Human Age Detection through Image Processing

Abstract

Recently, the need for human age detection has risen in many applications because of the emergence of social media platforms. Nevertheless, the performance of the already existing techniques used to apply the real-world image data is still lower and bad for image-related tasks like facial prediction and recognition, etc. We will apply a deep learning model in this project known as Convolutional Neural Network (CNN) to increase the performance of image-related data and tasks [1]. We will propose a solution using evolutional deep learning technique which can improve the performance for these tasks and can also produce the same performance if the data is restricted. We will access and produce our model according to the recent standards for human age predictions and will show how well it will perform as compared to other methods.

Chapter 1 Introduction

1.1. Overview

Because in evaluation of internet and social media platforms, Human age act as important needs of the day in social interactions. Because every language has its own set of grammatical rules and solutions for the interactions we made in our daily life like it has its own rules for the interaction and communication of women and men, children and old people .etc. In today's world, we are still far from gaining the ability to detect or predict these social interactions by using facial images in our applications accurately [2]. This task of facial recognition is still a mystery. In these past years, different methods are used to detect and predict human age by using facial features, for example, some of these applications use classification algorithms made specifically for this kind of task.

Many Machine learning techniques are applied in the past but they cannot meet the challenge of different imaging situations and errors like low-resolution images, different kinds of poses and variations in the acts, and many more. As the facial feature is a key for this purpose of prediction and detection of human age. In recent years the boom of Artificial Intelligence and its algorithms of AI algos has shown great development in the field of face recognition and human age predictions. In this project, we intend to make a revaluation in automatic face recognition and human age prediction.

In this project, we will produce a different but simple method that can be more reliable and accurate than the past methods. Our method will be applied and tested on the most recent facial datasets considering a very low availability of labeled data. It will show that the complex imaging conditions and the simplest road map of our method will show the great gap of improvement in this field as compared to the past [3]. This technique will open the doors to more advancement and improvement in the field of facial recognition and human age classification fields.

1.2. Description

The description explained and is as follows:-

1.2.1. Objectives

We have made some points for the objectives as mentioned below:-

- The main purpose is to help the user and make their easy when determining human age.
- If we see the project of making a framework that can detect human age by facial images is very difficult because of the complex nature of different facial image features.

- So our main goal is to design an architecture that can easily deal with the complex facial features and predict more accurately and it will be more reliable to do this job.
- We will use the deep learning technique known as Convolutional Neural network for this purpose.

1.2.2. Problem Description

In the past, many techniques are applied to the problem of human age prediction but they are unable to perform well because of the complexity of the different facial features, for example, low-resolution images, different kinds of poses, and variations in the acts, and many more. As the facial feature is a key for this purpose of prediction and detection of human age. In recent years the boom of AI with its different techniques and algorithms has shown great development in the field of face recognition and human age predictions. In this project, we intend to make a revaluation in automatic face recognition and human age prediction and we will make simple but effective architecture and framework to efficiently detect the human age form facial images.

1.2.3. Methodology

We will purpose AI-based framework which will use Deep Convolutional Neural Network (CNN) for the complex facial imaging features and extraction purpose for the detection of human age because in the past the failure cause of the methods used for human age prediction problems and facial recognition problems is the complexity of the facial images and their features [4]. The CNN will perform outstandingly for this purpose and it is more reliable for feature extraction from complex images than the other method used in the past. Our deep learning model allows us to focus on the complex and vital informative parts of the face. we will train our model in a synchronized and parallelized way which helps us to be more effective, and more reliable and it will enhance the accuracy of our predictions according to the system's needs.

1.2.4. Product Scope

The need for automatic human age prediction is the need of the day in many applications of today’s world because of its many applications in lots of fields. So, will create such a framework that aids all the applications needed to detect human age from facial images. In this project, we intend to make a revaluation in automatic face recognition and human age prediction and we will make simple but effective architecture and framework to efficiently detect the human age form facial images.

1.2.5. Integrated Applications Environment

The project system needs the following requirenments as follow:

- **Least Necessities**
  - **Operating Systems:** Windows
  - **Processor:** Intel Core i5 3rd generation processor or better
  - **Memory:** 4GB

1.3. Hypotheses and Reliance's

AI-based Models will be used for this system in a control environment which will help in good study and investigating the problem and the facial image dataset will be used for this system

1.4. Hardware and Software Specifications

The Hardware and Software requirements for the system are as follows:

1.4.1. Hardware

- Any type of computer machine fulfilling the software requirements.

