like a web. While traditional programs build analysis with data in a linear way, the hierarchical function of deep learning systems enables machines to process data with a nonlinear approach. Deep Neural Network is a neural network with a certain level of complexity (having multiple hidden layers in between input and output layers). They are capable of modelling and processing non-linear relationships.

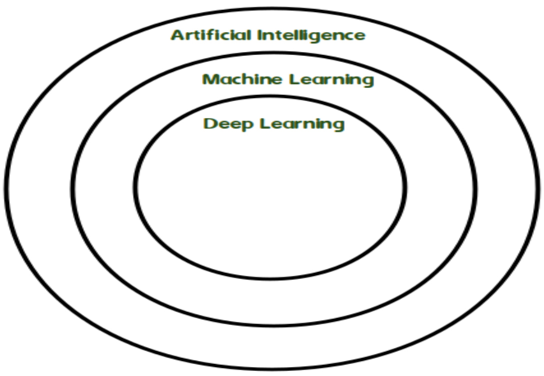


Fig 5: Architecture of Deep learning

WORKING OF DEEP LEARNING:

- First, we need to identify the actual problem in order to get the right solution and it should be understood, the feasibility of the Deep Learning should also be checked (whether it should fit Deep Learning or not).
- Second, we need to identify the relevant data which should correspond to the actual problem and should be prepared accordingly.
- Third, Choose the Deep Learning Algorithm appropriately.
- Fourth, Algorithm should be used while training the dataset. Fifth, Final testing should be done on the dataset.

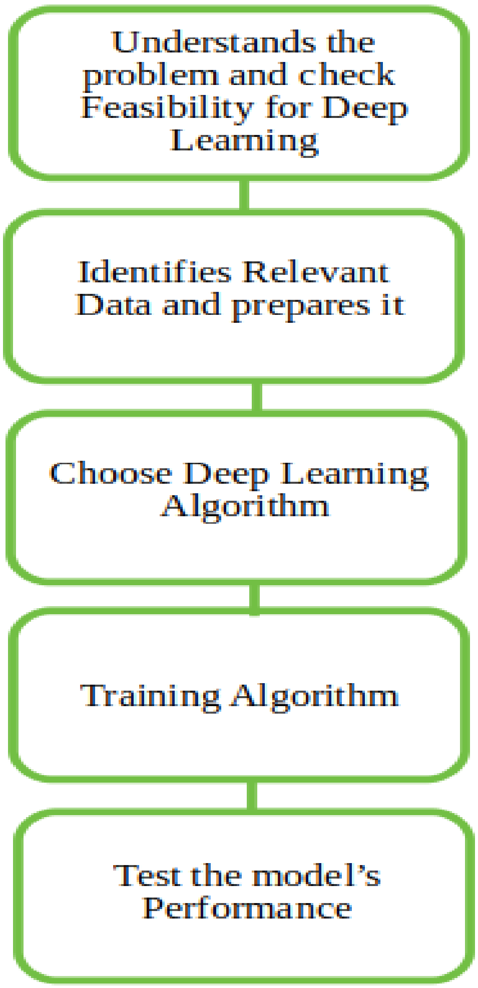


Fig 6: Flow chart of Deep Learning

DEEP LEARNING LAYERS:

- The Input Layer
- The Hidden Layer
- The Output Layer

1.The Input layer:

It receives all the inputs and the last layer is the output layer which provides the desired output.

2. Hidden Layers:

All the layers in between these layers are called hidden layers. There can be n number of hidden layers. The hidden layers and perceptron's in each layer will depend on the use-case you are trying to solve.

3. Output Layers:

It provides the desired output. In this project, we will use Deep Learning to accurately identify the gender and age of a person from a single image of a face. The predicted gender may be one of 'Male' and 'Female', and the predicted age may be from 0-100. It is very difficult to accurately guess an exact age from a single image because of factors like makeup, lighting, obstructions, and facial expressions. And so, we make this a classification problem instead of making it one of regression.

ADVANTAGES:

1. No Need for Feature Engineering

Feature engineering is the process of extracting features from raw data to better describe the underlying problem. It is a fundamental job in machine learning as it improves model accuracy. The process can sometimes require domain knowledge about a given problem.

2. Best Results with Unstructured Data

According to research from Gartner, up to 80% of a company's data is unstructured because most of it exists in different formats such as texts, pictures, pdf files and more. Unstructured data is hard to analyse for most machine learning algorithms, which means it's also going unutilized. That is where deep learning can help. Deep learning algorithms can be trained using different data formats, and still derive insights that are relevant to the purpose of its training.

3. No Need for Labelling of Data

Getting good-quality training data is one of the biggest problems in machine learning because data labelling can be a tedious and expensive job. Sometimes, the data labelling process is simple but time-consuming. For example, labelling photos "dog" or "muffin" is an easy task, but an algorithm needs thousands of pictures to tell the difference. Other times, data labelling may require the judgments of highly skilled industry experts, and that is why, for some industries, getting high-quality training data can be very expensive.

4. Efficient at Delivering High-quality Results

Humans need rest and fuel. They get tired or hungry and make careless mistakes. That is not the case for neural networks. Once trained correctly, a deep learning brain can perform thousands of repetitive, routine tasks within a shorter period of time than it would take a human being. The quality of its work never diminishes, unless the training data includes raw data that does not represent the problem you are trying to solve.

APPLICATIONS:

- Self-Driving Cars
- News Aggregation and Fraud News Detection
- Natural Language Processing
- Entertainment
- Visual Recognition
- Fraud Detection
- Healthcare
- Automatic Game Playing
- Language Translations
- Pixel Restoration
- Photo Descriptions
- Demographic and Election Predictions

1.3 MOTIVATION WORK

Age and gender classification play a very important role in our social lives, by which we can find whether the persons we contact are “sir” or “madam” and young or old. These behaviors are heavily dependent on our ability to estimate these individual traits: age and gender, which are from facial appearances. These attributes are important in our lives while the ability to estimate them accurately and reliably from facial appearance is still far from satisfying the needs of commercial applications. The number of crimes has been increasing daily at a much faster rate. It has become a necessity to identify criminals as soon as possible. The traditional way of identification is a slow process while the proposed approach can be used to counter terrorism by identifying the features at a much faster rate. The project can also be used to overcome the frauds that can take place during voting i.e. can be used for voter identification. The old generation has the difficulty to operate computers with ease. This bridge can be lessened by improving Human-Computer Interaction (HCI). The child molestation cases can be tackled at a faster rate by comparing school surveillance camera images to know child molesters and the same can be used for verifying the court records thereby minimizing victim trauma. Similarly, it can also be used for surveillance at banks and residential areas.

1.4 PROBLEM STATEMENT

Automatically detecting the gender of a person or estimating his/her age is valuable in many fields of work. In the area of data science, there exists a variety of possible approaches. The goal of this project is to train and evaluate a given convolutional neural network to accomplish gender classification and age estimation on images. This comprises certain tasks as finding and processing a suitable dataset for evaluating, training and testing, as well as evaluating the solution.

2. LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system, the above consideration is taken into account for developing the proposed system.

2.1 Face Processing Changes in Normal Aging Revealed by fMRI Adaptation:

Yunjo Lee proposed that the fMRI method is used to study upon age detection methods. The study involves a proper recording of the variations of people on the basis of their changes according to age, gender, identity and other features. The brain activation tasks related to face matching are performed and tested outside the scanner. There was a same result in face processing in older as well as young adults. The performance results high in both the cases having same facial viewpoints. The aging of the elders is not based on any one factor. It is combination of various factors that result in accountancy of such results. The results need to be kept a track on which are based on all credentials kept in certain environments.

2.2 Using Artificial Neural Network for Human Age Estimation Based on Facial Images:

Sarah N. Kohail proposed that the age estimation is now the current challenge being faced. Here, the article puts forward the approach of neural networks to estimate the age of humans. The main change that has been made in this method is the fine tuning of the age ranges. To learn the multi-layer perception neural networks (MLP) the facial features of the new images were extracted and recorded. The results have shown the MLP method as a good method with minimum errors in the results. These results can be used in many of the applications like age-based access control applications and also in the age adaptive human machine interaction. The upgradations are to be made in the system, where the system is to be made more automatic and also the numbers of input features to be provided are to be reduced.

2.3 Facial Age Estimation by Conditional Probability Neural Network:

Chao Yin proposed that the Conditional Probability Neural Network (CPNN) is a distribution learning algorithm used for the age estimation using facial expressions. It follows the three-layer neural network system in which the target values and the conditional feature vectors are used as an input. This can help it in learning the real ages. The relationship between the face image and the related label distribution through the neural network is used as the learning method for this system. CPNN has proved to be providing better results than all the previously made methods. Through this method the results provided were very easy, there was less computational involved and the outcomes very efficient. Due to all such advantages it was preferred more than the others.

2.4 Age Classification System with ICA Based Local Facial Features:

Hang Qi proposed that various techniques have been arising for the detection of faces which can also identify the age of the person. Here, an automated system has been proposed which can classify the age and help distinguishing kids face from that of an adults face. There are three parts that the system encompasses. They are face detection, face alignment and normalization, and age classification. Face samples are created by the normal face detection and alignment methods. ICA is used for the extraction of the local facial components that are present in the images. This system has been proved to be much faster and the results are efficient. So this system can be used in future as a prototype.

2.5 Neural networks for detection and classification of walking pattern changes due to ageing:

R. Begg proposed a methodology, that the aging through the artificial neural networks will change the walking by using automatic recognition is the aim of the article. The balance control of the locomotors system will get disturbed due to the manner of walking which are caused through patterns which are generated according to the increasing age. There are many good reasons to use such techniques. The first one was standard back propagation and second one was scaled conjugate gradient and last one was back propagation with the help of Bayesian regularization ware three methods used. The three networks came out with improved results but from above all methods Bayesian regularization method was the one of the better methods which gave greater result in some of the fields. The neural networks thus are a one of the best method to find age identification process.

3. PROPOSED METHODOLOGY

CNNs are a type of feed-forward neural networks made up of many layers. CNNs consist of filters or kernels or neurons that have learnable weights or parameters and biases. Each filter takes some inputs, performs convolution and optionally follows it with a non-linearity. The structure of CNN contains Convolutional, pooling, Rectified Linear Unit (ReLU), and Fully Connected layers.

Methodology consists of five phases namely:

- Real Time dataset
- Pre Processing
- Normalization
- Feature Extraction that extracts the unique features such as skin texture, beard, moustache, hair.
- Classification is the combination of both gender and age predictions which has certain classes in it.

Real Time Dataset:

Initially, the Webcam is triggered by the capture function according to the code functioned using Python framework. Once the face is detected by Webcam it recognizes the face and captures it which will be preprocessed later.

Pre-processing:

The face image of a person is captured by a digital camera. Pre-processing includes three steps as detecting the image, converting to gray scale & noise reduced image. The color image is converted into gray scale image. The Matlab code is used for conversion of RGB to Gray scale image represented in binary digits 0 and 1. There are different types of filtration methods used for noise reduction techniques. Gaussian filtering method is used for noise reduction.

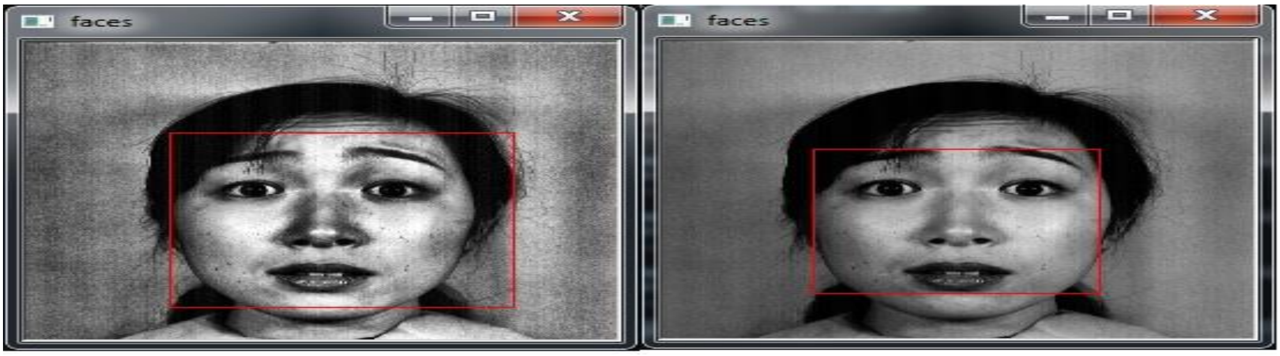


Fig 7: Pre-processing of Face

Normalization:

In normalization process the system crop the detected rectangular face area using Matlab in-built object function. Then, detect the eye pair, mouth, nose, and chin. It gives the specific images of left eye, right eye, left eyebrow, right eyebrow, mouth i.e. image of lips & also detects chin hair line part of face image and also gives the nose image.

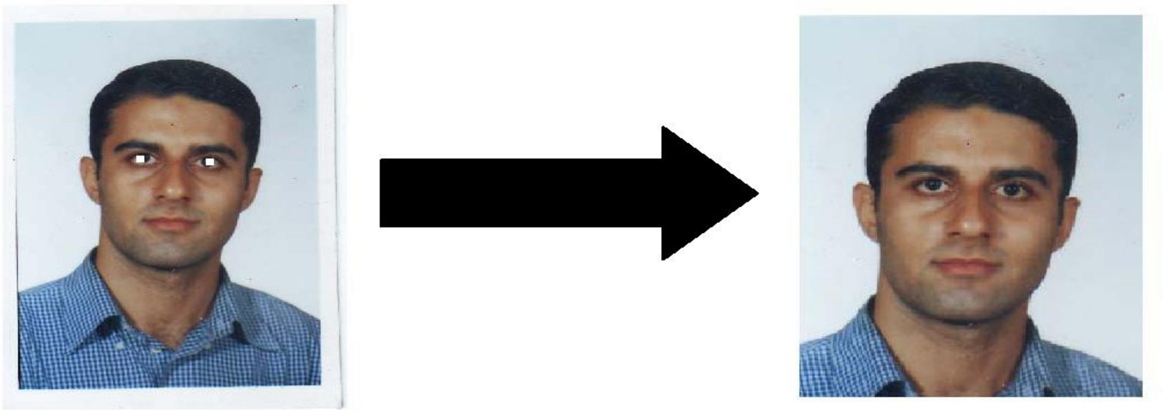


Fig 8: Normalization of Face

Feature Extraction:

A combination of global and grid features is extracted from face images. The global features such as distance between two eye balls, eye to nose tip, eye to chin, and eye to lip is calculated using four distance values, four features are calculated.

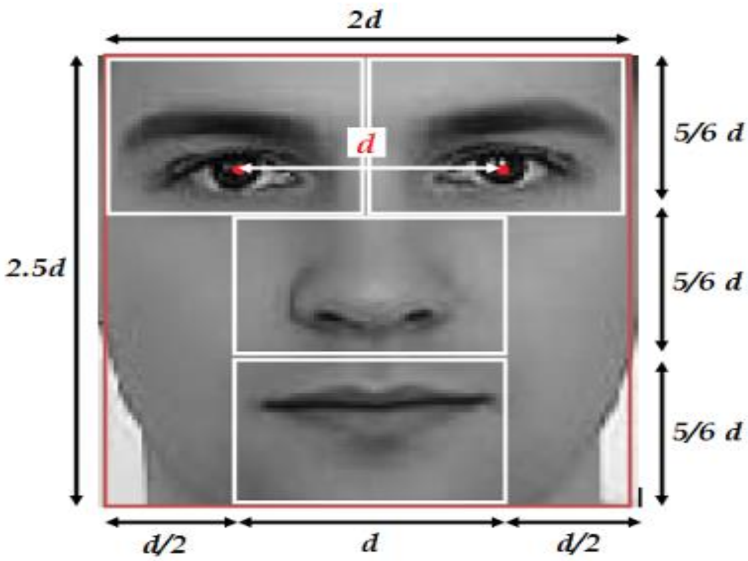


Fig 9: Feature Extraction

1)Gender Detection:

Gender Detection is the procedure followed by Pre Processing and afterward gender (male or female) is lead to identification. The gender determination phase comprises of two fundamental characterizations. By prompting the CNN's (Convolutional Neural Network) calculation the comparison is to be held right now finding the least complex way to gender discovery.

For example, Length of the hair, Beard, Moustache and skin texture. Every single highlights which are extricated by CNN's calculation.

2)Age prediction:

Age Prediction consists of eight several classifications, they are (0-2), (4-6), (8-12), (15-20), (25-32), (38-43), (48-53) and (60-100). By CNN’s algorithm the comparison is to be held to predict the age of the particular individual. The main feature which is extracted to predict the age is skin texture, as it varies for different age group. For example, the skin texture of a kid is comparatively softer than the aged people and then the aged people and youngsters. In which the frame is captured while the face is detected in real time video format and then it converts the captured face image to a gray scale image and then age is predicted by extracting the texture of the skin.

Classification:

Age ranges and gender are classified dynamically depending on number of groups based on the features. The Softmax classifier is a probability distribution function that turns a vector of K real values into a vector of K real values that sum to 1. The input values can be positive, negative, zero, or greater than one, but it transforms them into values between 0 and 1, so that they can be interpreted as probabilities. If one of the inputs is small or negative, the softmax turns it into a small probability, otherwise, it turns it into a large probability, but it will always remain between 0 and 1.

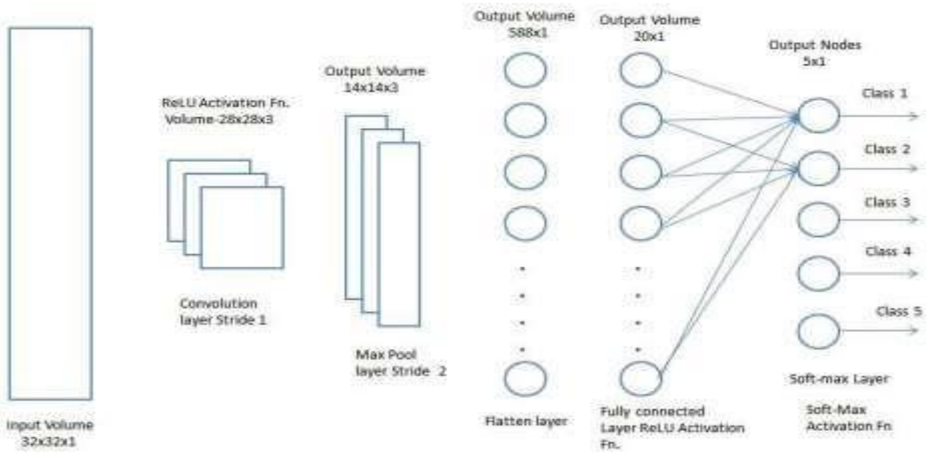


Fig 10: Fully Connected Layer

3.1 FACE DETECTION:

The Viola-Jones Algorithm, developed in 2001 by Paul Viola and Michael Jones, the Viola-Jones algorithm is an object-recognition framework that allows the detection of image features in real-time. Viola-Jones is quite powerful and its application has proven to be exceptionally notable in real-time face detection. The framework is still a leading player in face detection alongside many of its CNNs counter parts. The Viola-Jones Object Detection Framework combines the concepts of **Haar-like Features, Integral Images, the AdaBoost Algorithm, and the Cascade Classifier** to create a system for object detection that is fast and accurate.

Viola-Jones was designed for frontal faces, so it is able to detect frontal the best rather than faces looking sideways, upwards or downwards. Before detecting a face, the image is converted into grayscale, since it is easier to work with and there's lesser data to process. The Viola-Jones algorithm first detects the face on the grayscale image and then finds the location on the coloured image.

Viola-Jones outlines a box (as you can see on the right) and searches for a face within the box. It is essentially searching for these haar-like features, which will be explained later. The box moves a step to the right after going through every tile in the picture. In this case, I've used a large box size and taken large steps for demonstration, but in general, you can change the box size and step size according to your needs. With smaller steps, a number of boxes detect face-like features (Haar-like features) and the data of all of those boxes put together, helps the algorithm determine where the face is.

3.1.1 Haar-like Features

Haar-like features are named after Alfred Haar, a Hungarian mathematician in the 19th century who developed the concept of Haar wavelets (kind of like the ancestor of haar-like features). The features below show a box with a light side and a dark side, which is how the machine determines what the feature is. Sometimes one side will be lighter than the other, as in an edge of an eyebrow. Sometimes the middle portion may be shinier than the surrounding boxes, which can be interpreted as a nose.

There are 3 types of Haar-like features that Viola and Jones identified in their research:

- Edge features
- Line-features
- Four-sided features

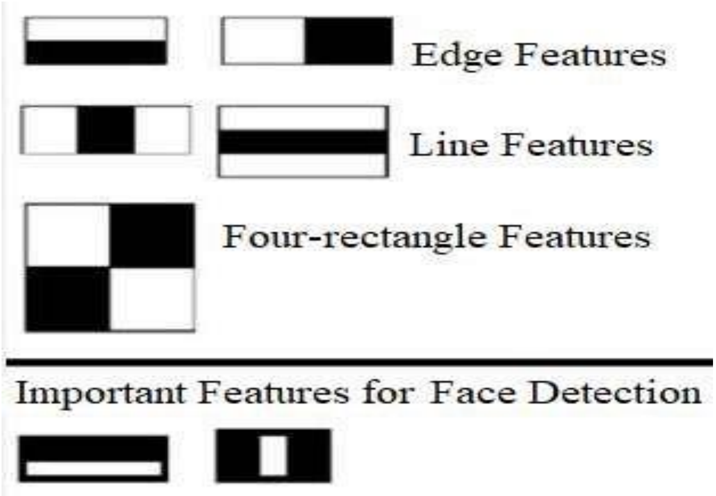


Fig 11: Haar-like Features

These features help the machine understand what the image is. Imagine what the edge of a table would look like on a b&w image. One side will be lighter than the other, creating that edge like b&w feature as you can see in the picture above. In the two important features for Face Detection, the horizontal and the vertical features describe what eyebrows and the nose, respectively, look like to the machine. Additionally, when the images are inspected, each feature has a value of its own. It's quite easy to calculate: Subtract White area from the Black area. For example, look at the image below.

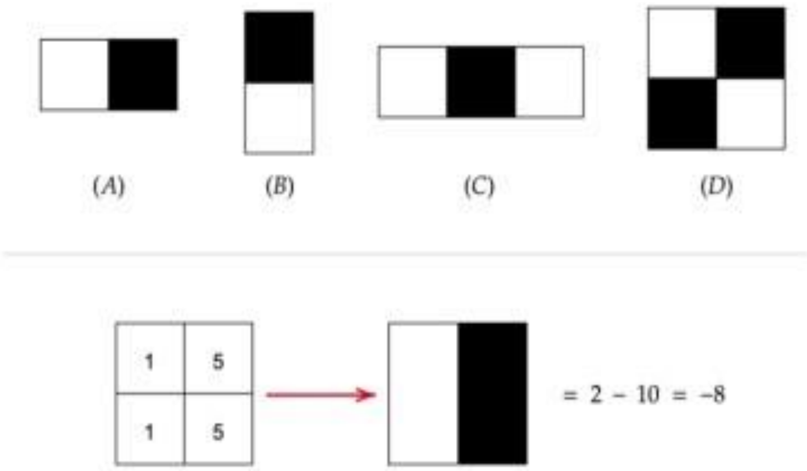


Fig. 12: Feature Value Calculation

3.1.2 Integral Image

We calculated the value of a feature. In reality, these calculations can be very intensive since the number of pixels would be much greater within a large feature. The integral image plays its part in allowing us to perform these intensive calculations quickly so we can understand whether a feature of a number of features fit the criteria. To calculate the value of a single box in the integral image, we take the sum of all the boxes to its left.

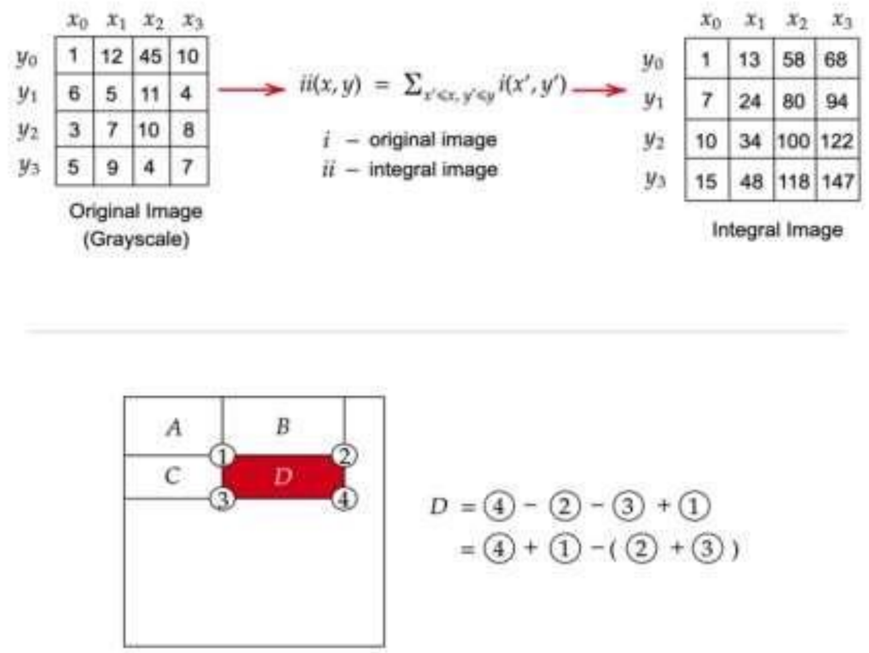


Fig 13: Integral Image

Use of Integral Image:

Haar-like features are actually rectangular, and the integral image process allows us to find a feature within an image very easily as we already know the sum value of a particular square and to find the difference between two rectangles in the regular image, we just need to subtract two squares in the integral image. So even if you had 1000 x 1000 pixels in your grid, the integral image method makes the calculations much less intensive and can save a lot of time for any facial detection model.

3.1.3 Adaptive Boosting (AdaBoost)

The AdaBoost (Adaptive Boosting) Algorithm is a machine learning algorithm for selecting the best subset of features among all available features. The output of the algorithm is a classifier (Prediction Function, Hypothesis Function) called a "Strong Classifier". A Strong Classifier is made up of a linear combination of "Weak Classifiers" (best features). From a high level, in order to find these weak classifiers, the algorithm runs for T iterations where T is the number of weak classifiers to find and it is set by you. In each iteration, the algorithm finds the error rate for all features and then choose the feature with the lowest error rate for that iteration.

The algorithm learns from the images we supply it and is able to determine the false positives and true negatives in the data, allowing it to be more accurate. We would get a highly accurate model once we have looked at all possible positions and combinations of those features. Training can be super extensive because of all the different possibilities and combinations you would have to check for every single frame or image.

Let's say we have an equation for our features that determines the success rate (as seen in the image), with f1, f2 and f3 as the features and a1, a2, a3 as the respective weights of the features. Each of the features is known as a weak classifier. The left side of the equation F(x) is called a strong classifier. Since one weak classifier may not be as good, we get a strong classifier when we have a combination of two or three weak classifiers. As you keep adding, it gets stronger and stronger. This is called an ensemble.

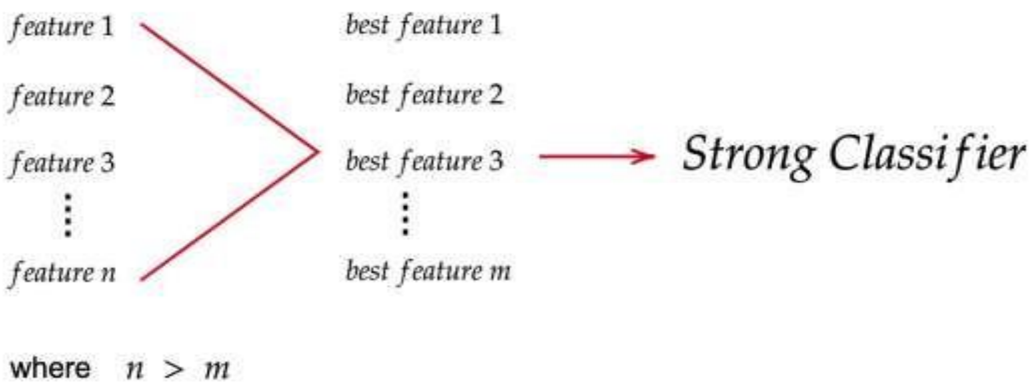


Fig 14: Adaptive Boosting

3.1.4 The Cascade Classifier

A Cascade Classifier is a multi-stage classifier that can perform detection quickly and accurately. Each stage consists of a strong classifier produced by the AdaBoost Algorithm. From one stage to another, the number of weak classifiers in a strong classifier increases. An input is evaluated on a sequential (stage by stage) basis. If a classifier for a specific stage outputs a negative result, the input is discarded immediately. In case the output is positive, the input is forwarded onto the next stage.

According to Viola & Jones (2001), this multi-stage approach allows for the construction of simpler classifiers which can then be used to reject most negative (non face) input quickly while spending more time on positive (face) input. It is another sort of "hack" to boost the speed and accuracy of our model. So, we start by taking a sub window and within this sub window, we take our most important or best feature and see if it is present in the image within the sub window. If it is not in the sub window, then we don't even look at the sub window, we just discard it. Then if it is present, we look at the second feature in the sub window. If it isn't present, then we reject the sub window. We go on for the number of features have, and reject the sub windows without the feature.

Evaluations may take split seconds but since you have to do it for each feature, it could take a lot of time. Cascading speeds up this process a lot, and the machine is able to deliver results much faster.

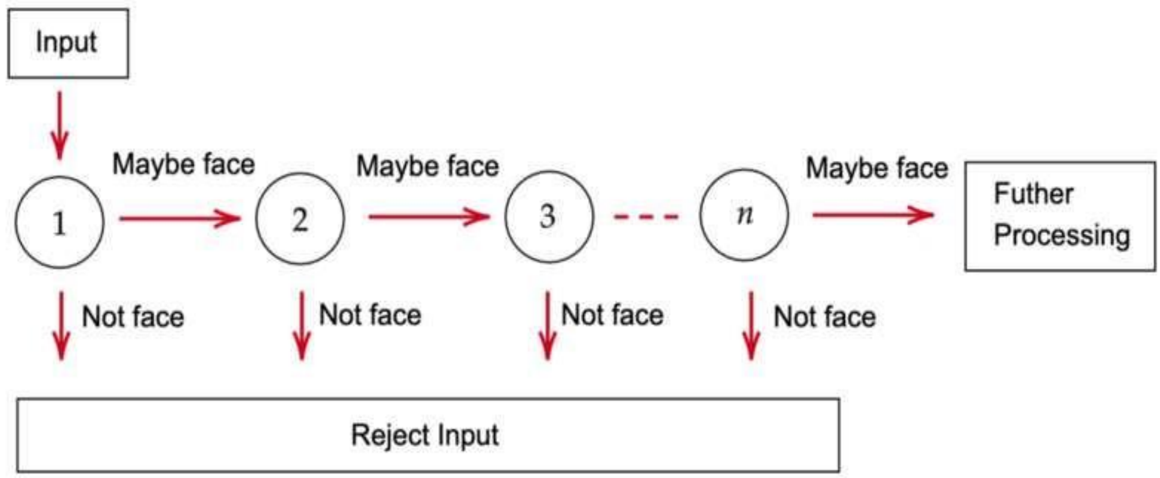


Fig 15: Cascade Classifier

3.2 FACE RECOGNITION:

3.2.1 Convolution Neural Networks

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 networks are 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.

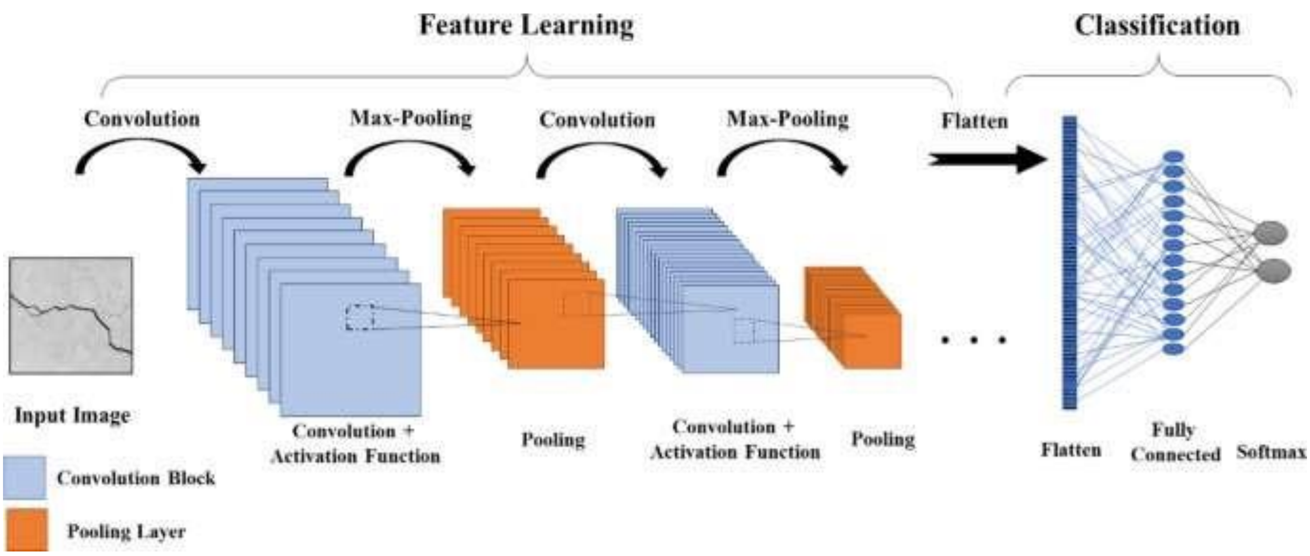


Fig 16: Basic CNN

3.2.2 Architecture of CNN

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 networks are based on computational models. Neural networks are based either on the study of the brain or on the application of neural networks to artificial intelligence.

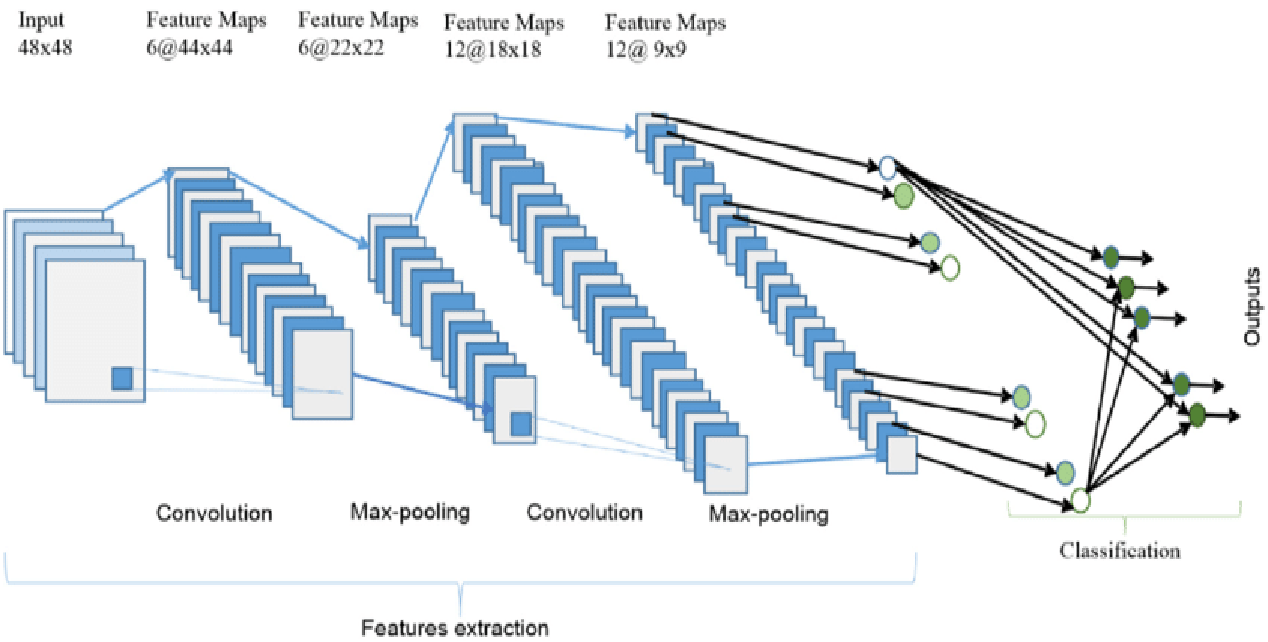


Fig 17: CNN Architecture

1) Face detection using CNN

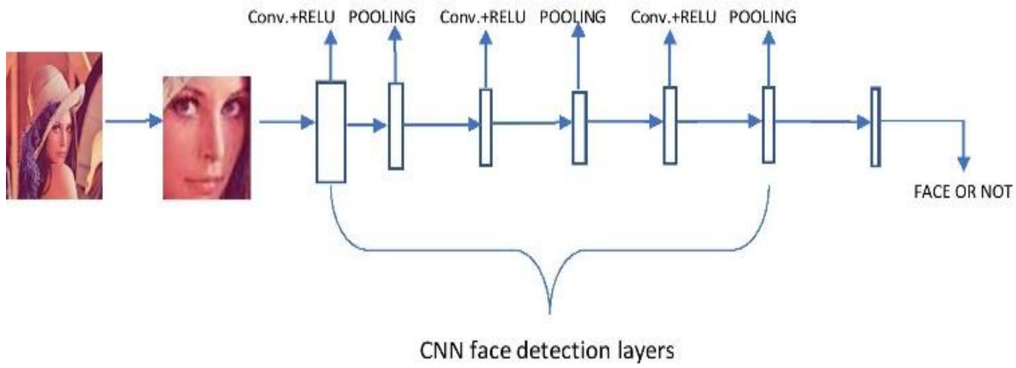


Fig 18: Face Detection using CNN

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

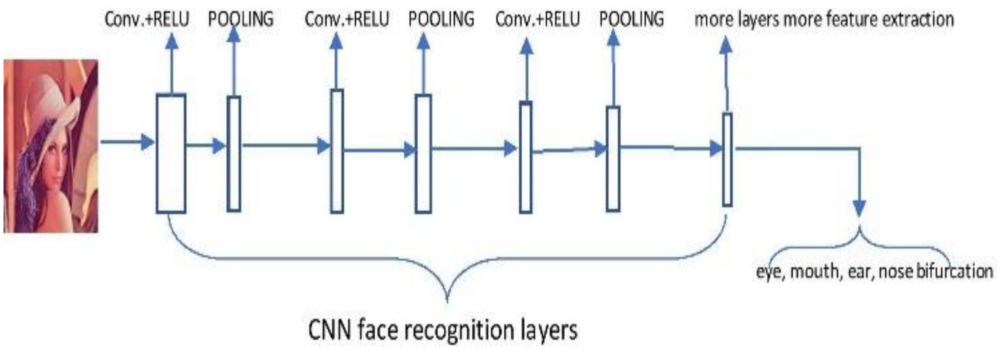


Fig 19: Face Recognition using CNN

The face recognition algorithm is used to find the characteristics which accurately represents a picture with the features which are already extracted, scaled, and converted into grey scale.

A convolutional neural network consists of an input layer, **hidden layers** and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are masked by the activation function and final **convolution**. In a convolutional neural network, the hidden layers include layers that perform convolutions.

Typically, this includes a layer that performs a **dot product** of the convolution kernel with the layer's input matrix. This product is usually the **Frobenius inner product**, and its activation function is commonly **RELU**. As the convolution kernel slides along the input matrix for the layer, the convolution operation generates a feature map, which in turn contributes to the input of the next layer. This is followed by other layers such as pooling layers, fully connected layers, and normalization layers.

3.2.3 Defining the CNN Model

- Input layer
- Convo layer (Convo + ReLU)
- Pooling layer
- Fully connected(FC) layer
- Softmax/Logistic layer
- Output layer

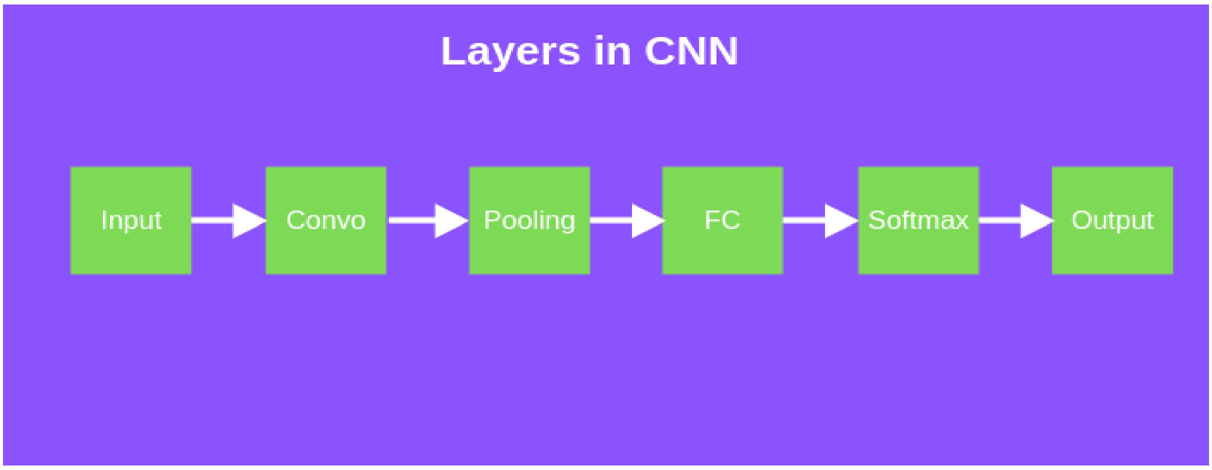


Fig 20: CNN Layers

1. Input Layer: The input layer in CNN should contain image data. Image data is represented by a three-dimensional 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 = 784, m.