1.4.2. Software

Human age prediction from images is a deep learning problem. The following tool will be used in the development of the algorithm:

- Anaconda 3 2020.11(64 bit)
- Jupiter notebook
- PyCharm

1.5. System Features

Our anticipation is to make such a system which uses AI-based recent models for prediction of human age. These AI-based algos can automatically and deterministically acquire the features from the images. These algos are more precise and accurate then the past algos were in use.

1.5.1. Significance and Response Sequence

The system will obtain the image from dataset. Then it will preprocess it to extract features and then it will classify and predict.

Chapter 2 Literature Review

2.1. Introduction

From a historic point of view, the research on the topic of human age estimation has been going on for ages. Many solutions have been introduced during the time with a significantly low level of gain and low results. In the early ages, the research focused on manually classifying and extracting features of facial details and using the difference in these features for prediction purposes. These kinds of approaches focus on the following features for example size, distance or gap .etc.

In the past, the research on human age estimation can be found the early 1990s when neural networks are used for this purpose [2]. After that, we found that in the early 2000s the past algos were used to detect and classify human age prediction and they gained some accuracy when predicting the blur images. However, the bad implementation and characterization of their data stop their systems from getting improvement in performance and accuracy for the complex imaging data. But every method was able to serve in single-purpose like either age estimation or estimation.

So, for the first time in history, this custom was changed in early 2015 [2], when a framework was generated by the researchers which can estimate both human age prediction and they also found that every real-world image has different features and different alignment settings. They focus on simplifying the image before getting it to classify like its different features and other things and they use a different variety of angles when they trained their images for this purpose they use different computer vision techniques. These techniques help them gain significantly good performance as compared to other methods.

2.2. Related Works

Machine learning is a technique in which we can manually feature extraction tasks and used it for many purposes like classification, prediction problems, etc. Whereas deep learning can automatically extract features from the dataset and best for many images related problems. In the past since 2012, researchers from ImageNet competition has found that these deep learning algorithms perform well and give good performance than the other algorithm for image-related problems. A researcher in the competition named A. Krizhevsky et al. [4]were able to find a method that uses a CNN for the classification of the images having a range of up to 1000 classes achieving the test error rate below 15.3% which was a good improvement and the 2nd best entry in the competition.

Many early techniques are applied in the past but they cannot meet the challenge of different imaging situations and errors like low-resolution images, different kinds of poses and variations in the acts, and many more. As the facial feature is a key for this purpose of prediction and detection of human age. For example, in early 1999, [4]Kwon and Lobo fucoses on the different variations in the features of facial images like dimensions and were able to find a fast solution for estimating human age problems. But there were some flaws in their system that it cannot classify the difference between young adults and old adults' people. So, in 2004, a researcher name Lanita et al. give the solution to this problem by focusing on both geometrical and texture features by using a technique known as the Active Appearance Model (AMM). This technique does not produce good results if the dataset includes different variational facial images.

2.2.1. Terminology

Following are the terminologies used in this project:

- **Deep Learning Models:** Convolutional Neural Network
- **Domain:** Deep Learning
- **Dataset**: Facial images Datasets of different age groups
- Facial Images of different age groups

2.2.2. Categorization of Existing Techniques/Works/Research

In the past many frameworks are reveled that they only focus and used old |AI-based classification algos like different old ML and old DL algos were applied to certain facial images for the purpose of classification or detection of human age. But, we will use this innovative methodology which will defiantly make a mark to the world for this problem.

2.2.3. Proposed Improvements in Existing Works

This sort of research has been not done before, and it provides a revolutionary deep learning approach for the diagnosis of human age. Our strategy is primarily designed to make the life of a common person easier. When it comes to predicting human age, our technologies have significantly reduced time consummation. Additionally, our technology has the potential to reduce inter-observer variability and different dimensionality in the facial features when it comes to predicting the human age from the facial images.

2.3. Summary

According to the findings of this study CNN models are used to design a method for the automated classification of human age. These Deep Learning models are used to develop the proposed detection model we will be using and for the training of the model, the Facial images dataset of different age groups will be used. This study demonstrates that deep learning has the potential to be used to create a detecting classifier that may give near real-time detection of labeling for inappropriate facial images by being trainable on a library of facial images taken from online sites.