2. 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 the Convo layer to perform convolution operation as we saw earlier and calculate the dot product between the filter and the receptive. 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.

3. Pooling Layer:

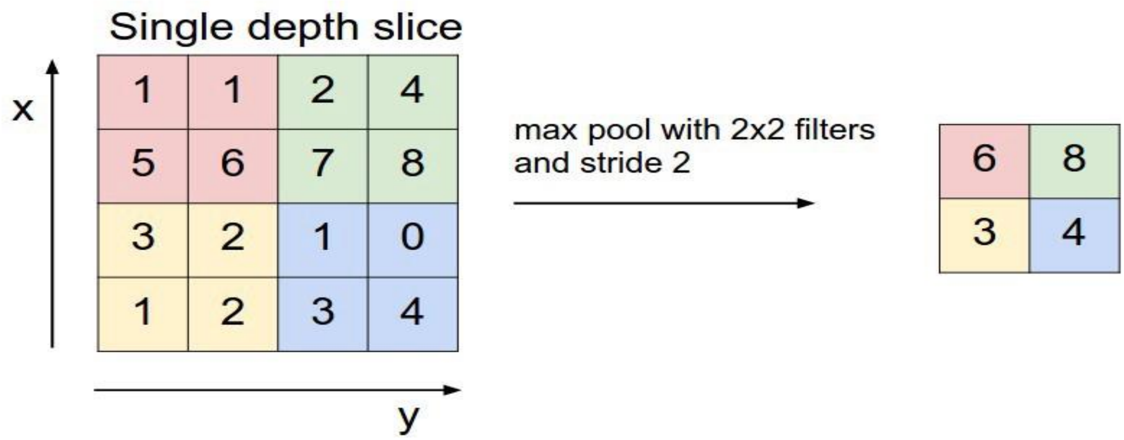


Fig 21: Pooling Layer

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.

Flatten Layer:
It is used to flatten all the layers into a single 1D layer.

Dropout Layer:
It is used to prevent the model from overfitting.

4. Fully Connected Layer:

A fully connected layer includes the biases, neurons, and weights. It interlinks the neurons in one layer to neurons in another layer. It is used to segregate images between various types by training.

5. Softmax or Dense or Logistic Layer:

Softmax or Dense layer is the final layer of CNN. The activation function here is softmax which will output a vector with two probability distribution values. It resides at the penultimate layer of the Output layer. Softmax is for multi-classification, and Logistic is used for binary classification.

6. Output Layer:

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

Filters in Convolutional Neural Networks:

Within a convolutional layer, the input is transformed before being passed to the next layer. A CNN transforms the data by using filters. 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 22: 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.

3.2.4 CNN for Age and Gender Estimation

CNN can classify the age and gender of unfiltered face images relying on its good feature extraction technique. CNN model can learn compact and discriminative facial features, especially when the volume of training images is sufficiently large, to obtain the relevant information needed for the two classifications. Data-sets for age and gender estimation from real-world social images are therefore relatively limited in size and presently no match in size with the much larger image classification data-sets. Overfitting is common problem when machine learning based methods are used on such small image collections. This problem is exacerbated when considering deep convolutional neural networks due to their huge numbers of model parameters. Care must therefore be taken in order to avoid overfitting under such circumstances.

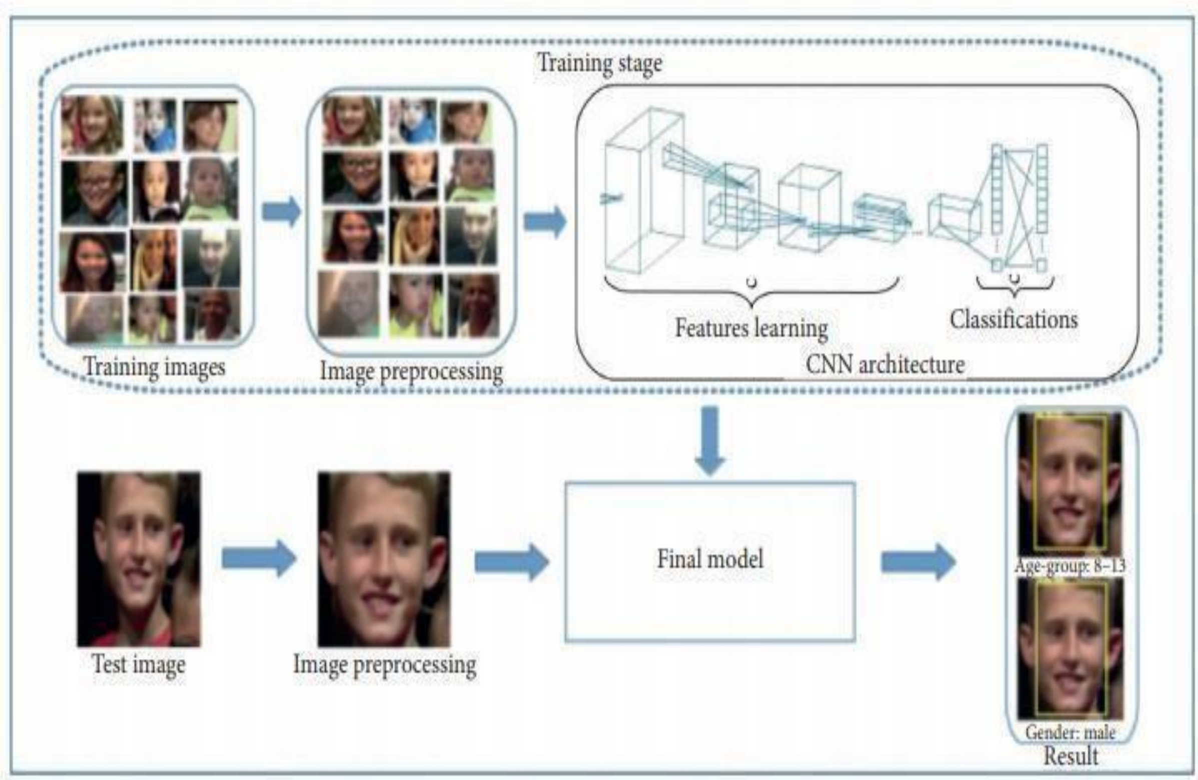


Fig 23: CNN for Age and Gender

4. DESIGN

4.1 System Architecture

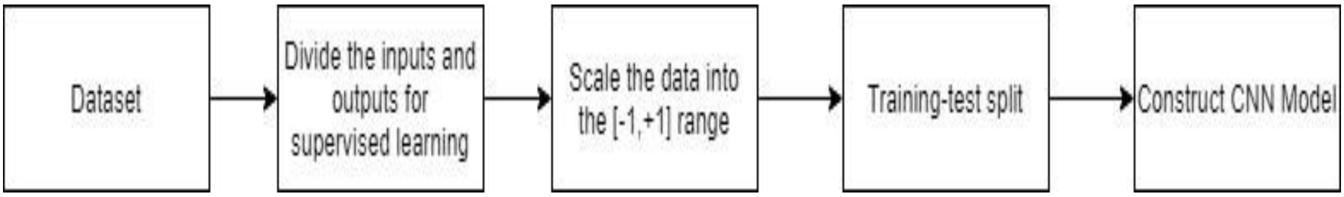


Fig 24: Pre-processing of data

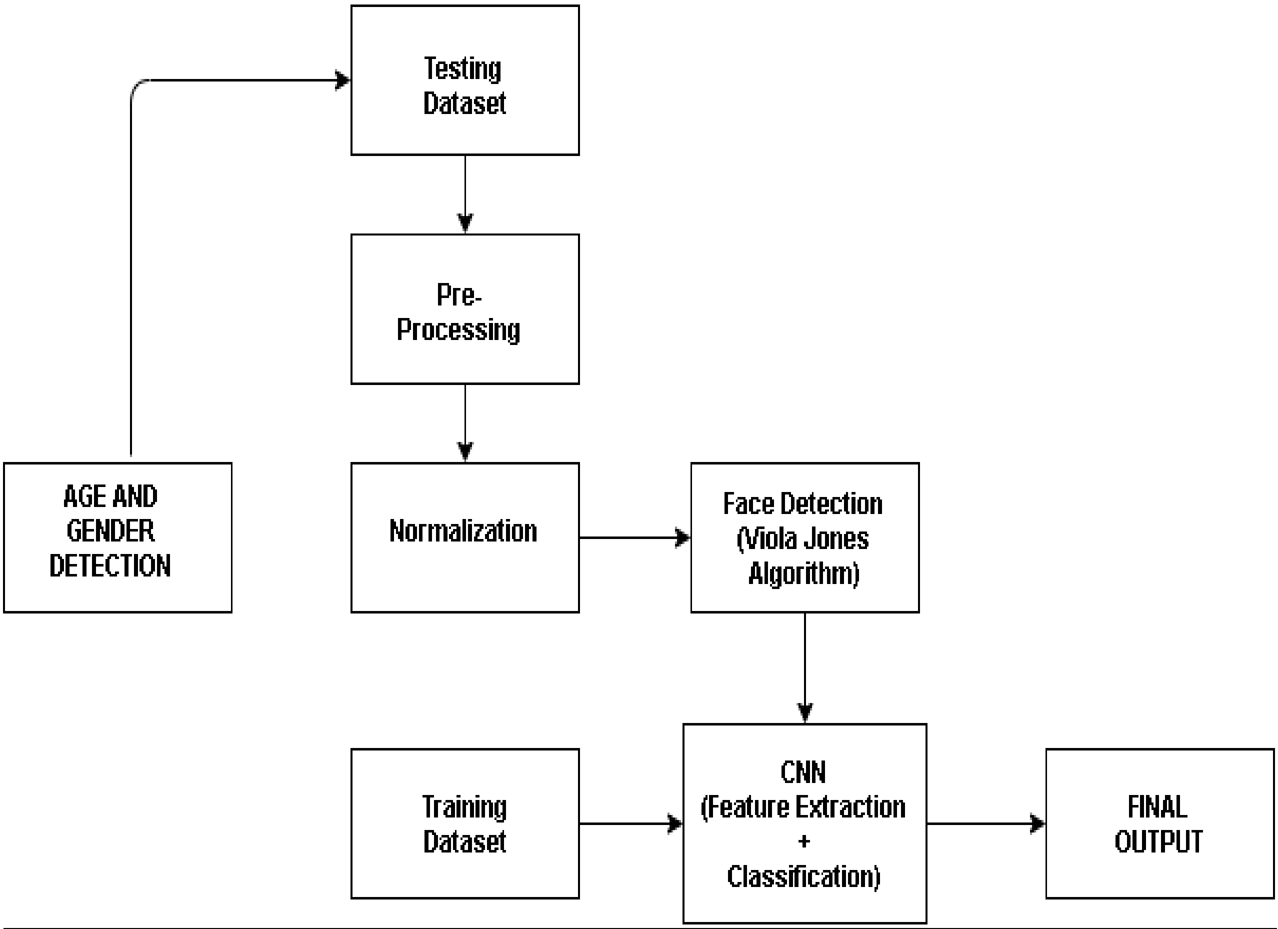


Fig 25: System Architecture

4.2 Network Architecture

Our proposed network architecture is used throughout our experiments for both age and gender classification. We are attempting to solve: age classification on the UTKFace data set which requires distinguishing between eight classes; gender only two. This, compared to, e.g., the ten thousand identity classes used to train the network used for face recognition Our network architecture comprises four convolutional layers and two fully connected layers comprising a finite amount of neurons. The main motivation of our project is to design a small architecture that can increase performance and reduce the risk of overfitting.

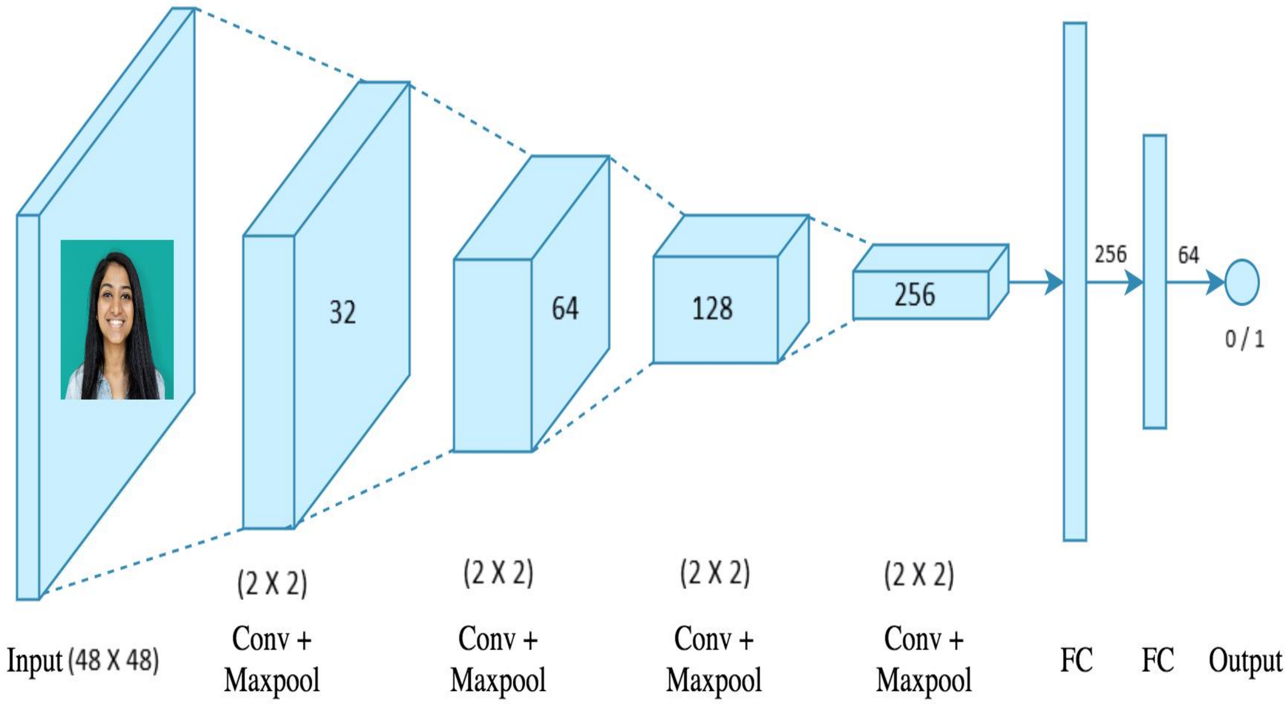


Fig 26: Network Architecture

The input image is first rescaled to a size of 48 X 48 and then fed to the convolutional neural network model where it then detects the age and gender of a person from the set of images that we have given for training.

1. 32 filters of size 24 X 24 pixels are applied to the input in the first convolutional layer, followed by a rectified linear operator (ReLU), a max-pooling layer that takes an input of 2 × 2 regions with a one-pixel stride.

2. The output of the first layer is then fed to the second convolutional layer, containing 64 filters of size 12 X 12 pixels which are again followed by ReLU and a max-pooling layer.

3. The third convolutional layer operates by applying a set of 128 filters of size 6 X 6 pixels, followed by ReLU and a max-pooling layer.

4. Finally, the fourth convolutional layer i.e the last layer contains 256 filters of size 3 X 3 pixels followed by ReLU and max-pooling similar to the first three layers as discussed above.

5. The first fully connected layer receives the output of the fourth convolutional layer which contains 256neurons, followed by a ReLU and a dropout layer.

6. The second fully connected layer that receives output from the first fully connected layer contains 64 neurons, followed by a ReLU and a dropout layer.

Finally, the output of the second fully connected layer is fed to a soft-max which is a probability distribution function calculated for each class that predicts the age and gender by taking the class with the maximum probability of either 1 or 0 for the given test image where 0 indicates Female and 1 indicates the gender Male.

4.3 Structure Chart

A structure chart (SC) in software engineering and organizational theory is a chart which shows the breakdown of a system to its lowest manageable levels. They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name.

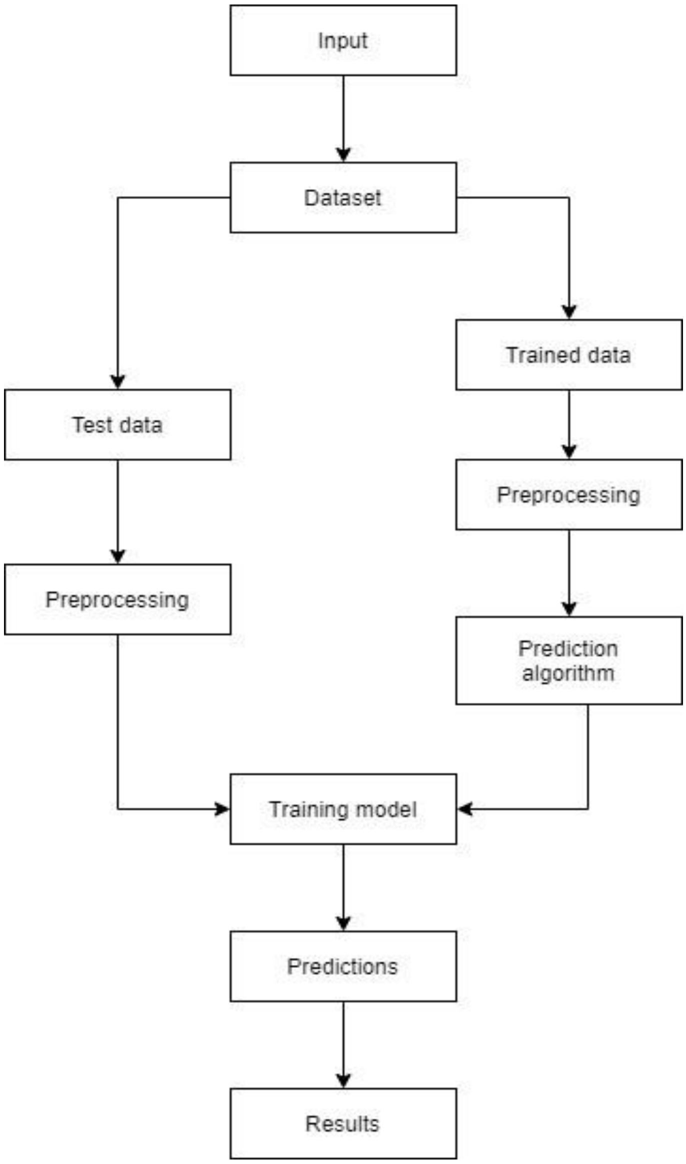


Fig 27: Structure Chart

4.4 UML Diagrams

A UML diagram is a partial graphical representation (view) of a model of a system under design, implementation, or already in existence. UML diagram contains graphical elements (symbols) - UML nodes connected with edges (also known as paths or flows) - that represent elements in the UML model of the designed system. The UML model of the system might also contain other documentation such as use cases written as template texts. The kind of the diagram is defined by the primary graphical symbols shown on the diagram.

For example, a diagram where the primary symbols in the contents area are classes is class diagram. A diagram which shows use cases and actors is use case diagram. A sequence diagram shows sequence of message exchanges between lifelines.

UML specification does not preclude mixing of different kinds of diagrams, e.g. to combine structural and behavioral elements to show a state machine nested inside a use case. Consequently, the boundaries between the various kinds of diagrams are not strictly enforced. At the same time, some UML Tools do restrict set of available graphical elements which could be used when working on specific type of diagram.

UML specification defines two major kinds of UML diagram: structure diagrams and behavior diagrams.

Structure diagrams show the static structure of the system and its parts on different abstraction and implementation levels and how they are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts.

Behavior diagrams show the dynamic behavior of the objects in a system, which can be described as a series of changes to the system over time.

4.4.1 Dataflow Diagrams

DFD is the abbreviation for **Data Flow Diagram**. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. A data flow diagram (DFD) maps out the flow of information for any process or system. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. A DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences.

- **Training phase:**

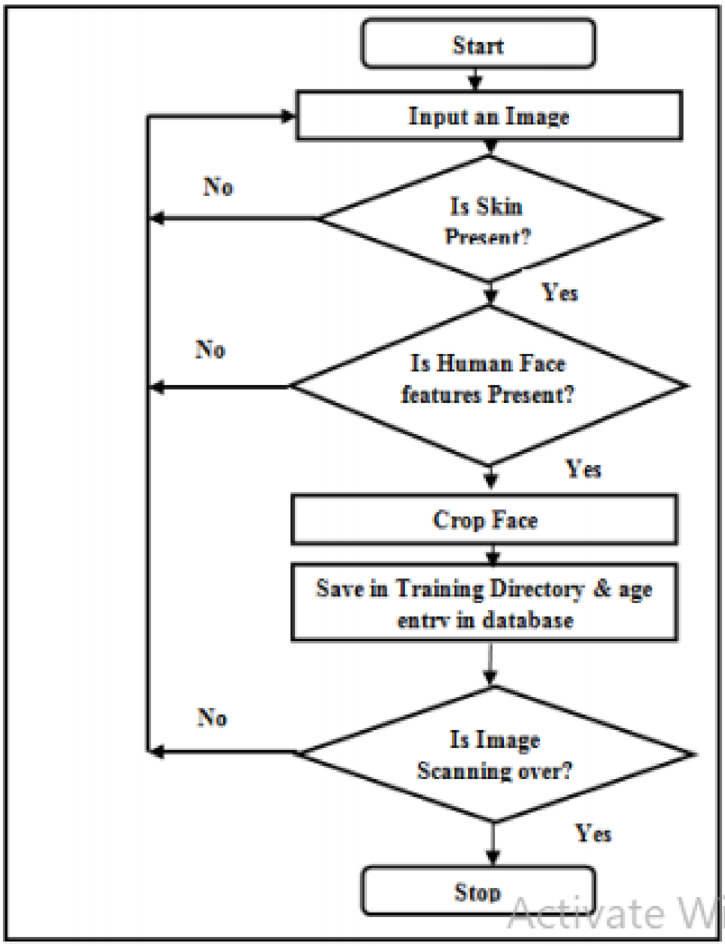


Fig 28: DFD for Training phase

- **Testing Phase:**

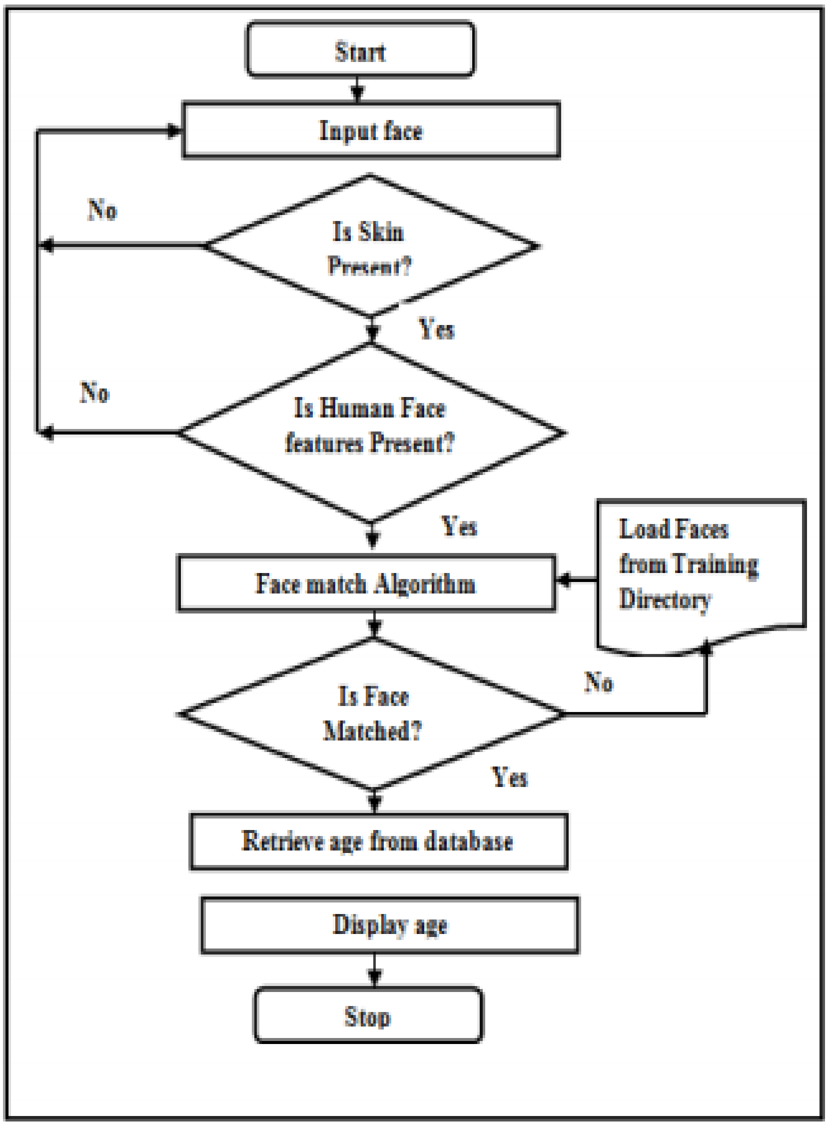


Fig 29: DFD for Testing phase

5. DATASET DETAILS

In this project, we have used the UTKFace dataset which intends to facilitate the study of age and gender recognition. The data included in this collection is intended to be as true as possible to the challenges of real-world imaging conditions. This dataset serves as a benchmark for face photos and is inclusive of various real-world imaging conditions like noise, lighting, pose, appearance and more, that can be expected of images taken without careful preparation or posing.

UTKFace dataset is a large-scale face dataset with long age span (range from 0 to 116 years old). The dataset consists of over 20,000 face images with annotations of age, gender, and ethnicity. The images cover large variation in pose, facial expression, illumination, occlusion, resolution, etc. This dataset could be used on a variety of tasks, e.g., face detection, age estimation, age progression/regression, landmark localization, etc. The model we will use have been trained on this dataset.

Fig 30: The UTKFace samples

- consists of 20k+ face images in the wild (only single face in one image)
- provides the correspondingly aligned and cropped faces
- provides the corresponding landmarks (68 points)
- images are labelled by age, gender, and ethnicity

6. EXPERIMENTAL ANALYSIS & RESULTS

6.1 SYSTEM CONFIGURATION

This project can run on commodity hardware. We ran this entire project on an Intel I5 processor with 8 GB Ram, 2 GB NVidia Graphic Processor, it also has 2 cores which runs at 1.7 GHz, 2.1 GHz respectively.

First part of the is training phase which takes 20-30mins of time and the second part is testing part which only takes few seconds to make prediction and calculate the accuracy and finally using the prediction playlist is generated.

6.1.1 Software Requirements

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. The software requirements that are required for this project are:

• Python 3.5

• Libraries used are:

1. Open CV
2. NumPy
3. Keras
4. TensorFlow
5. MatPlotlib

• Operating System: Windows 7 and above or Linux based OS or MAC OS.

6.1.2 Hardware Requirements:

• RAM: 4 GB

• Storage: 500 GB

• Webcam

• CPU: 2 GHz or faster

• Architecture: 32-bit or 64-bit

6.2 DEFINING LIBRARIES

TensorFlow:

Tensorflow is used as a backend in the application of this project. TensorFlow is a brilliant records circulation in the Machine Learning Library made through the Google Brain Team and made supply in 2014. It is designed for ease of use and remarkably relevant to each numeric and neural gadget trouble, relatively like the variety of spaces. TensorFlow is a low-level math-entangled tool that tracks experts who capture what they're doing to construct experimental studying design, play around with them, and turn them into running executive programs. For the most part, it is considered as a programming part in which equations can be authorized as graphs. Math values are implemented using nodes in the chart, and the edges include the multidimensional facts clusters that are termed as tensors connected to nodes. It is an open source framework to run deep learning and other statistical and predictive analytics workloads. It is a python library that supports many classification and regression algorithms and more generally deep learning.

OpenCV:

OpenCV (*Open-source computer vision*) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by willow garage then Itseez (which was later acquired by Intel). The library is cross platform and free for use under the open-source BSD license. OpenCV supports some models from deep learning frameworks like TensorFlow, Torch, PyTorch (after converting to an ONNX model) and Caffe according to a defined list of supported layers. It promotes Open Vision Capsules. which is a portable format, compatible with all other formats. It is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including feature like face detection and object detection.

Python:

Python is used for the period of the assignment's implementation; several traces of code had been brought to complete the assignment requirements. Python is a high-level programming language and an interpreted language. Furthermore, Python is a programming language that gives the chances to work within a short span and more precisely organized structures.

Anaconda:

The Anaconda is used as an Integrated Development Environment all through the implementation of this project. Anaconda is an open-source distribution and free of the Python and R programming languages for scientific and analytics computing to simplify package management and deployment. It's various applications such as statistics science, A.I. applications, large-scale details, visionary investigation, etc. Anaconda goes along with over 1,400 programs like the Conda package and digital frames.

Keras:

Keras is used to build a model to order the layers in the implementation of this project. Keras is an open-source software library that is written in Python that can run on top of TensorFlow, CNTK, or Theano predominantly works in the field of A.I. It used to be developed with an essential point on allowing for quick experimentation. For example, the key to doing an accurate lookup can go from notion to result with the minor delay viable. Thus, Keras approves for handy and speedy prototyping (personal friendliness, modularity, and extensibility). Similarly, it supports each convolutional network and recurrent networks as correctly as a mixture of the two and runs smoothly on Central Processing Unit and graphical processing unit.

NumPy:

NumPy is used for mathematical computations to display out the predicted rows in the project. NumPy is the core bundle with Python adding support for arrays, matrices, multidimensional arrays, and computing. It is a flexible, enlightened (broadcasting) with N-dimensional array object created for a specific software program for implementing the C or C++ and Fortran code, linear algebra, Fourier transforms, and random number capacity.

Matplotlib:

Matplot is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits. Matplotlib is an extraordinary tool to plot model accuracy and loss in a graphical plot for this project. It is a library in Python 2D plotting that gives outputs as an exact figure for publication across different platforms in various hardcopy formats and responsive environments. It can be used in Python scripts, Jupyter notebook, Web software servers, and graphical users.

6.3 TECHNOLOGIES DESCRIPTION

Image Processing:

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too. Image processing basically includes the following three steps:

- Importing the image via image acquisition tools.
- Analyzing and manipulating the image.
- Output in which result can be altered image or report that is based on image analysis.

Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction.

Computer Vision:

Computer vision is a field of artificial intelligence that trains computers to interpret and understand the visual world. Using digital images from cameras and videos and deep learning models, machines can accurately identify and classify objects — and then react to what they "see." Computer vision works much the same as human vision, except humans have a head start. Human sight has the advantage of lifetimes of context to train how to tell objects apart, how far away they are, whether they are moving and whether there is something wrong in an image.

Computer vision needs lots of data. It runs analyses of data over and over until it discerns distinctions and ultimately recognize images. For example, to train a computer to recognize automobile tires, it needs to be fed vast quantities of tire images and tire-related items to learn the differences and recognize a tire, especially one with no defects. Two essential technologies are used to accomplish this: a type of machine learning called deep learning and a convolutional neural network (CNN). Machine learning uses algorithmic models that enable a computer to teach itself about the context of visual data. A CNN helps a machine learning or deep learning model "look" by breaking images down into pixels that are given tags or labels.

6.4 SAMPLE CODE

Data Preprocessing:

```
import cv2
ages=[]
genders=[]
images=[]for fle in files:
  age=int(fle.split('_')[0])
  gender=int(fle.split('_')[1])
  total=fldr+'/'+fle
  print(total)
  image=cv2.imread(total)  image = cv2.cvtColor(image,
cv2.COLOR_BGR2RGB)
  image= cv2.resize(image,(48,48))
  images.append(image)
  ages.append(age)
  genders.append(gender)
```

```
labels=[]i=0
while i<len(ages):
  label=[]
  label.append([ages[i]])
  label.append([genders[i]])
  labels.append(label)
  i+=1
```

```
images_f=np.array(images)
labels_f=np.array(labels)
images_f_2=images_f/255
X_train, X_test, Y_train, Y_test= train_test_split(images_f_2,
labels_f,test_size=0.25)
```

Training:

```
Y_train_2=[Y_train[:,1],Y_train[:,0]]
Y_test_2=[Y_test[:,1],Y_test[:,0]]
```

Building Model:

```
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten,BatchNormalization
from tensorflow.keras.layers import Dense, MaxPooling2D,Conv2D
from tensorflow.keras.layers import Input,Activation,Add
from tensorflow.keras.models import Model
from tensorflow.keras.regularizers import l2
from tensorflow.keras.optimizers import Adam
import tensorflow as tfdef Convolution(input_tensor,filters):

    x = Conv2D(filters=filters,kernel_size=(3, 3),padding =
'same',strides=(1, 1),kernel_regularizer=l2(0.001))(input_tensor)
    x = Dropout(0.1)(x)
    x= Activation('relu')(x)    return x
def model(input_shape):
  inputs = Input((input_shape))

  conv_1= Convolution(inputs,32)
  maxp_1 = MaxPooling2D(pool_size = (2,2)) (conv_1)
  conv_2 = Convolution(maxp_1,64)
  maxp_2 = MaxPooling2D(pool_size = (2, 2)) (conv_2)
  conv_3 = Convolution(maxp_2,128)
  maxp_3 = MaxPooling2D(pool_size = (2, 2)) (conv_3)
  conv_4 = Convolution(maxp_3,256)
  maxp_4 = MaxPooling2D(pool_size = (2, 2)) (conv_4)
  flatten= Flatten() (maxp_4)
  dense_1= Dense(64,activation='relu')(flatten)
  dense_2= Dense(64,activation='relu')(flatten)
  drop_1=Dropout(0.2)(dense_1)
  drop_2=Dropout(0.2)(dense_2)
  output_1= Dense(1,activation="sigmoid",name='sex_out')(drop_1)
  output_2= Dense(1,activation="relu",name='age_out')(drop_2)
  model = Model(inputs=[inputs], outputs=[output_1,output_2])
  model.compile(loss=["binary_crossentropy","mae"], optimizer="Adam",
      metrics=["accuracy"])

  return model
```

Training Model:

```
from tensorflow.keras.callbacks import ModelCheckpoint import
tensorflow as tf
fle_s='Age_sex_detection.h5'
checkpointer = ModelCheckpoint(fle_s,
monitor='val_loss',verbose=1,save_best_only=True,save_weights_only=Fal
se, mode='auto',save_freq='epoch')
Early_stop=tf.keras.callbacks.EarlyStopping(patience=75,
monitor='val_loss',restore_best_weights=True),
callback_list=[checkpointer,Early_stop]
History=Model.fit(X_train,Y_train_2,batch_size=64,validation_data=(X_t
est,Y_test_2),epochs=500,callbacks=[callback_list])
```

Evaluating Model:

```
def test_image(ind,images_f,images_f_2,Model):
cv2_imshow(images_f[ind])
image_test=images_f_2[ind]
pred_1=Model.predict(np.array([image_test]))
#print(pred_1)
sex_f=['Male','Female']
age=int(np.round(pred_1[1][0]))
sex=int(np.round(pred_1[0][0]))
print("Predicted Age: "+ str(age))
print("Predicted Sex: "+ sex_f[sex])
```

1.DETECTING FACE THROUGH WEBCAM:

Training code:

```
import numpy as np
from sklearn.model_selection import train_test_split
from tensorflow.keras.layers import Dropout,Dense,Flatten,MaxPooling2D,Conv2D,Input,
Activation
from tensorflow.keras.models import Model
from tensorflow.keras.regularizers import l2
from tensorflow.keras.callbacks import ModelCheckpoint
import tensorflow as tf
from sklearn.metrics import classification_report
import warnings
warnings.filterwarnings("ignore")

epochs=500
totalimages=10000


fldr="./UTKFace"
files=os.listdir(fldr)
ages=[]
genders=[]
images=[]
n=totalimages
for fle in files:
  n-=1
  if(n==0):
    break
  age=int(fle.split('_')[0])
  gender=int(fle.split('_')[1])
  total=fldr+'/'+fle
  print(total)
  image=cv2.imread(total)

  image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
  image= cv2.resize(image,(48,48))
  images.append(image)


n = totalimages
for fle in files:
  n -= 1
  if (n == 0):
    break
```

```
  age=int(fle.split('_')[0])
  gender=int(fle.split('_')[1])
  ages.append(age)
  genders.append(gender)


images_f=np.array(images)

labels=[]

i=0
while i<len(ages):
  label=[]
  label.append([ages[i]])
  label.append([genders[i]])
  labels.append(label)
  i+=1


images_f_2=images_f/255
labels_f=np.array(labels)
X_train, X_test, Y_train, Y_test= train_test_split(images_f_2, labels_f,test_size=0.
25)
Y_train_2=[Y_train[:,1],Y_train[:,0]]
Y_test_2=[Y_test[:,1],Y_test[:,0]]




def Convolution(input_tensor, filters):
  x = Conv2D(filters=filters, kernel_size=(3, 3), padding='same', strides=(1, 1), ke
rnel_regularizer=l2(0.001))(
    input_tensor)
  x = Dropout(0.1)(x)
  x = Activation('relu')(x)

  return x


def model(input_shape):
  inputs = Input((input_shape))

  conv_1 = Convolution(inputs, 32)
  maxp_1 = MaxPooling2D(pool_size=(2, 2))(conv_1)

  conv_2 = Convolution(maxp_1, 64)
  maxp_2 = MaxPooling2D(pool_size=(2, 2))(conv_2)
```

```
  conv_3 = Convolution(maxp_2, 128)
  maxp_3 = MaxPooling2D(pool_size=(2, 2))(conv_3)




  conv_4 = Convolution(maxp_3, 256)
  maxp_4 = MaxPooling2D(pool_size=(2, 2))(conv_4)

  flatten = Flatten()(maxp_4)

  dense_1 = Dense(64, activation='relu')(flatten)
  dense_2 = Dense(64, activation='relu')(flatten)

  drop_1 = Dropout(0.2)(dense_1)
  drop_2 = Dropout(0.2)(dense_2)


  output_1 = Dense(1, activation="sigmoid", name='sex_out')(drop_1)
  output_2 = Dense(1, activation="relu", name='age_out')(drop_2)

  model = Model(inputs=[inputs], outputs=[output_1, output_2])
  model.compile(loss=["binary_crossentropy", "mae"], optimizer="Adam",
                metrics=["accuracy"])

  return model


Model=model((48,48,3))


History=Model.fit(X_train,Y_train_2,batch_size=64,validation_data=(X_test,Y_test_2),
epochs=epochs)

Model.evaluate(X_test,Y_test_2)
Model.save("Model.h5")

pred=Model.predict(X_test)
i=0
Pred_l=[]

while(i<len(pred[0])):

  Pred_l.append(int(np.round(pred[0][i])))
  i+=1

report=classification_report(Y_test_2[0], Pred_l)
print(report)
```

Testing code:

```
import tensorflow as tf
import cv2
import numpy as np
import os
import warnings
warnings.filterwarnings("ignore")

Model = tf.keras.models.load_model('./Model.h5')


def projimg(image,Model):
  image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
  image = cv2.resize(image, (48, 48))

  pred_1=Model.predict(np.array([image]))

  sex_f = ['Female', 'Male']
  age=int(np.round(pred_1[1][0]))
  sex=int(np.round(pred_1[0][0]))

  return "Gender: "+sex_f[sex]+", Age: "+str(age)[0:2]


def draw_border(img, pt1, pt2, color, thickness, r, d):

    x1,y1 = pt1
    x2,y2 = pt2

    # Top left
    cv2.line(img, (x1 + r, y1), (x1 + r + d, y1), color, thickness)
    cv2.line(img, (x1, y1 + r), (x1, y1 + r + d), color, thickness)
    cv2.ellipse(img, (x1 + r, y1 + r), (r, r), 180, 0, 90, color, thickness)
    # Top right
    cv2.line(img, (x2 - r, y1), (x2 - r - d, y1), color, thickness)
    cv2.line(img, (x2, y1 + r), (x2, y1 + r + d), color, thickness)
    cv2.ellipse(img, (x2 - r, y1 + r), (r, r), 270, 0, 90, color, thickness)
    # Bottom left
    cv2.line(img, (x1 + r, y2), (x1 + r + d, y2), color, thickness)
    cv2.line(img, (x1, y2 - r), (x1, y2 - r - d), color, thickness)
    cv2.ellipse(img, (x1 + r, y2 - r), (r, r), 90, 0, 90, color, thickness)
    # Bottom right
    cv2.line(img, (x2 - r, y2), (x2 - r - d, y2), color, thickness)
    cv2.line(img, (x2, y2 - r), (x2, y2 - r - d), color, thickness)
    cv2.ellipse(img, (x2 - r, y2 - r), (r, r), 0, 0, 90, color, thickness)
```

```
cap = cv2.VideoCapture(0)

while(True):
    ret, frame = cap.read()
    height,width = frame.shape[:2]
    label=projimg(frame,Model)

    cascade = cv2.CascadeClassifier("./haarcascade_frontalface_default.xml")
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    rects = cascade.detectMultiScale(gray, 1.2, 3,minSize=(50, 50))

    if len(rects) > 0:
        # Draw a rectangle around the faces
        for (x, y, w, h) in rects:
            draw_border(frame, (x, y), (x + w, y + h), (255, 0, 105),4, 15, 10)

        cv2.putText(frame, str(label), (100,height-
20), cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (255,255,255), 1, cv2.LINE_AA)

    else:
        cv2.putText(frame, str("No Face Detected"), (100,height-
20), cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (255,255,255), 1, cv2.LINE_AA)


    cv2.imshow('frame', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

2. PASSING IMAGE AS INPUT:

```
import tensorflow as tf
import cv2
import numpy as np
import os

Model = tf.keras.models.load_model('./Model.h5')
files=os.listdir('./Testimages/')

images=[]
filepaths=[]
for fle in files:
  total='./Testimages'+'/'+fle
```

```
  filepaths.append(total)

image=cv2.imread(total)

  image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
  image= cv2.resize(image,(48,48))
  images.append(image)

def test_image(filename,images_f,images_f_2,Model):
  ind=-1
  for path in filepaths:
    ind+=1
    if(path[13:]==filename):
      break

  originalImage=cv2.imread(filepaths[ind])
  cv2.imshow("",originalImage)
  cv2.waitKey(0)
  image_test=images_f_2[ind]
  pred_1=Model.predict(np.array([image_test]))

  print(pred_1)

  sex_f=['Male','Female']
  age=int(np.round(pred_1[1][0]))
  sex=int(np.round(pred_1[0][0]))

  print("Predicted Age: "+ str(age))
  print("Predicted Sex: "+ sex_f[sex])

def projimg(image,Model):
  image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
  image = cv2.resize(image, (48, 48))

  pred_1=Model.predict(np.array([image]))

  sex_f = ['Female', 'Male']
  age=int(np.round(pred_1[1][0]))
  sex=int(np.round(pred_1[0][0]))

  print("Predicted Age: "+ str(age))
  print("Predicted Sex: "+ sex_f[sex])

  return "Gender: "+sex_f[sex]+", Age: "+str(age)[0:2]