Chapter 3 System Design

3.1. Introduction

The project is a AI research-based project as we have dicussed it in earlier chapters and its objective is to mechanizes and systemize the classification and prediction of human age groups,so we have so little for the discussion in the project because it's a research-based project and have almost small UI.

3.1.1. Purpose

This chapter will make its accommodation just a little bit by making a road-map for the project of human age prediction using images data. The chapter will help us design some activity diagrams and work flow diagrams for the project.

3.1.2. System Overview

This project aims to build a model based on Deep learning techniques to predict the age group and gender of a person without any human effort. The project is mainly to build and train a model which can recognize a pattern of a different facial images of different age groups for this purpose, a dataset is available that contains facial images of different age groups data. The model will take these images as input and decide the human age of that facial image.

3.1.3. Design Map

To build the model, real-time facial images will be given to the model then the image will be preprocessed, and then object detection will be performed on it. After successful object detection feature extraction will be performed to predict the human age.

3.2. Design Considerations

There is not much consideration for the design as it is a research-based project.

3.2.1. Assumptions

The project assumes that the environment is a facial image that discriminates against their human age.

3.2.2. Constraints

Data should be in the form of facial images, other forms of data would not be process-able for the model and might not be able to perform any useful task.

3.2.3. Risks and Volatile Areas

The project does not have any risks and volatile areas infect the project aims public safety.

3.2.4. Risk Mitigation

There is no risk mitigation in the project.

3.3. Activity Diagram

The below workflow diagram and activity diagrams shows the implementation process of this project and the steps of how it will work and predict the human age of images.