a=input("Enter Filename: ")
images_f = np.array(images)
images_f_2 = images_f / 255
```

6.5 SAMPLE INPUTS AND OUTPUTS

Output 1: Male

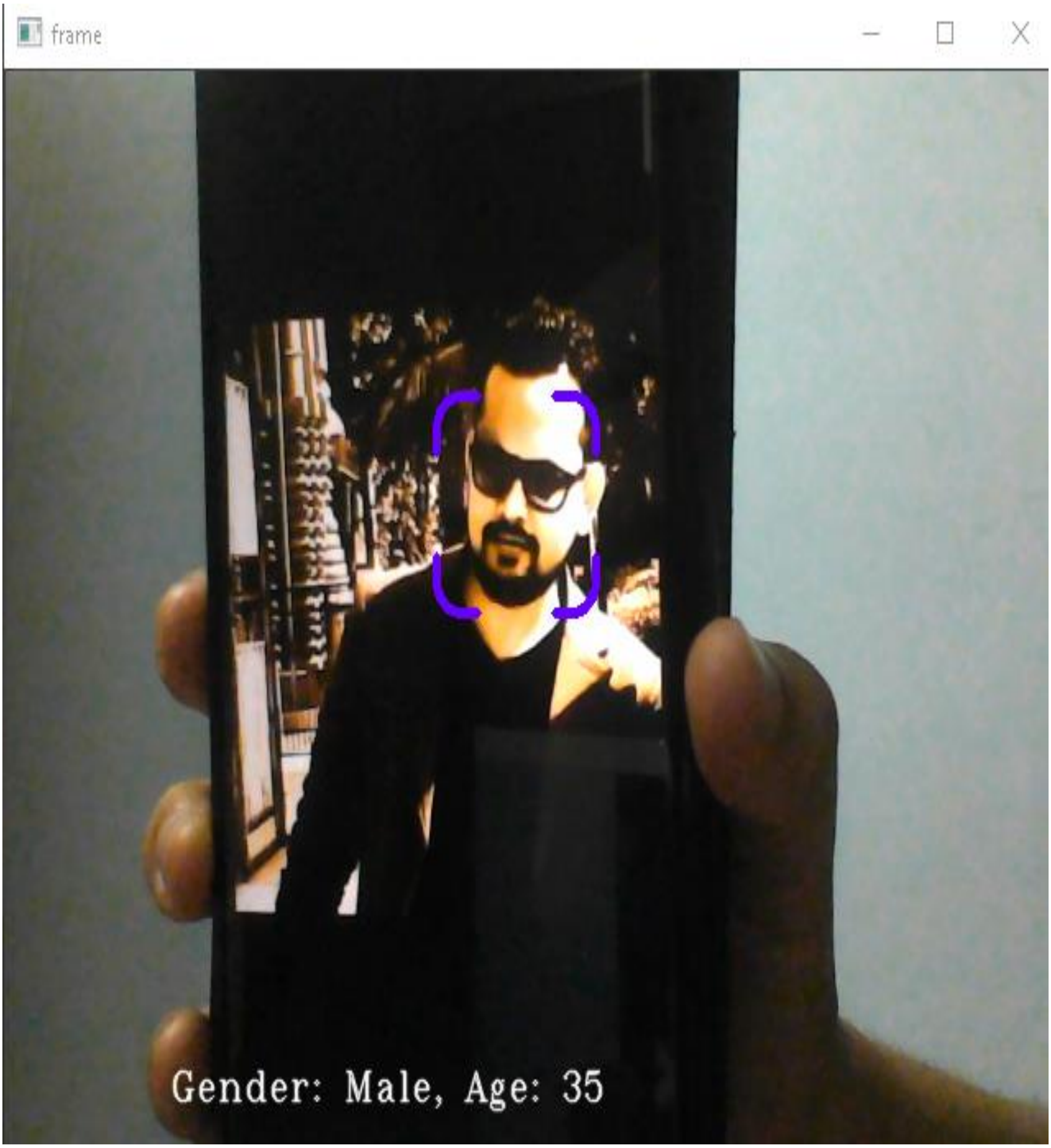


Fig 31: Output 1

Output 2: Female

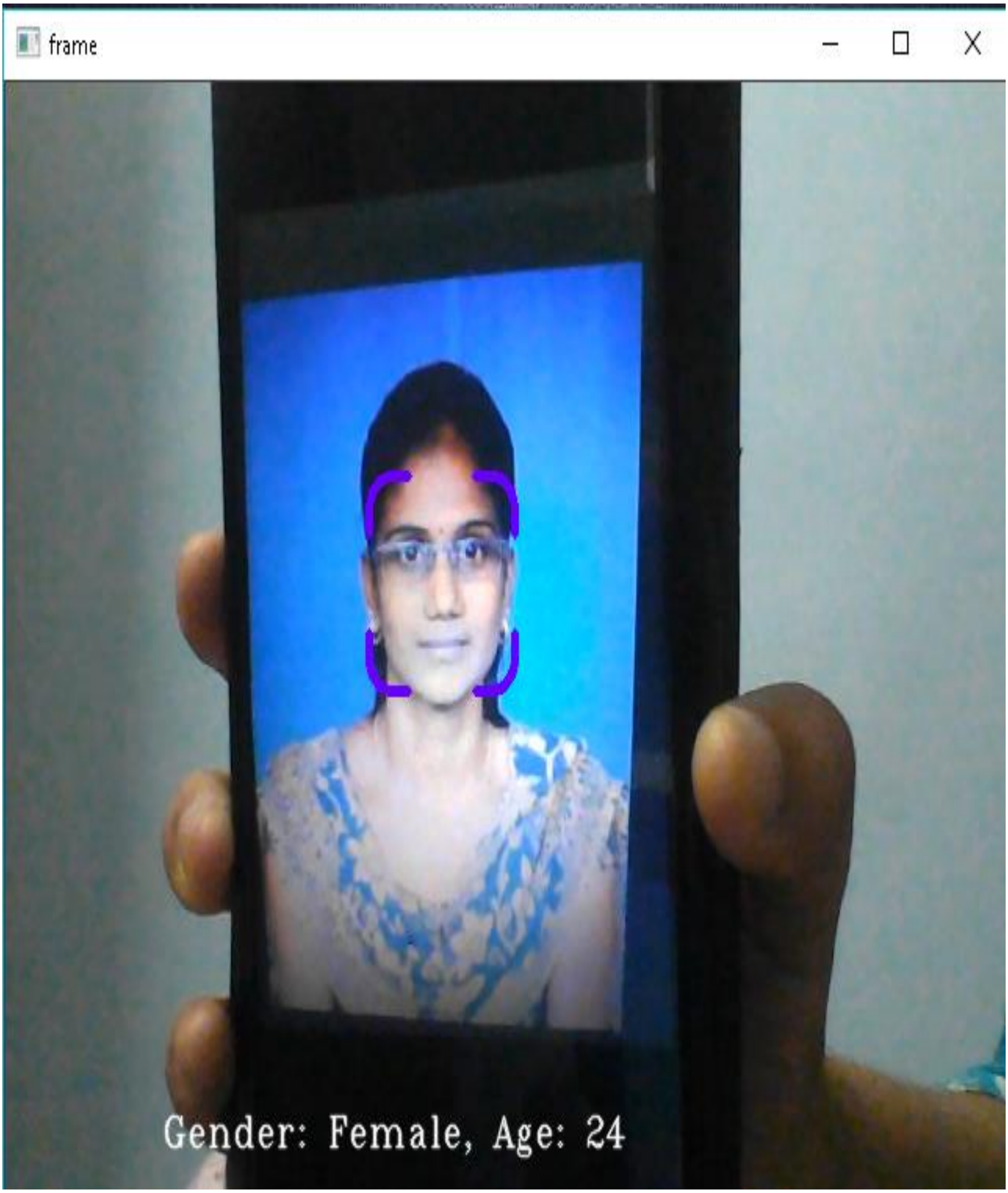


Fig 32: Output 2

Output 3: Male

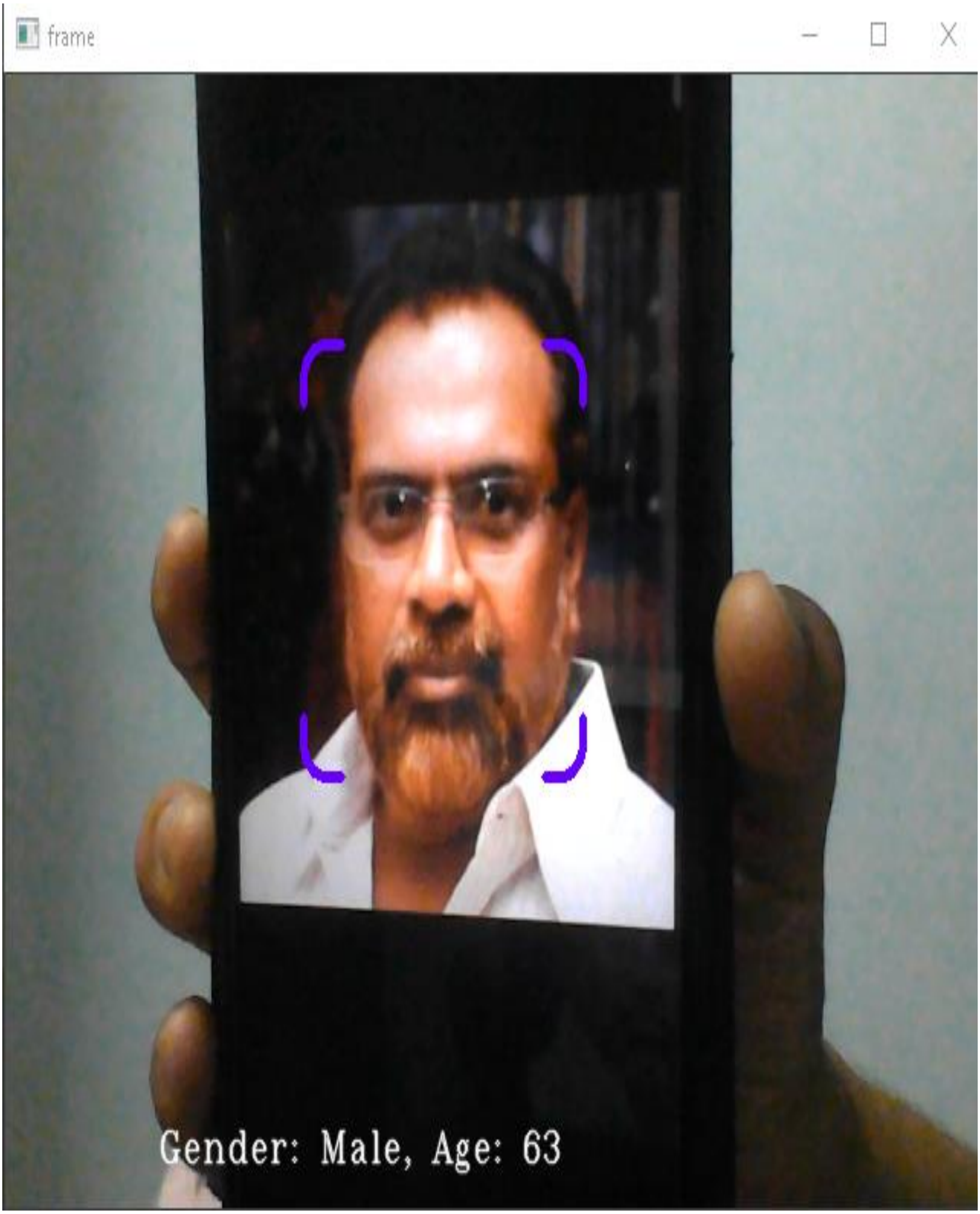


Fig 33: Output 3

Output 4: No Face Detected

Fig 34: Output 4

6.6 PERFORMANCE MEASURE

For Performance measure we can use the following to find whether our model is accurate or not.

1.Accuracy: Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

Accuracy= Correct predictions/ Total Predictions

2.Precision: Precision is defined as the fraction of relevant examples (true positives) among all of the examples which were predicted to belong in a certain class.

Precision = True Positives / True Positives + False Positives

3.f1 Score: It is the harmonic mean of precision and recall. It takes both false positive and false negatives into account. Therefore, it performs well on an imbalanced dataset.

For this we need to use the following:

- **True Positive (TP)** — model correctly predicts the positive class (prediction and actual both are positive).
- **True Negative (TN)** — model correctly predicts the negative class (prediction and actual both are negative).
- **False Positive (FP)** — model gives the wrong prediction of the negative class (predicted-positive, actual-negative).
- **False Negative (FN)** — model wrongly predicts the positive class (predicted-negative, actual-positive).

4.Recall: Recall is defined as the fraction of examples which were predicted to belong to a class with respect to all of the examples that truly belong in the class.

Recall = True Positives / True Positives + False Negatives

Thus, f1 score = 2 / (1 / Precision + 1 / Recall)

=2 * (Precision * Recall /Precision + Recall)

The Accuracy and f1 score for our model is:

	precision	recall	f1-score	support
0	0.86	0.84	0.85	1112
1	0.87	0.89	0.88	1388
accuracy			0.87	2500
macro avg	0.87	0.86	0.87	2500
weighted avg	0.87	0.87	0.87	2500

Fig 35: Performance Measure

Our model obtained an F1 score of 0.85 for the female gender and 0.88 for Male gender. So, it classifies male gender better than females.

The term Epoch, is once all the images are processed one time individually of forward and backward to the network. Usually, we feed a neural network the training data for more than one epoch in different patterns by which a better generalization can be there when an unseen input data is given. If there is a large but finite training dataset, then it gives the network a chance to see the previous data to readjust the model parameters so that the model is not biased towards the last few data points during training.

The term Loss, is nothing but a prediction error of neural network and the method to calculate the loss is called loss function. A loss function is used to optimize the machine learning algorithm. The loss is calculated on training and validation sets and its interpretation is based on how well the model is doing in these two sets. It is the sum of errors made for each example in training or validation sets. Loss value implies how poorly or well a model behaves after each iteration of optimization.

An accuracy metric is used to measure the algorithm's performance in an interpretable way. Accuracy of a model is usually determined after the model parameters and is calculated in the form of percentage. It is the measure of how accurate the model prediction is compared to the true data i.e., training data.

6.7 TESTING

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and or/a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations does not fail in unacceptable manner. There are various types of tests. Each test type addresses a specific requirement.

TYPES OF TESTINGS

1. Unit Testing

Unit testing involves the design of test cases that validate the internal program logic is functioning properly, and that program inputs procedure valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is structural testing that relies on knowledge of its construction and is invasive. Unit test perform basic test at component level and test a specific business, application and/or system configuration Unit test ensures that each unique path of princess performs accurately to the documented specifications and contains clearly defined inputs and expected results.

2. Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing in event driven and more concerned with the basic outcome of screens or fields. Integration test demonstrate that although the components were individually satisfaction, as shown successfully by unit testing the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination, of components

3. Functional Testing

Functional test provides systematic demonstrations that function tests are available as specified by the business and technical requirements requirement's, system documentation and user manuals. Functional testing is centred on the following items:

Valid input: identify classes of valid input must be accepted.

Invalid Input: Identify classes of id pus must be rejected,

Functions: Identified be exercised identities function must be exercised

Output: identify classes of application outputs must be exercised

Procedures: interfacing systems or procedures must be invoked

Organization and preparation of functions test is focused on requirements, key functions or special test cases in addition, systematic coverage pertaining to identify business process flow; data fields, predefined process and successive process must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current test is determined.

4. System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results an example on site testing is configuration oriented system integration test.

5. White box Testing

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing. It is a testing in which the software tester has and of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

White box testing involves the testing of the software code for the following:

- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output
- The functionality of conditional loops
- Testing of each statement, object, and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of white box testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

6. Black Box Testing

Black Box Testing a testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kind of tests must be written from a definitive source document, such as specification requirements document. It is a testing in which the software under test is treated as black box you cannot see into it. The test provides inputs and responds to outputs without considering how the software works.

Test plan:

A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process. Follow the below steps to create a test plan as per IEEE 829.

Analyse the system: A system/product can be analysed only when the tester has any information about it i.e., how the system works, who the end users are, what software/hardware the system uses, what the system is for etc.

Design the Test Strategy: Designing a test strategy for all different types of functioning, hardware by determining the efforts and costs incurred to achieve the objectives of the system.

Define the Test Objectives: Test objective is the overall goal and achievement of the test execution. Objectives are defined in such a way that the system is bug-free and is ready to use by the end-users. Test objective can be defined by identifying the software features that are needed to test and the goal of the test, these features need to achieve to be noted as successful.

Define Test Criteria: Test Criteria is a standard or rule on which a test procedure or test judgment can be based. There are two such test criteria: Suspension criteria where if the specific number of test cases are failed, then the tester should suspend all the active test cycle till the criteria is resolved, exit criteria which specifies the criteria that denote a successful completion of a test phase.

Resource Planning: Resource plan is a detailed summary of all types of resources required to complete the project task. Resource could be human, equipment and materials needed to complete a project.

Plan Test Environment: A testing environment is a setup of software and hardware on which the testing team is going to execute test cases.

Schedule & Estimation: Preparing a schedule for different testing stages and estimating the time and man power needed to test the system is mandatory to mitigate the risk of completing the project within the deadline. It includes creating the test specification, test execution, test report, test delivery.

7. CONCLUSION AND FUTURE WORK

Age, gender and other facial traits represent information important to a wide range of tasks. Overall study of gender classification and age estimation can be used in to solve the real-time application problems. The real-time image sensor detection and tracking of the face became a challenge for several researchers. This project demonstrates a system which detects and tracks faces in real time and estimates age and gender. The CNN is used to provide enhanced results of age and gender estimation, even by considering limited training dataset of unconstrained labeled images for age and gender. The simplified network architecture will resolve the issue of over-fitting of data and will yield better results for other training datasets as well as testing real-time images. In this project, most of the research work done is in Convolutional Neural Networks. Though many previous methods have addressed the problems of age and gender classification, much of this work has focused on constraints. The key features of the images are the color and texture of the image. We provide results with a lean deep-learning architecture designed to avoid overfitting due to the limitation of limited labeled data. We further inflate the size of the training data by artificially adding cropped versions of the images in our training set.

Two important conclusions can be made from our results. First, CNN makes the detection of age and gender a lot easier and the performance is also improved a way better. Second, As the CNN is a more elaborate system, so the accuracy of the analysis could be more efficient and standards of the prediction would meet reality. This project includes real time dataset collection, followed by pre-processing and classification. Using those image processing techniques real time face datasets are analyzed and their gender and age is predicted. Performance analysis is done in terms of accuracy where 90.15% is obtained for CNN whereas 87.95% for existing system.

For future works, we will consider a deeper CNN architecture and a more robust image processing algorithm for exact age estimation. Also, the apparent age estimation of human's face will be interesting research to investigate in the future.

8.REFERENCES

[1]. Yunjo Lee, Cheryl L. Grady, Claudine Habak, Hugh R. Wilson, and Morris Moscovitch," Face Processing Changesin Normal Aging Revealed by fMRI Adaptation", 2011 Massachusetts Institute of Technology

[2]. Sarah N. Kohail," Using Artificial Neural Network for Human Age Estimation Based on Facial Images", 2012 International Conference on Innovations in Information Technology.

[3]. Chao Yin and Xin Geng," Facial Age Estimation by Conditional Probability Neural Network", 2012 SpringerVerlag Berlin Heidelberg

[4]. Hang Qi and Liqing Zhang," Age Classification System with ICA Based Local Facial Features", 2009 Springer-Verlag Berlin Heidelberg

[5]. Anusha, A. V., J. K. Jayasree, Anusree Bhaskar, and R. P. Aneesh, "Facial expression recognition and gender classification using facial patches," In 2016 International Conference on Communication Systems and Networks (ComNet), IEEE, 2016.

[6]. Geng, Xin, Zhi-Hua Zhou, and Kate Smith-Miles, "Automatic age estimation based on facial aging patterns," IEEE Transactions on pattern analysis and machine intelligence, 2007.

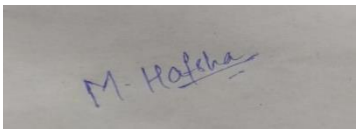
[7]. Da'San, Mohammad, Amin Alqudah, and Olivier Debeir, "Face detection using Viola and Jones method and neural networks," In 2015 International Conference on Information and Communication Technology Research (ICTRC), IEEE, 2015.

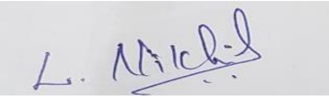
[8]. S.L.Happy and A. Routray,"Automatic facial expression recognition using features of salient facial patches," in IEEE Transactions on Affective Computing, vol, 1 Jan.-March 2015.

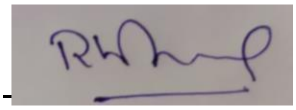
[9]. T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition.

[10]. F. H. C. Tivive and A. Bouzerdoum. A gender recognition system using shunting inhibitory convolutional neural networks. Proceedings of the International Joint Conference on Neural Networks (IJCNN '06).

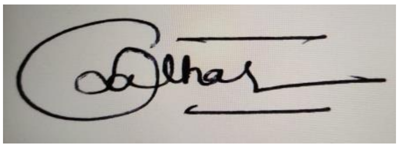
BATCH 12C-TEAM MEMBERS SIGNATURE

1)Md. Hafsha Firdous (317126510152)- 

2)L. Nikhil (317126510149)- 

3)R. Bhanu Praveen(317126510161)- 

GUIDE SIGNATURE

G. Sathar - 

Age and Gender Classification using Convolutional Neural Networks

Gil Levi and Tal Hassner
Department of Mathematics and Computer Science
The Open University of Israel
gil.levi100@gmail.com hassner@openu.ac.il

Abstract

Automatic age and gender classification has become relevant to an increasing amount of applications, particularly since the rise of social platforms and social media. Nevertheless, performance of existing methods on real-world images is still significantly lacking, especially when compared to the tremendous leaps in performance recently reported for the related task of face recognition. In this paper we show that by learning representations through the use of deep-convolutional neural networks (CNN), a significant increase in performance can be obtained on these tasks. To this end, we propose a simple convolutional net architecture that can be used even when the amount of learning data is limited. We evaluate our method on the recent Adience benchmark for age and gender estimation and show it to dramatically outperform current state-of-the-art methods.

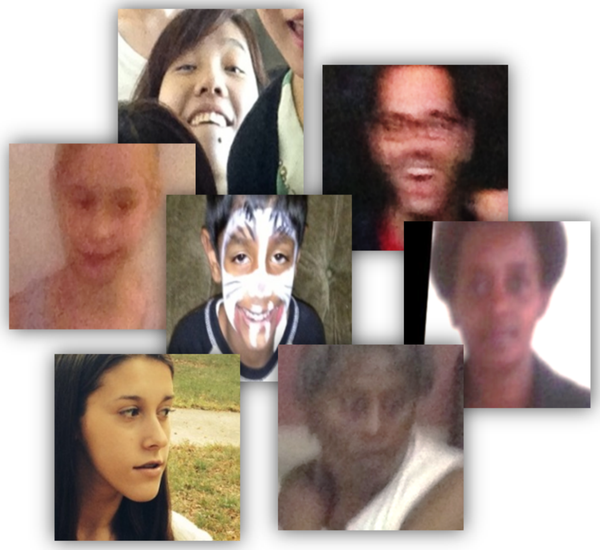


Figure 1. **Faces from the Adience benchmark for age and gender classification [10].** These images represent some of the challenges of age and gender estimation from real-world, unconstrained images. Most notably, extreme blur (low-resolution), occlusions, out-of-plane pose variations, expressions and more.

1. Introduction

Age and gender play fundamental roles in social interactions. Languages reserve different salutations and grammar rules for men or women, and very often different vocabularies are used when addressing elders compared to young people. Despite the basic roles these attributes play in our day-to-day lives, the ability to automatically estimate them accurately and reliably from face images is still far from meeting the needs of commercial applications. This is particularly perplexing when considering recent claims to super-human capabilities in the related task of face recognition (e.g., [48]).

Past approaches to estimating or classifying these attributes from face images have relied on differences in facial feature dimensions [29] or "tailored" face descriptors (e.g., [10, 15, 32]). Most have employed classification schemes designed particularly for age or gender estimation tasks, including [4] and others. Few of these past methods were designed to handle the many challenges of unconstrained imaging conditions [10]. Moreover, the machine learning methods employed by these systems did not fully exploit the massive numbers of image examples and data available through the Internet in order to improve classification capabilities.

In this paper we attempt to close the gap between automatic face recognition capabilities and those of age and gender estimation methods. To this end, we follow the successful example laid down by recent face recognition systems: Face recognition techniques described in the last few years have shown that tremendous progress can be made by the use of deep convolutional neural networks (CNN) [31]. We demonstrate similar gains with a simple network architecture, designed by considering the rather limited availability of accurate age and gender labels in existing face data sets.

We test our network on the newly released Adience benchmark for age and gender classification of unfiltered face images [10]. We show that despite the very challenging nature of the images in the Adience set and the simplicity of our network design, our method outperforms existing state of the art by substantial margins. Although these results provide a remarkable baseline for deep-learning-based approaches, they leave room for improvements by more elaborate system designs, suggesting that the problem of accurately estimating age and gender in the unconstrained settings, as reflected by the Adience images, remains unsolved. In order to provide a foothold for the development of more effective future methods, we make our trained models and classification system publicly available. For more information, please see the project webpage www.openu.ac.il/home/hassner/projects/cnn_agegender.

2. Related Work

Before describing the proposed method we briefly review related methods for age and gender classification and provide a cursory overview of deep convolutional networks.

2.1. Age and Gender Classification

Age classification. The problem of automatically extracting age related attributes from facial images has received increasing attention in recent years and many methods have been put fourth. A detailed survey of such methods can be found in [11] and, more recently, in [21]. We note that despite our focus here on age group *classification* rather than precise age estimation (i.e., age regression), the survey below includes methods designed for either task.

Early methods for age estimation are based on calculating ratios between different measurements of facial features [29]. Once facial features (e.g. eyes, nose, mouth, chin, etc.) are localized and their sizes and distances measured, ratios between them are calculated and used for classifying the face into different age categories according to hand-crafted rules. More recently, [41] uses a similar approach to model age progression in subjects under 18 years old. As those methods require accurate localization of facial features, a challenging problem by itself, they are unsuitable for in-the-wild images which one may expect to find on social platforms.

On a different line of work are methods that represent the aging process as a subspace [16] or a manifold [19]. A drawback of those methods is that they require input images to be near-frontal and well-aligned. These methods therefore present experimental results only on constrained data-sets of near-frontal images (e.g UIUC-IFP-Y [12, 19] ,FG-NET [30] and MORPH [43]). Again, as a consequence, such methods are ill-suited for unconstrained images.

Different from those described above are methods that use local features for representing face images. In [55] Gaussian Mixture Models (GMM) [13] were used to represent the distribution of facial patches. In [54] GMM were used again for representing the distribution of local facial measurements, but robust descriptors were used instead of pixel patches. Finally, instead of GMM, Hidden-Markov-Model, super-vectors [40] were used in [56] for representing face patch distributions.

An alternative to the local image intensity patches are robust image descriptors: Gabor image descriptors [32] were used in [15] along with a Fuzzy-LDA classifier which considers a face image as belonging to more than one age class. In [20] a combination of Biologically-Inspired Features (BIF) [44] and various manifold-learning methods were used for age estimation. Gabor [32] and local binary patterns (LBP) [1] features were used in [7] along with a hierarchical age classifier composed of Support Vector Machines (SVM) [9] to classify the input image to an age-class followed by a support vector regression [52] to estimate a precise age.

Finally, [4] proposed improved versions of relevant component analysis [3] and locally preserving projections [36]. Those methods are used for distance learning and dimensionality reduction, respectively, with Active Appearance Models [8] as an image feature.

All of these methods have proven effective on small and/or constrained benchmarks for age estimation. To our knowledge, the best performing methods were demonstrated on the Group Photos benchmark [14]. In [10] state-of-the-art performance on this benchmark was presented by employing LBP descriptor variations [53] and a dropout-SVM classifier. We show our proposed method to outperform the results they report on the more challenging Adience benchmark, designed for the same task.

Gender classification. A detailed survey of gender classification methods can be found in [34] and more recently in [42]. Here we quickly survey relevant methods.

One of the early methods for gender classification [17] used a neural network trained on a small set of near-frontal face images. In [37] the combined 3D structure of the head (obtained using a laser scanner) and image intensities were used for classifying gender. SVM classifiers were used by [35], applied directly to image intensities. Rather than using SVM, [2] used AdaBoost for the same purpose, here again, applied to image intensities. Finally, viewpoint-invariant age and gender classification was presented by [49].

More recently, [51] used the Webers Local texture Descriptor [6] for gender recognition, demonstrating near-perfect performance on the FERET benchmark [39]. In [38], intensity, shape and texture features were used with mutual information, again obtaining near-perfect results on the FERET benchmark.

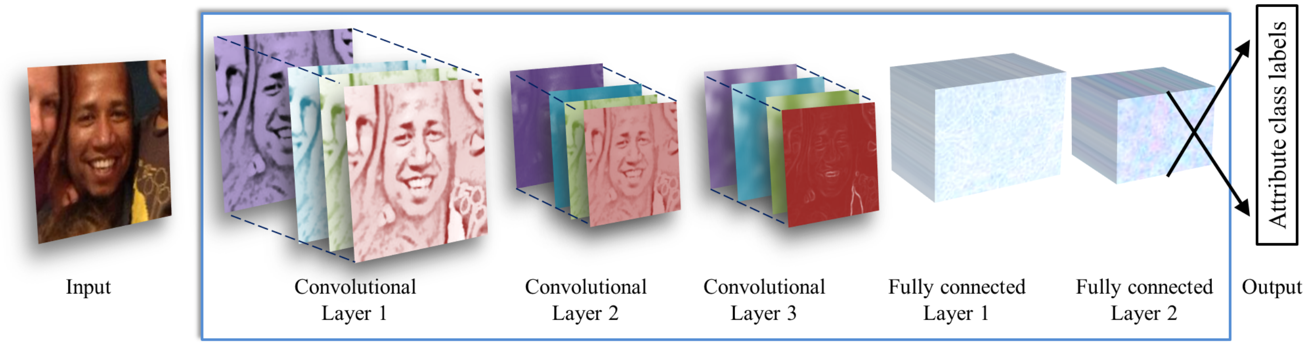


Figure 2. **Illustration of our CNN architecture.** The network contains three convolutional layers, each followed by a rectified linear operation and pooling layer. The first two layers also follow normalization using local response normalization [28]. The first Convolutional Layer contains 96 filters of 7×7 pixels, the second Convolutional Layer contains 256 filters of 5×5 pixels, The third and final Convolutional Layer contains 384 filters of 3×3 pixels. Finally, two fully-connected layers are added, each containing 512 neurons. See Figure 3 for a detailed schematic view and the text for more information.