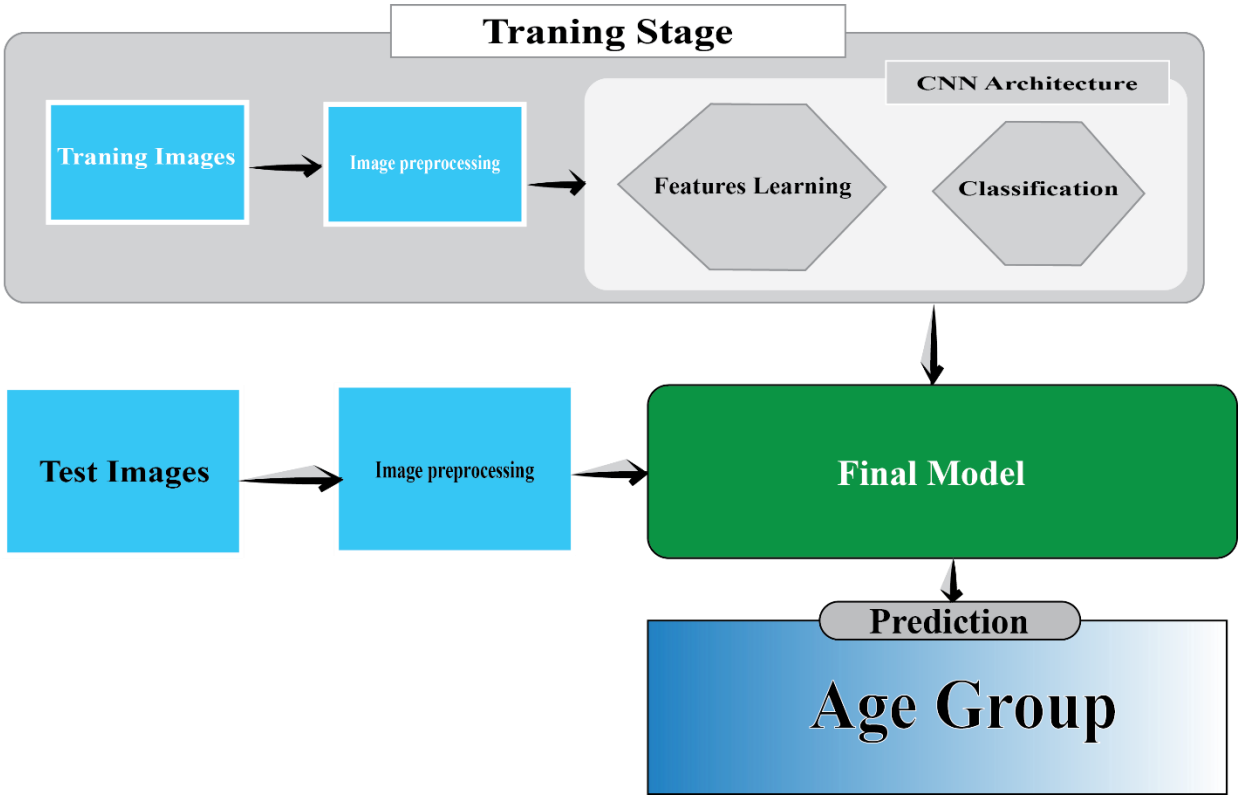

Figure 3. 1 Workflow Diagram

The above figure 3.1 is a Workflow diagram of the project which shows the two stages, the first stage is the training stage in which first we enter the training data of the program which are the facial images and 70% of the overall dataset then next is image processing in which image features are extracted and cleaning the image, etc., then feature learning will be performed and at last step of the training stage classification will be performed where CNN will be applied and then at the final model the age is estimated and data will be trained on these steps. The next stage is the testing phase where 30% of the dataset will be tested and the result will be analyzed and we will

check the accuracy of the model on this data. Below shown figure 3.2 is activity diagram which shows the working of our project in very simple steps.

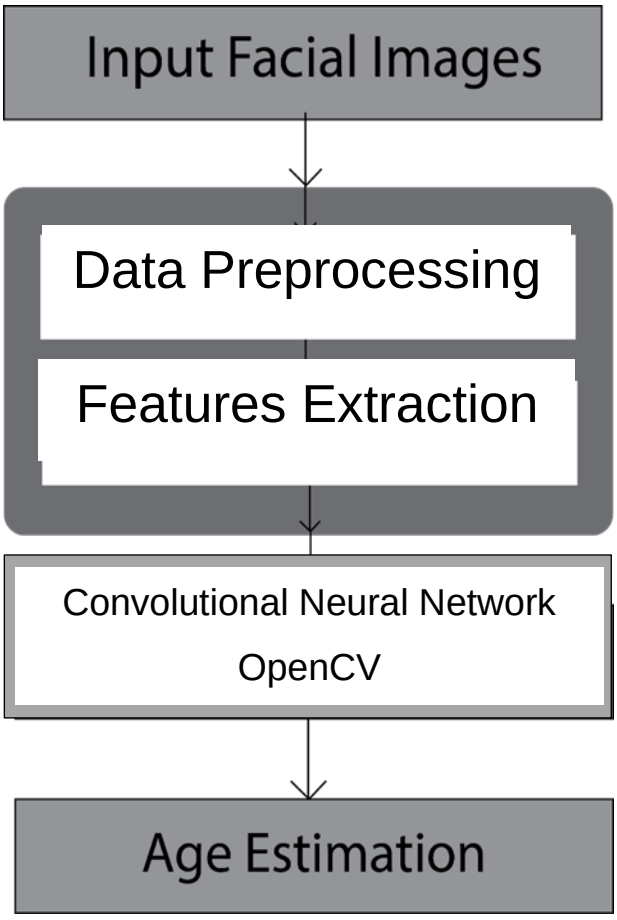

Figure 3. 2 Activity Diagram

The above figure 3.2 is an activity diagram of the project which shows how this project will work, basically, we can say that it's a methodology of the project in the first step the facial images are input, and then the features are extracted after that the model is applied and then the system predicted the age on real-time.

Below in figure 3.3 shows the deployment diagram of the human age detection using image processing. To deploy a human age detection system using CNN and OpenCV first Collect and annotate a dataset of images containing human faces and their corresponding ages. Pre-process the images to ensure that they are all of the same size and format. Split the dataset into a training set and a test set. Train a deep learning model known as CNN using the training set. Test the

model on the test set to evaluate its performance. Deploy the trained model to a server or device that can process live video streams or images in real time.

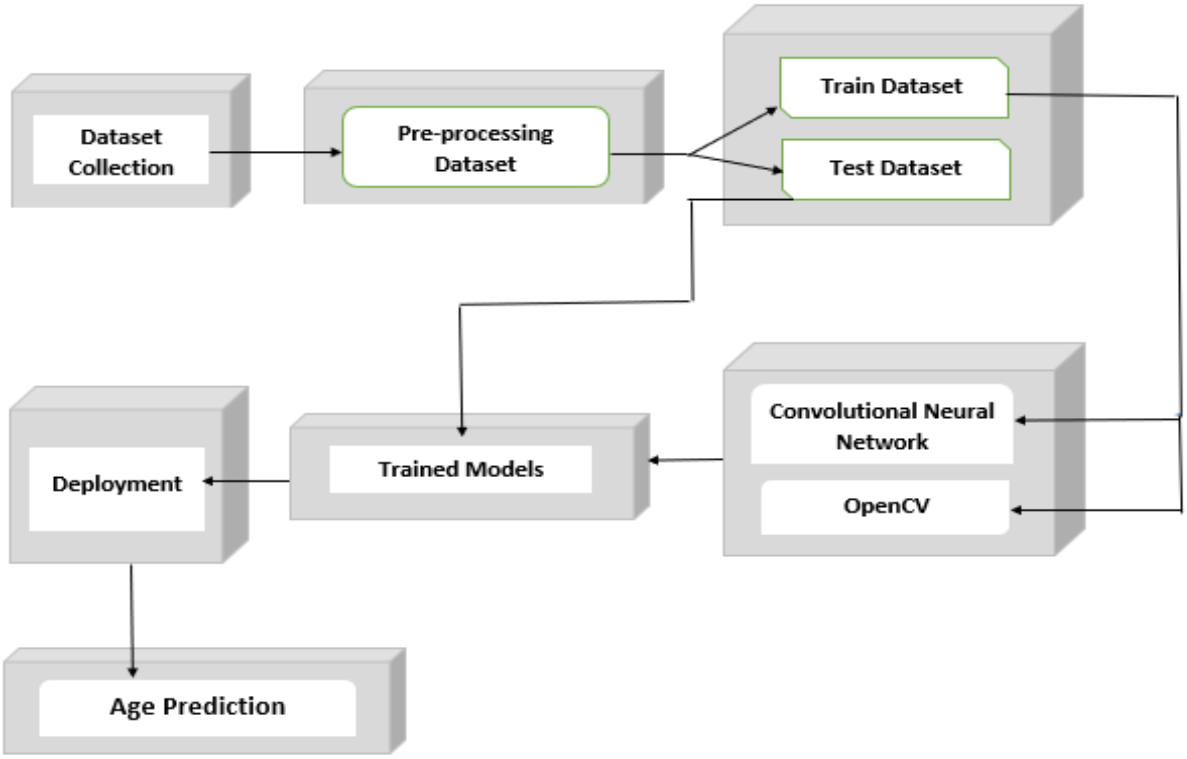

Figure 3. 3 Deployment Diagram

Now below in figure 3.4 shows the Architecture diagram of a system that could be used to detect the age of a human in an image using CNN and OpenCV:

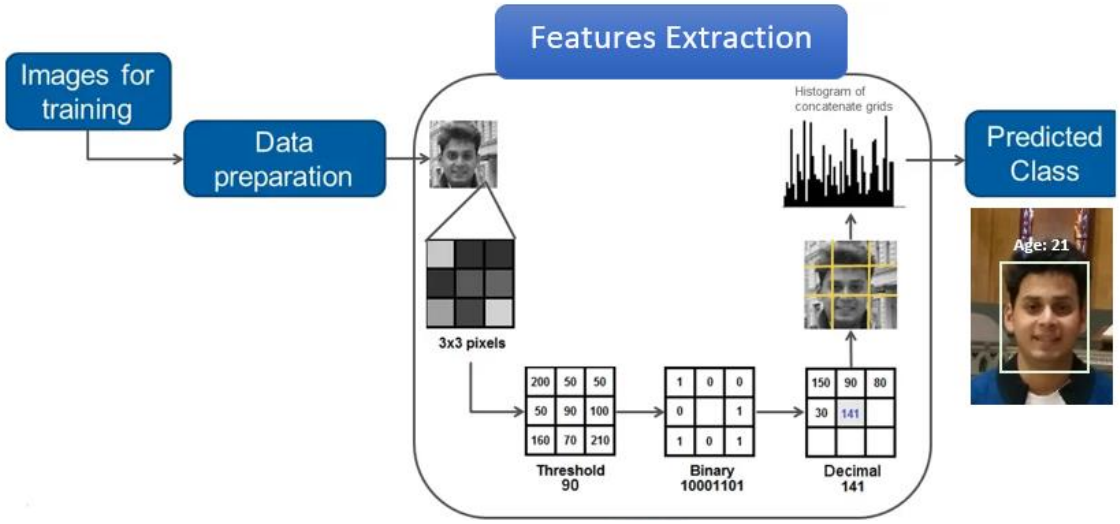

Figure 3. 4 Architecture Diagram

The above image shows the architectural steps for the system of human age detection. The first step is **Input:** An image containing a human face. **Preprocessing:** Ensure that they are all of the same size and format and clean the dataset and convert it to grayscale, and apply any necessary filtering or image enhancement techniques. **Face detection:** The preprocessed image is then passed through a face detection algorithm, which locates and crops the face from the image. **Age estimation:** The cropped face image is then passed through an age estimation model, which will predict the age of the person in the image. **Output:** The age of the person in the image is displayed on the screen.