Most of the methods discussed above used the FERET benchmark [39] both to develop the proposed systems and to evaluate performances. FERET images were taken under highly controlled condition and are therefore much less challenging than in-the-wild face images. Moreover, the results obtained on this benchmark suggest that it is saturated and not challenging for modern methods. It is therefore difficult to estimate the actual relative benefit of these techniques. As a consequence, [46] experimented on the popular Labeled Faces in the Wild (LFW) [25] benchmark, primarily used for face recognition. Their method is a combination of LBP features with an AdaBoost classifier.

As with age estimation, here too, we focus on the Adience set which contains images more challenging than those provided by LFW, reporting performance using a more robust system, designed to better exploit information from massive example training sets.

2.2. Deep convolutional neural networks

One of the first applications of convolutional neural networks (CNN) is perhaps the LeNet-5 network described by [31] for optical character recognition. Compared to modern deep CNN, their network was relatively modest due to the limited computational resources of the time and the algorithmic challenges of training bigger networks.

Though much potential laid in deeper CNN architectures (networks with more neuron layers), only recently have they became prevalent, following the dramatic increase in both the computational power (due to Graphical Processing Units), the amount of training data readily available on the Internet, and the development of more effective methods for training such complex models. One recent and notable examples is the use of deep CNN for image classification on the challenging Imagenet benchmark [28]. Deep CNN have additionally been successfully applied to applications including human pose estimation [50], face parsing [33], facial keypoint detection [47], speech recognition [18] and action classification [27]. To our knowledge, this is the first report of their application to the tasks of age and gender classification from unconstrained photos.

3. A CNN for age and gender estimation

Gathering a large, *labeled* image training set for age and gender estimation from social image repositories requires either access to personal information on the subjects appearing in the images (their birth date and gender), which is often private, or is tedious and time-consuming to manually label. Data-sets for age and gender estimation from real-world social images are therefore relatively limited in size and presently no match in size with the much larger image classification data-sets (e.g. the Imagenet dataset [45]). Overfitting is common problem when machine learning based methods are used on such small image collections. This problem is exacerbated when considering deep convolutional neural networks due to their huge numbers of model parameters. Care must therefore be taken in order to avoid overfitting under such circumstances.

3.1. Network architecture

Our proposed network architecture is used throughout our experiments for both age and gender classification. It is illustrated in Figure 2. A more detailed, schematic diagram of the entire network design is additionally provided in Figure 3. The network comprises of only three convolutional layers and two fully-connected layers with a small number of neurons. This, by comparison to the much larger architectures applied, for example, in [28] and [5]. Our choice of a smaller network design is motivated both from our desire to reduce the risk of overfitting as well as the nature

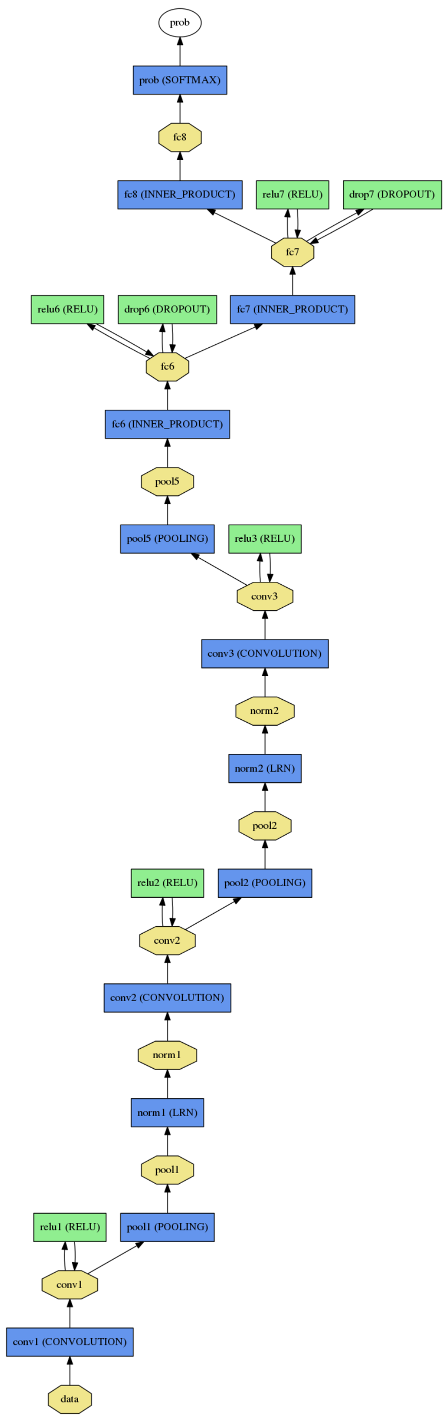


Figure 3. **Full schematic diagram of our network architecture.** Please see text for more details.

of the problems we are attempting to solve: age classification on the Adience set requires distinguishing between eight classes; gender only two. This, compared to, e.g., the ten thousand identity classes used to train the network used for face recognition in [48].

All three color channels are processed directly by the network. Images are first rescaled to 256×256 and a crop of 227×227 is fed to the network. The three subsequent convolutional layers are then defined as follows.

1. 96 filters of size $3\times7\times7$ pixels are applied to the input in the first convolutional layer, followed by a rectified linear operator (ReLU), a max pooling layer taking the maximal value of 3×3 regions with two-pixel strides and a local response normalization layer [28].

2. The $96 \times 28 \times 28$ output of the previous layer is then processed by the second convolutional layer, containing 256 filters of size $96 \times 5 \times 5$ pixels. Again, this is followed by ReLU, a max pooling layer and a local response normalization layer with the same hyper parameters as before.

3. Finally, the third and last convolutional layer operates on the $256 \times 14 \times 14$ blob by applying a set of 384 filters of size $256 \times 3 \times 3$ pixels, followed by ReLU and a max pooling layer.

The following fully connected layers are then defined by:

4. A first fully connected layer that receives the output of the third convolutional layer and contains 512 neurons, followed by a ReLU and a dropout layer.

5. A second fully connected layer that receives the 512-dimensional output of the first fully connected layer and again contains 512 neurons, followed by a ReLU and a dropout layer.

6. A third, fully connected layer which maps to the final classes for age or gender.

Finally, the output of the last fully connected layer is fed to a soft-max layer that assigns a probability for each class. The prediction itself is made by taking the class with the maximal probability for the given test image.

3.2. Testing and training

Initialization. The weights in all layers are initialized with random values from a zero mean Gaussian with standard deviation of 0.01. To stress this, we do not use pre-trained models for initializing the network; the network is trained, from scratch, without using any data outside of the images and the labels available by the benchmark. This, again, should be compared with CNN implementations used for face recognition, where hundreds of thousands of images are used for training [48].

Target values for training are represented as sparse, binary vectors corresponding to the ground truth classes. For each training image, the target, label vector is in the length of the number of classes (two for gender, eight for the eight age classes of the age classification task), containing 1 in the index of the ground truth and 0 elsewhere.

Network training. Aside from our use of a lean network architecture, we apply two additional methods to further limit the risk of overfitting. First we apply dropout learning [24] (i.e. randomly setting the output value of network neurons to zero). The network includes two dropout layers with a dropout ratio of 0.5 (50% chance of setting a neuron's output value to zero). Second, we use data-augmentation by taking a random crop of 227×227 pixels from the 256×256 input image and randomly mirror it in each forward-backward training pass. This, similarly to the multiple crop and mirror variations used by [48].

Training itself is performed using stochastic gradient decent with image batch size of fifty images. The initial learning rate is $e^{-}3$, reduced to $e^{-}4$ after 10K iterations.

Prediction. We experimented with two methods of using the network in order 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 (occlusions, 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 rather directly feeding the network with multiple translated versions of the same face.

4. Experiments

Our method is implemented using the Caffe open-source framework [26]. 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.

4.1. The Adience benchmark

We test the accuracy of our CNN design using the recently released Adience benchmark [10], designed for age and gender classification. The Adience set consists of images automatically uploaded to Flickr from smart-phone devices. Because these images were uploaded without prior manual filtering, as is typically the case on media webpages (e.g., images from the LFW collection [25]) or social websites (the Group Photos set [14]), viewing conditions in these images are highly unconstrained, reflecting many of the real-world challenges of faces appearing in Internet images. Adience images therefore capture extreme variations in head pose, lightning conditions quality, and more.

The entire Adience collection includes roughly 26K images of 2,284 subjects. Table 1 lists the breakdown of the collection into the different age categories. Testing for both age or gender classification is performed using a standard five-fold, subject-exclusive cross-validation protocol, defined in [10]. We use the in-plane aligned version of the faces, originally used in [10]. These images are used rater than newer alignment techniques in order to highlight the performance gain attributed to the network architecture, rather than better preprocessing.

We emphasize that the same network architecture is used for all test folds of the benchmark and in fact, for both gender and age classification tasks. This is performed in order to ensure the validity of our results across folds, but also to demonstrate the generality of the network design proposed here; the same architecture performs well across different, related problems.

We compare previously reported results to the results computed by our network. Our results include both methods for testing: center-crop and over-sampling (Section 3).

4.2. Results

Table 2 and Table 3 presents our results for gender and age classification respectively. Table 4 further provides a confusion matrix for our multi-class age classification results. 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.

Both tables compare performance with the methods described in [10]. Table 2 also provides a comparison with [23] which used the same gender classification pipeline of [10] applied to more effective alignment of the faces; faces in their tests were synthetically modified to appear facing forward.

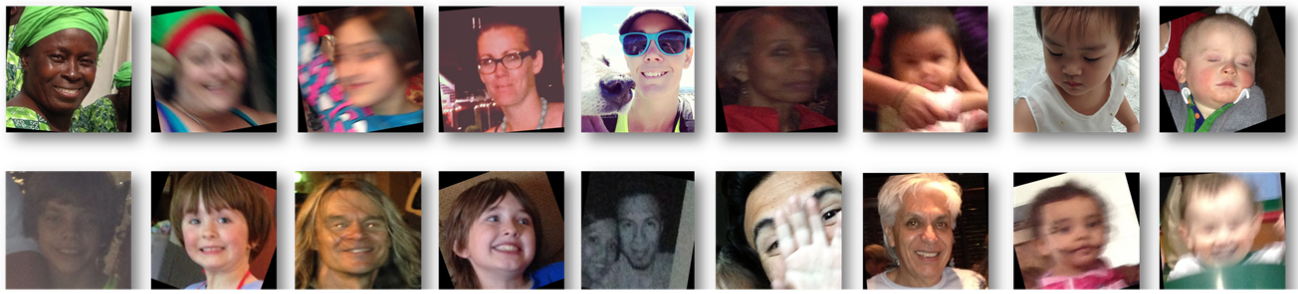


Figure 4. **Gender misclassifications.** Top row: Female subjects mistakenly classified as males. Bottom row: Male subjects mistakenly classified as females

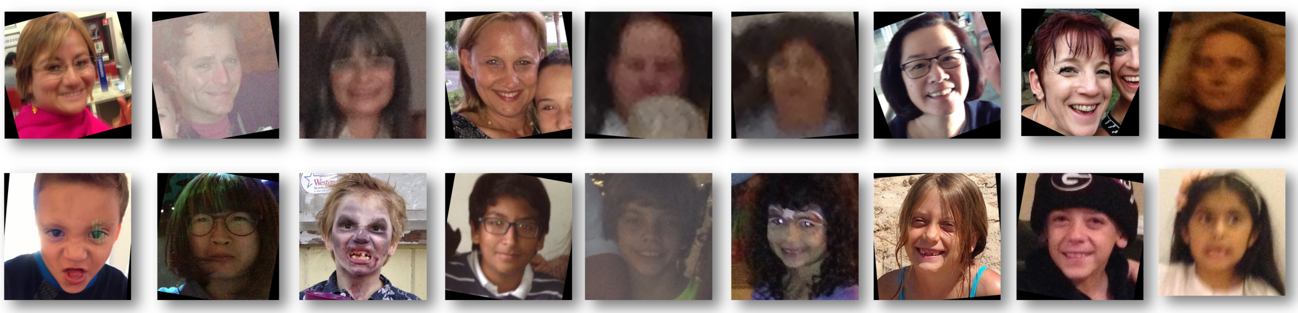


Figure 5. **Age misclassifications.** Top row: Older subjects mistakenly classified as younger. Bottom row: Younger subjects mistakenly classified as older.

	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 1. **The AdienceFaces benchmark.** Breakdown of the AdienceFaces benchmark into the different Age and Gender classes.

Method	Accuracy
Best from [10]	77.8 $\pm$ 1.3
Best from [23]	79.3 $\pm$ 0.0
Proposed using single crop	85.9 $\pm$ 1.4
Proposed using over-sample	**86.8** $\pm$ **1.4**

Table 2. **Gender estimation results on the Adience benchmark.** Listed are the mean accuracy $\pm$ standard error over all age categories. Best results are marked in bold.

Method	Exact	1-off
Best from [10]	45.1 $\pm$ 2.6	79.5 $\pm$1.4
Proposed using single crop	49.5 $\pm$ 4.4	84.6 $\pm$ 1.7
Proposed using over-sample	**50.7** $\pm$ **5.1**	**84.7** $\pm$ **2.2**

Table 3. **Age estimation results on the Adience benchmark.** Listed are the mean accuracy $\pm$ standard error over all age categories. Best results are marked in bold.

Evidently, the proposed method outperforms the reported state-of-the-art on both tasks with considerable gaps. Also evident is the contribution of the over-sampling approach, which provides an additional performance boost over the original network. This implies that better alignment (e.g., frontalization [22, 23]) may provide an additional boost in performance.

We provide a few examples of both gender and age misclassifications in Figures 4 and 5, respectively. These show that many of the mistakes made by our system are due to extremely challenging viewing conditions of some of the Adience benchmark images. Most notable are mistakes caused by blur or low resolution and occlusions (particularly from heavy makeup). Gender estimation mistakes also frequently occur for images of babies or very young children where obvious gender attributes are not yet visible.

5. Conclusions

Though many previous methods have addressed the problems of age and gender classification, until recently, much of this work has focused on constrained images taken in lab settings. Such settings do not adequately reflect appearance variations common to the real-world images in social websites and online repositories. Internet images, however, are not simply more challenging: they are also abun-

	0-2	4-6	8-13	15-20	25-32	38-43	48-53	60-
0-2	**0.699**	0.147	0.028	0.006	0.005	0.008	0.007	0.009
4-6	0.256	**0.573**	0.166	0.023	0.010	0.011	0.010	0.005
8-13	0.027	0.223	**0.552**	0.150	0.091	0.068	0.055	0.061
15-20	0.003	0.019	0.081	**0.239**	0.106	0.055	0.049	0.028
25-32	0.006	0.029	0.138	0.510	**0.613**	0.461	0.260	0.108
38-43	0.004	0.007	0.023	0.058	0.149	**0.293**	0.339	0.268
48-53	0.002	0.001	0.004	0.007	0.017	0.055	**0.146**	0.165
60-	0.001	0.001	0.008	0.007	0.009	0.050	0.134	**0.357**

Table 4. **Age estimation confusion matrix on the Adience benchmark.**

dant. The easy availability of huge image collections provides modern machine learning based systems with effectively endless training data, though this data is not always suitably labeled for supervised learning.

Taking example from the related problem of face recognition we explore how well deep CNN perform on these tasks using Internet data. We provide results with a lean deep-learning architecture designed to avoid overfitting due to the limitation of limited labeled data. Our network is "shallow" compared to some of the recent network architectures, thereby reducing the number of its parameters and the chance for overfitting. We further inflate the size of the training data by artificially adding cropped versions of the images in our training set. The resulting system was tested on the Adience benchmark of unfiltered images and shown to significantly outperform recent state of the art.

Two important conclusions can be made from our results. First, CNN can be used to provide improved age and gender classification results, even considering the much smaller size of contemporary unconstrained image sets labeled for age and gender. Second, the simplicity of our model implies that more elaborate systems using more training data may well be capable of substantially improving results beyond those reported here.

Acknowledgments

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA 2014-14071600010. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright annotation thereon.

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

[2] S. Baluja and H. A. Rowley. Boosting sex identification performance. *Int. J. Comput. Vision*, 71(1):111–119, 2007. 2

[3] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *Int. Conf. Mach. Learning*, volume 3, pages 11–18, 2003. 2

[4] W.-L. Chao, J.-Z. Liu, and J.-J. Ding. Facial age estimation based on label-sensitive learning and age-oriented regression. *Pattern Recognition*, 46(3):628–641, 2013. 1, 2

[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. 3

[6] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao. Wld: A robust local image descriptor. *Trans. Pattern Anal. Mach. Intell.*, 32(9):1705–1720, 2010. 2

[7] S. E. Choi, Y. J. Lee, S. J. Lee, K. R. Park, and J. Kim. Age estimation using a hierarchical classifier based on global and local facial features. *Pattern Recognition*, 44(6):1262–1281, 2011. 2

[8] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *European Conf. Comput. Vision*, pages 484–498. Springer, 1998. 2

[9] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 2

[10] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. *Trans. on Inform. Forensics and Security*, 9(12), 2014. 1, 2, 5, 6

[11] Y. Fu, G. Guo, and T. S. Huang. Age synthesis and estimation via faces: A survey. *Trans. Pattern Anal. Mach. Intell.*, 32(11):1955–1976, 2010. 2

[12] Y. Fu and T. S. Huang. Human age estimation with regression on discriminative aging manifold. *Int. Conf. Multimedia*, 10(4):578–584, 2008. 2

[13] K. Fukunaga. *Introduction to statistical pattern recognition*. Academic press, 1991. 2

[14] A. C. Gallagher and T. Chen. Understanding images of groups of people. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 256–263. IEEE, 2009. 2, 5

[15] F. Gao and H. Ai. Face age classification on consumer images with gabor feature and fuzzy lda method. In *Advances in biometrics*, pages 132–141. Springer, 2009. 1, 2

[16] X. Geng, Z.-H. Zhou, and K. Smith-Miles. Automatic age estimation based on facial aging patterns. *Trans. Pattern Anal. Mach. Intell.*, 29(12):2234–2240, 2007. 2

[17] B. A. Golomb, D. T. Lawrence, and T. J. Sejnowski. Sexnet: A neural network identifies sex from human faces. In *Neural Inform. Process. Syst.*, pages 572–579, 1990. 2

[18] A. Graves, A.-R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013. 3

[19] G. Guo, Y. Fu, C. R. Dyer, and T. S. Huang. Image-based human age estimation by manifold learning and locally adjusted robust regression. *Trans. Image Processing*, 17(7):1178–1188, 2008. 2

[20] G. Guo, G. Mu, Y. Fu, C. Dyer, and T. Huang. A study on automatic age estimation using a large database. In *Proc. Int. Conf. Comput. Vision*, pages 1986–1991. IEEE, 2009. 2

[21] H. Han, C. Otto, and A. K. Jain. Age estimation from face images: Human vs. machine performance. In *Biometrics (ICB), 2013 International Conference on*. IEEE, 2013. 2

[22] T. Hassner. Viewing real-world faces in 3d. In *Proc. Int. Conf. Comput. Vision*, pages 3607–3614. IEEE, 2013. 6

[23] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. *Proc. Conf. Comput. Vision Pattern Recognition*, 2015. 5, 6

[24] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 5

[25] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007. 3, 5

[26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[27] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 1725–1732. IEEE, 2014. 3

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Inform. Process. Syst.*, pages 1097–1105, 2012. 3, 4

[29] Y. H. Kwon and N. da Vitoria Lobo. Age classification from facial images. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 762–767. IEEE, 1994. 1, 2

[30] A. Lanitis. The FG-NET aging database, 2002. Available: `www-prima.inrialpes.fr/FGnet/html/benchmarks.html`. 2

[31] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 1, 3

[32] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *Trans. Image Processing*, 11(4):467–476, 2002. 1, 2

[33] P. Luo, X. Wang, and X. Tang. Hierarchical face parsing via deep learning. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 2480–2487. IEEE, 2012. 3

[34] E. Makinen and R. Raisamo. Evaluation of gender classification methods with automatically detected and aligned faces. *Trans. Pattern Anal. Mach. Intell.*, 30(3):541–547, 2008. 2

[35] B. Moghaddam and M.-H. Yang. Learning gender with support faces. *Trans. Pattern Anal. Mach. Intell.*, 24(5):707–711, 2002. 2

[36] X. Niyogi. Locality preserving projections. In *Neural Inform. Process. Syst.*, volume 16, page 153. MIT, 2004. 2

[37] A. J. O'toole, T. Vetter, N. F. Troje, H. H. Bülthoff, et al. Sex classification is better with three-dimensional head structure than with image intensity information. *Perception*, 26:75–84, 1997. 2

[38] C. Perez, J. Tapia, P. Estévez, and C. Held. Gender classification from face images using mutual information and feature fusion. *International Journal of Optomechatronics*, 6(1):92–119, 2012. 2

[39] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and vision computing*, 16(5):295–306, 1998. 2, 3

[40] L. Rabiner and B.-H. Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986. 2

[41] N. Ramanathan and R. Chellappa. Modeling age progression in young faces. In *Proc. Conf. Comput. Vision Pattern Recognition*, volume 1, pages 387–394. IEEE, 2006. 2

[42] D. Reid, S. Samangooei, C. Chen, M. Nixon, and A. Ross. Soft biometrics for surveillance: an overview. *Machine learning: theory and applications. Elsevier*, pages 327–352, 2013. 2

[43] K. Ricanek and T. Tesafaye. Morph: A longitudinal image database of normal adult age-progression. In *Int. Conf. on Automatic Face and Gesture Recognition*, pages 341–345. IEEE, 2006. 2

[44] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019–1025, 1999. 2

[45] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014. 3

[46] C. Shan. Learning local binary patterns for gender classification on real-world face images. *Pattern Recognition Letters*, 33(4):431–437, 2012. 3

[47] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 3476–3483. IEEE, 2013. 3

[48] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 1891–1898. IEEE, 2014. 1, 4, 5

[49] M. Toews and T. Arbel. Detection, localization, and sex classification of faces from arbitrary viewpoints and under occlusion. *Trans. Pattern Anal. Mach. Intell.*, 31(9):1567–1581, 2009. 2

[50] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 1653–1660. IEEE, 2014. 3

[51] I. Ullah, M. Hussain, G. Muhammad, H. Aboalsamh, G. Bebis, and A. M. Mirza. Gender recognition from face images with local wld descriptor. In *Systems, Signals and Image Processing*, pages 417–420. IEEE, 2012. 2

[52] V. N. Vapnik and V. Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998. 2

[53] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *post-ECCV Faces in Real-Life Images Workshop*, 2008. 2

[54] S. Yan, M. Liu, and T. S. Huang. Extracting age information from local spatially flexible patches. In *Acoustics, Speech and Signal Processing*, pages 737–740. IEEE, 2008. 2

[55] S. Yan, X. Zhou, M. Liu, M. Hasegawa-Johnson, and T. S. Huang. Regression from patch-kernel. In *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2008. 2