Chapter 4 Implementation

4.1. Discussion

We are using the facial images of different age groups. Because this is a classification problem and the dataset is based on images, we use Convolutional Neural Network for the prediction of the age group and its . On the first phase of the project, we work on our dataset. We upload the data and we extract features from it.

4.2. Implementation Methodologies

This system for the prediction and estimation of age groups will be accomplished in different steps. Firstly, we pre-process our data, after that we will use AI-based algorithm Convolutional Neural Network for the prediction of age groups.

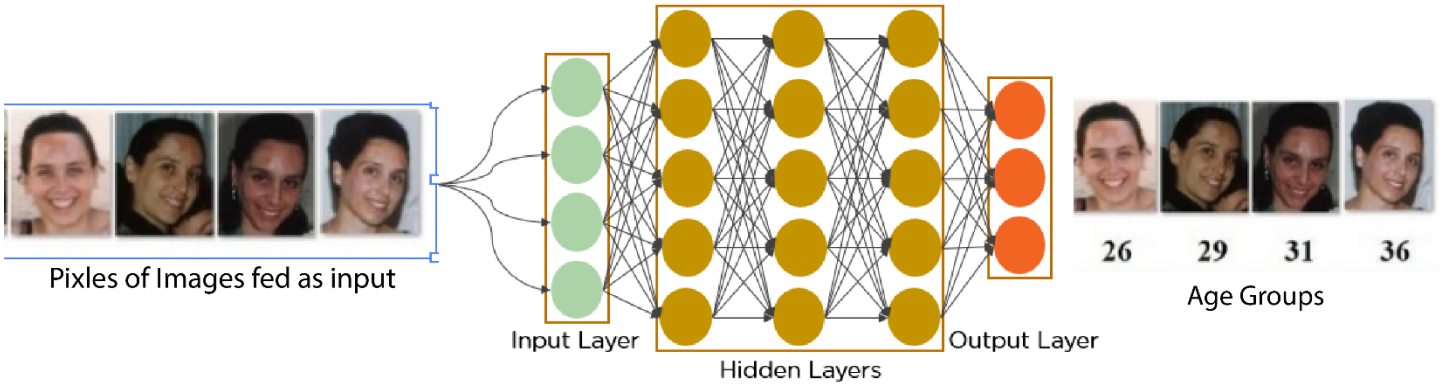

Figure 4. 1 Development Methodology

Convolutional neural networks power image recognition and computer vision tasks. Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs, and based on those inputs, it can take action. This ability to provide recommendations distinguishes it from image recognition tasks

4.3. Implementation Tools and Technologies

We have chosen this project because we have the knowledge, skills and courage to complete this project. We will use anaconda for the programming purposes and pythos as programming language.

4.3.1. Technologies

Following are the technologies used in this project:

- **Hardware**

- Personal Laptops
- **Algorithms**
- CNN (Convolutional Neural Network)
- **Software**
- Python
- Anaconda as IDE.

4.3.2. Convolutional Neural Network

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets can learn these filters/characteristics.