[56] X. Zhuang, X. Zhou, M. Hasegawa-Johnson, and T. Huang. Face age estimation using patch-based hidden markov model supervectors. In *Int. Conf. Pattern Recognition*. IEEE, 2008. 2

IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 5, May 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

9940 572 462 6381 907 438 ijircce@gmail.com www.ijircce.com

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905244 |

Analysis and Detection of Age and Gender using Deep Heuristic

G Sathar[1], Surya Pratibha[2], Hafsha Firdous[3], Lalam Nikhil[4], Bhanu Praveen[5]

Assistant Professor, Dept. of CSE, ANITS, Visakhapatnam, AP, India[1]

UG Student, Dept. of CSE, ANITS, Visakhapatnam, AP, India[2]

UG Student, Dept. of CSE, ANITS, Visakhapatnam, AP, India[3]

UG Student, Dept. of CSE, ANITS, Visakhapatnam, AP, India[4]

UG Student, Dept. of CSE, ANITS, Visakhapatnam, AP, India[5]

ABSTRACT: During this modern era, there is a drastic increase in technologies where Age and Gender detection became relevant and widely used in applications like social platforms and social media. Face detection and recognition play a crucial part in detecting age and gender. The attribute information such as age and gender plays a huge role in improving the accuracy and performance of face recognition. However, existing technologies on authentic image and gender classification lacks performance and efficiency. The main aim of this paper is to overcome this challenge and give a brief view on how classification is done using deep learning for which we used a Convolutional neural network (CNN) which performs well even when huge tasks are given. Deep-Convolutional neural networks study the face images of a person on which the CNN model has been trained. Many methods were proposed by researchers in the past for age estimation and gender detection but there were some disadvantages such as partial reflection of faces which caused a problem in detecting the age and gender of a person accurately. We used the UTKFace dataset which consists of various categories of real-world face images which were used for training our CNN model. To the end of this project, we have proposed a simple convolutional network architecture that increases the performance, accuracy, and efficiency even when the training data is limited.

KEYWORDS: Attribute information, Deep-Convolutional neural networks, face detection, face recognition, gender classification, UTKFace dataset.

I. INTRODUCTION

The analysis of age and gender detection has been playing an important role in this advanced world where machines are used to do the work instead of human beings.The reason why most people prefer machines is they are easy to operate and we can maintain them easily and the quality of the work will be more accurate when compared to the work done by a person. The major role of “analysis and detection of age” is to classify the gender and age of a person. Based on our daily life, we communicate with different kinds of people and based on age and gender the way we communicate to a person differs. The very first thing that a person does while meeting a new person is to classify his or her gender. This classification is mainly based on a person’s facial features. Face recognition plays a vital role in the surveillance sector. Mostly face recognition is used in places where high security must be provided to keep the place away from intruders. When a new face is detected the CNN trained models immediately recognizes and alerts the guards. By using face recognition, we can monitor at what place a particular individual is present and at what time, this makes it easier to keep an eye on a particular individual. In this way, face recognition is used in security surveillance to identify a particular person or to recognize his face and try to get his details, etc.

As humans are well versed to classify the gender of a person but a machine could not classify that instantly. To make this possible we developed a model and trained the model with different data sets to classify the gender of a person. So, to develop the model here we used Convolutional Neural Network (CNN) and deep learning to estimate age and gender detection. The model uses a computer that has a camera which is used to detect the face and the trained model scans the image. To make this work we trained the model with a data set that has a sufficient amount of data and each time the computer recognizes a face it uses its data to extract facial features and detects the gender and age of a person by comparing the person’s image with the trained data set. For this model we added another extension, with this we can also give a physical image to estimate its age and gender detection. The model is a high-level security feature

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905244 |

that can be an asset to the surveillance system so that, if any security breach had happened we can detect the person's face using this facial recognition and detect his gender too. This could help the organization to find the targeted person easily. The main purpose of the project is to make a system capable enough to recognize a person and to detect the gender and age of the person just like humans.

DEEP LEARNING:

Deep learning is a subset of machine learning which is used in artificial intelligence, it has networks that can learn from unsupervised data that is unstructured or unlabeled. It is also known as a deep neural network; the AI is independent therefore it can learn without human supervision.The use of deep learning is we can collect an enormous amount of data which is also known as “Big Data” and this data is collected from sources like social media, search engines, e-commerce platforms, but not all data is useful. Generally, what deep learning does is takes this unstructured-unsupervised data and learn from the data by itself. Here, we are using raw data which is unstructured and by using deep learning algorithms the AI differentiates the differences and adapts according to the data whereas in machine learning we have to train the model by providing data sets that are structured and labeled and the model gets prepared accordingly. Similar to the human brain the deep learning also uses a structure of algorithms that are multi-layered called neural networks. The main goal of deep learning algorithms is to enable computers that can mimic human behavior. The algorithms try to make similar decisions as humans do by repeatedly analyzing that has a logical structure. The architecture of neural networks is based on the structure of the human brain. Just like humans perceive data to identify patterns and classify the data accordingly the neural networks can be used to perform the tasks just like a human does in the data. As similar to how a human brain works whenever we receive new info the brain tries to compare with the known objects in the same way the concept of the neural network is designed. Neural networks help us to perform different tasks like clustering, classification, or regression. With this we can sort unstructured data into structured data with the given samples (OR) we can train the network by using a labeled data set to classify the objects in the data set into different categories.AI can solve problems that machine learning can never solve. The recent advancement of technology is due to the deep learning algorithms, without them we would not have personal assistants like Alexa, Siri. And some other examples are chatbots, self-driving cars, Google lenses. The advertisements we see on our mobile and the movie suggestions we get in an OTT platform are all made possible with the help of neural networks.The main advantage of deep learning over machine learning is a feature called feature extraction. Feature extraction is usually a complex process that is embedded in the deep learning system. It requires a piece of thorough knowledge of the problem statement. Then, the preprocessing layer must be altered, checked, and processes over several steps for ideal results.

CONVOLUTIONAL NEURAL NETWORKS:

CNN stands for Convolutional Neural Network which is a neural network that is specialized in processing data that has the input like 2d images which is used to classify or recognize the image. On seeing a new image, we can scan the image from any direction or any angle to find different features of the image and then we bring together all the features that we scanned to classify the image. This is how CNN works. The translational invariant feature which is a property of CNN helps in the recognition of an image irrespective of the size or the rotation the objects will be recognized so we don’t need to consider the image transformations like rotation, deformations. Recently we discovered that CNN has a large capacity to perform sequent data analysis like natural language processing.CNN is a unique form of deep neural network which is used to process data that have multiple arrays and grid-like topology. CNN's can be used on 3D (video), 2D (image), and 1D (text or audio) input data to perform in deep learning applications. It can draw out high-degree features from raw input features, which are much more robust than human-designed features. Thus, it has brought remarkable advances to several fields—for example, image segmentation and recognition. CNN is a type of radial basis function neural network in AI which is widely used for image recognition and the input data is of the form of multidimensional arrays. The more data we have the more effective the CNN.CNN has a feature named receptive field which is to extract every portion of the input image. CNN network has an input and an output layer and in between, there are multiple hidden layers. These hidden layers consist of a series of convolutional layers. This makes the training of the model completely computerized and the performance is better than the manual work done. It can also be used for various types of image recognition problems and image types.The efficiency of the CNN can be determined by the number of layers and the size of the network. Therefore, this technique is limited by computing power and the availability of larger data.

II. RELATED WORK

[1] Identifying age and gender became very prominent in recent times. it is prominent as well as gained significant attention recently due to the rise in social media. Yunjo Lee proposed that the fMRI method is employed to review age detection methods. The study involves a correct recording of the variations of individuals based on their changes consistent with age, gender, identity, and other features. The brain activation tasks associated with face matching are performed and tested outside

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905244 |

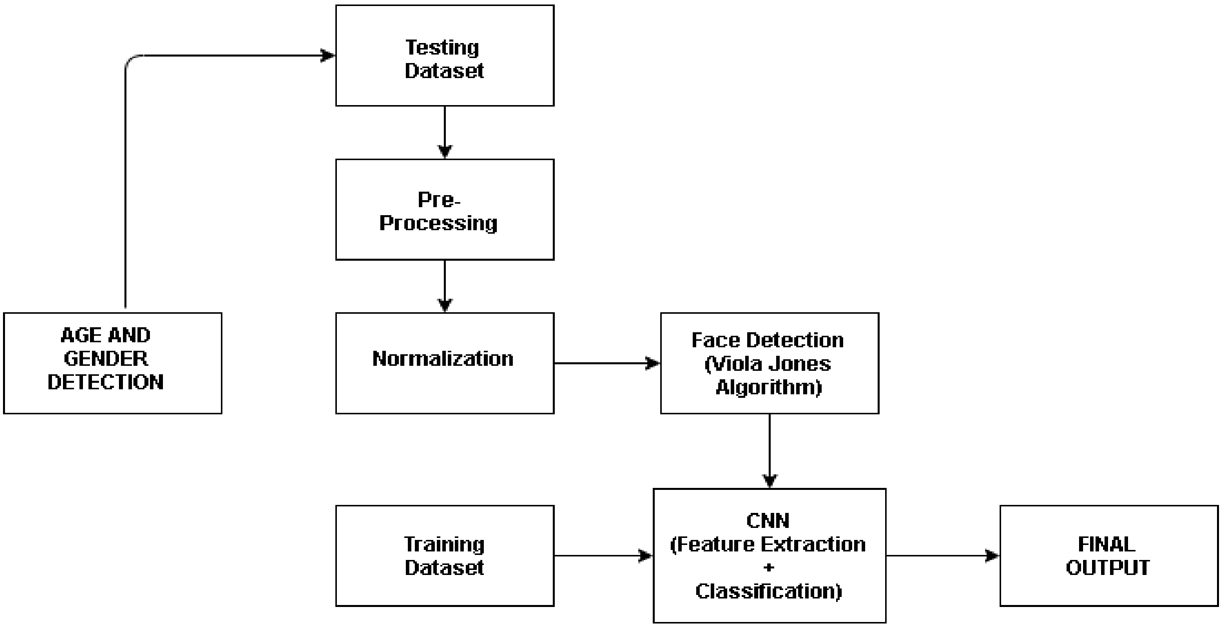
the scanner. There was the same end in face processing in older also as young adults. The results based on the performance are high in the two cases having the same facial viewpoints. The aging of the elders isn't supported anybody factor. It is a combination of varied factors that end in accountancy of such results. The results got to be kept a track on which are supported all credentials kept in certain environments.

[2] Sarah N. Kohail proposed that age estimation is now the present challenge being faced. Here, this paper finds the right support to move forward in the approach of neural networks to estimate the age of persons. Most of the change that has been made during this method is the fine-tuning of the age ranges. To find out the multi-layer perception neural networks (MLP) the countenance of the new images were extracted and recorded. The results have shown the MLP method as an honest method with minimum errors within the results. These results are often utilized in many applications like age-based access control applications and also within the age adaptive human-machine interaction. The upgrades are to be made within the system, where the system is to be made more automatic, and also the numbers of input facial features that are provided to be reduced.

[3] Chao Yin proposed that the Conditional Probability Neural Network (CPNN) is a distribution learning algorithm used for age estimation using facial expressions. It follows the three-layer neural network system in which the target values and the conditional feature vectors are used as input. This can help it in learning the real ages. The relationship between the face image and the related label distribution through the neural network is used as the learning method for this system. CPNN has proved to be providing better results than all the previously made methods. Through this method the results provided were very easy, there was less computational involved, and the outcomes very efficient. Due to all such advantages, it was preferred more than the others.

[4] Hang Qi proposed that various techniques are arising for the detection of faces which may also identify the age of the person. Here, an automatic system has been proposed which may classify the age and help distinguishing kids face from that of an adult face. There are three parts that the system encompasses. The first part is face detection the second one is face alignment and normalization and the last is one is age classification. Face samples are created by the traditional face detection and alignment methods. ICA is employed for the extraction of the local facial components that are present within the images. this technique has been proved to be much faster and the results are efficient. So this technique is often utilized in the future as a prototype.

III. SYSTEM ARCHITECTURE



TESTING DATASET: A testing dataset also called a Real-time dataset is the first step in age and gender detection where a webcam is triggered and is switched on to detect the face of a person. Webcam is triggered by using THE capture function which is used in OpenCV. Once the face is detected then, it is pre-processed in the later stage.

PREPROCESSING: This is used to enhance the quality of the image by converting the image into grayscale and then by reducing the noise in the image which is caused by the environment. It is the lowest level of abstraction where it smoothens the image without any noise by filtration methods.

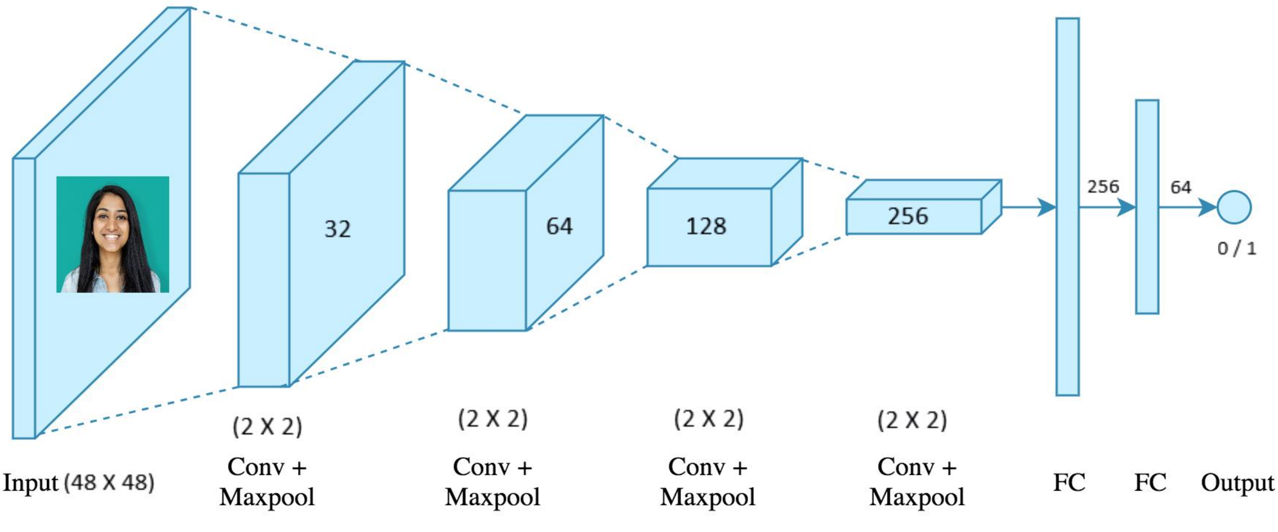
NORMALIZATION: It is a technique that is used to produce a uniform size image by cropping the detected part of the face into a rectangular box which eliminates unnecessary parts in the image. It is then used to detect every part of the face like the eyes, nose, lips, etc.

FACE DETECTION: For detecting the face we used the Viola-Jones algorithm which is the best Object detection algorithm which is used to detect the faces of a person from the input image. In this algorithm, we select Haar-like features where all the pixels are added and then make this into an Integral image where only the boundary pixels are added. The last two steps involve Adaboosting where training is done and the final step is Cascading the image.

FEATURE EXTRACTION: It gives meaningful information to the input image by calculating the distance between the eyeballs, the distance between nose and chin, etc. After feature extraction, classification of the image is done where the probability is calculated using Softmax or SVM to predict the age and gender based on the Training dataset where the CNN model gets trained. Finally giving the accurate output.

IV. NETWORK ARCHITECTURE

Our proposed network architecture comprises four convolutional layers and two fully connected layers comprising a finite amount of neurons. The main motivation of our project is to design a small architecture that can increase performance and reduce the risk of overfitting.



The input image is first rescaled to a size of 48 X 48 and then fed to the convolutional neural network model where it then detects the age and gender of a person from the set of images that we have given for training.

1.32 filters of size 24 X 24 pixels are applied to the input in the first convolutional layer, followed by a rectified linear operator (ReLU), a max-pooling layer that takes an input of 2 × 2 regions with a one-pixel stride.

2. The output of the first layer is then fed to the second convolutional layer, containing 64 filters of size 12 X 12 pixels which are again followed by ReLU and a max-pooling layer.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905244 |

3. The third convolutional layer operates by applying a set of 128 filters of size 6 X 6 pixels, followed by ReLU and a max-pooling layer.

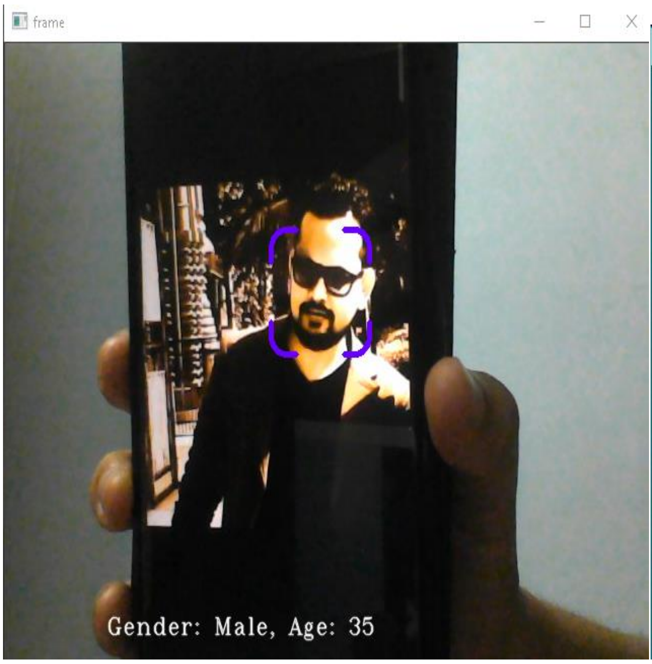
4. Finally, the fourth convolutional layer i.e the last layer contains 256 filters of size 3 X 3 pixels followed by ReLU and max-pooling similar to the first three layers as discussed above.

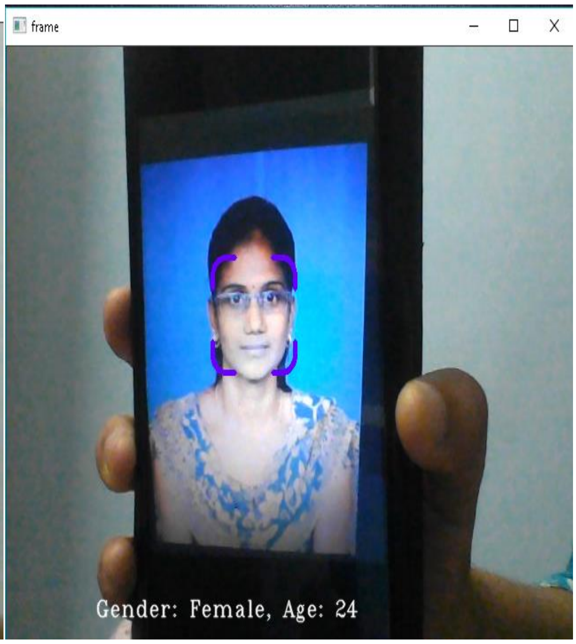
5. The first fully connected layer receives the output of the fourth convolutional layer which contains 256neurons, followed by a ReLU and a dropout layer.

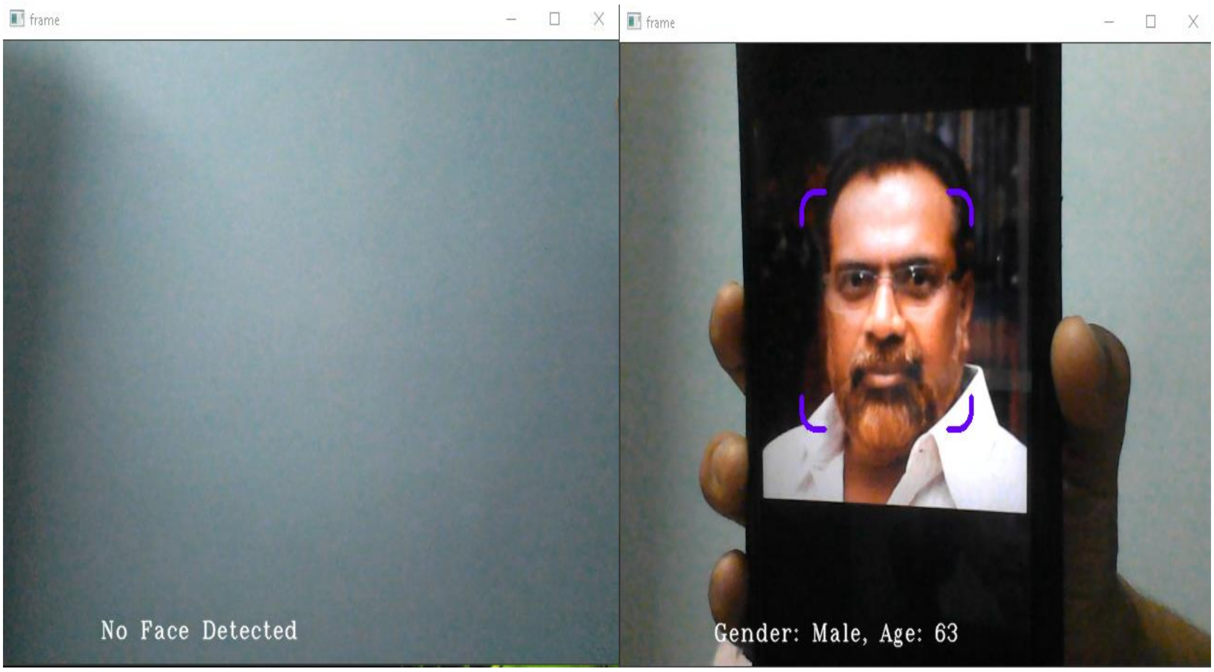
6. The second fully connected layer that receives output from the first fully connected layer contains 64 neurons, followed by a ReLU and a dropout layer.

Finally, the output of the second fully connected layer is fed to a soft-max which is a probability distribution function calculated for each class that predicts the age and gender by taking the class with the maximum probability of either 1 or 0 for the given test image where 0 indicates Female and 1 indicates the gender Male.

V. RESULTS







VI. CONCLUSION AND FUTURE WORK

CNN makes the detection of age and gender easier and also improves performance. As CNN is a more elaborate system, so the accuracy of the analysis could be more efficient and standards of the prediction would meet reality. Age and gender represent very important information of a wide range of tasks. An overall study of gender classification and age estimation can be used to solve real-time application problems. The real-time image sensor detection and tracking of the face became a challenge for several researchers. This project demonstrates a system that detects and tracks faces in real-time and estimates age and gender. The CNN is used to provide enhanced results of age and gender estimation, even by considering a limited training dataset of unconstrained labeled images for age and gender. The simplified network architecture will resolve the issue of over-fitting of data and will yield better results for other training datasets as well as testing real-time images. In this project, most of the research work done is in Convolutional neural networks. Though many previous methods have addressed the problems of age and gender classification, much of this work has focused on constraints. The key features of the images are the color and texture of the image. We provide results with a lean deep-learning architecture that is designed to avoid overfitting. After that, we increase the size of the training data by adding some cropped images to our training set. Two important conclusions can be made from our results. First, CNN makes the detection of age and gender a lot easier and the performance is also improved a way better. Second, as CNN a more elaborate system, so the accuracy of the analysis could be more efficient and standards of the prediction would meet reality. This project includes real-time dataset collection, followed by pre-processing and classification. Using these image processing techniques real-time face datasets are analyzed and their gender and age are predicted. Performance analysis is done in terms of accuracy where 90.15% is obtained for CNN whereas 87.95% for the existing system.

REFERENCES

[1]. Yunjo Lee, Cheryl L. Grady, Claudine Habak, Hugh R. Wilson, and Morris Moscovitch," Face Processing Changesin Normal Aging Revealed by fMRI Adaptation", 2011 Massachusetts Institute of Technology

[2]. Sarah N. Kohail," Using Artificial Neural Network for Human Age Estimation Based on Facial Images", 2012 International Conference on Innovations in Information Technology.

[3]. Chao Yin and Xin Geng," Facial Age Estimation by Conditional Probability Neural Network", 2012 SpringerVerlag Berlin Heidelberg

[4]. Hang Qi and Liqing Zhang," Age Classification System with ICA Based Local Facial Features", 2009 Springer-Verlag Berlin Heidelberg C. Romero and S. Ventura, ''Educational data mining: A survey from 1995 to 2005,'' *Expert Syst. Appl.*, vol. 33, no. 1, pp. 135–146, Jul. 2007.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905244 |

[5] R. Ghorbani and R. Ghousi, ''Predictive data mining approaches in medical diagnosis: A review of some diseases prediction,'' *Int. J. Data Netw. Sci.*, vol. 3, no. 2, pp. 47–70, 2019.
[6] D. Kabakchieva, ''Student performance prediction by using data mining classification algorithms,'' *Int. J. Comput. Sci. Manage. Res.*, vol. 1, no. 4, pp. 686–690, 2012.
[7] A. I. Adekitan and E. Noma-Osaghae, ''Data mining approach to predicting the performance of first year student in a University using the admis- sion requirements,'' *Educ. Inf. Technol.*, vol. 24, no. 2, pp. 1527–1543, Mar. 2019.
[8] E. Fernandes, M. Holanda, M. Victorino, V. Borges, R. Carvalho, and G. V. Erven, ''Educational data mining: Predictive analysis of academic performance of public school students in the capital of Brazil,'' *J. Bus. Res.*, vol. 94, pp. 335–343, Jan. 2019.
[9] R. S. Baker and K. Yacef, ''The state of educational data mining in 2009: A review and future visions,'' *J. Educ. Data Mining*, vol. 1, no. 1, pp. 3–17, 2009.
[10] E. Chandra and K. Nandhini, ''Knowledge mining from student data,'' *Eur. J. Sci. Res.*, vol. 47, no. 1, pp. 156–163, 2010.
[11] A. B. El Din Ahmed and I. S. Elaraby, ''Data mining: A prediction for student's performance using classification method,'' *World J. Comput. Appl. Technol.*, vol. 2, no. 2, pp. 43–47, 2014.
[12] M. M. A. Tair and A. M. El-Halees, ''Mining educational data to improve students' performance: A case study,'' *Int. J. Inf. Commun. Technol. Res.*, vol. 2, no. 2, pp. 1–7, 2012.
[13] H. A. A. Hamza and P. Kommers, ''A review of educational data mining tools & techniques,'' *Int. J. Educ. Technol. Learn.*, vol. 3, no. 1, pp. 17–23, 2018.