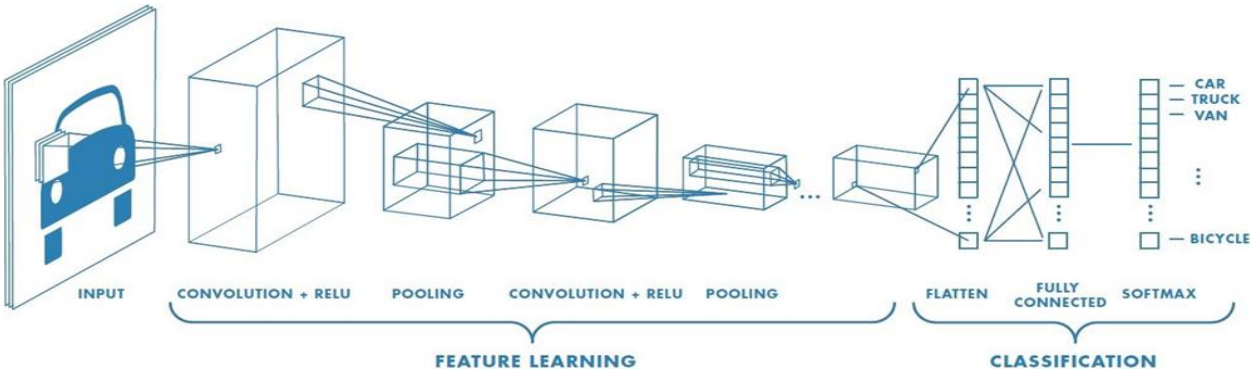

Figure 4. 2 CNN

A ConvNet can successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and the reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

4.3.3. OpenCV

OpenCV is a Python open-source library, which is used for computer vision in Artificial intelligence, Machine Learning, face recognition, etc. It is used for processing real-time image and video, as well as providing analytics, and machine learning capabilities. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms.

These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras find similar images from an image database, , etc.

4.4. Implementation

The implementation details of this project are discussed below:

4.5. Dataset

We used facial images data-sets which has different facial images of different age groups with different persons. Because this is a regression problem and dataset are based on images, we use Convolutional Neural Network for the prediction of the age group.Firstly, we will upload the dataset on the Anaconda compiler. If we talk about the data-set, it has approximately 5000 facial images folders as mentioned in the figure 5:

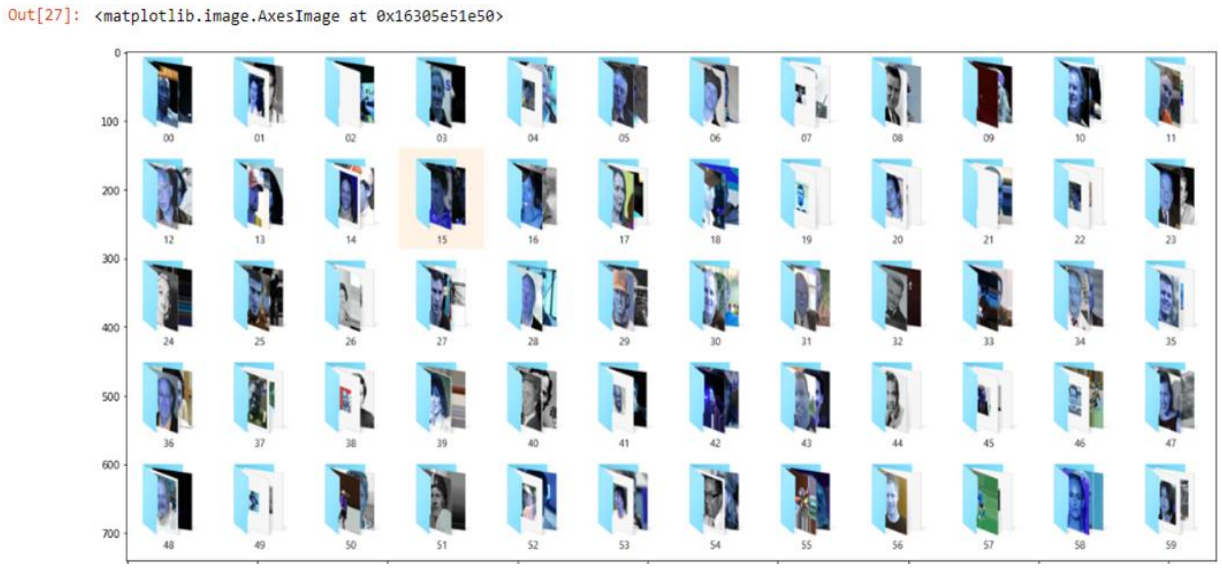

Figure 4. 3 Dataset Images Folders

We divide our data-set into two parts, one is for training which is 70% and other is for validation which is 30%. These images are the images of different persons of different age groups placed randomly in folders as shown in below figures is also given above.

4.5.1. Dataset Preprocessing

The first step of preprocessing data is to create folder specific for the specific age group facial images, so we have created a folder 1 to 100 for this purpose and every folder is then named as certain age folder as mentioned in the figure below:

Figure 4. 4 Folder Creation

Now in the next step after specifying the folders, we move the corresponding facial age group image to its specific folder, as mentioned in the figures below:

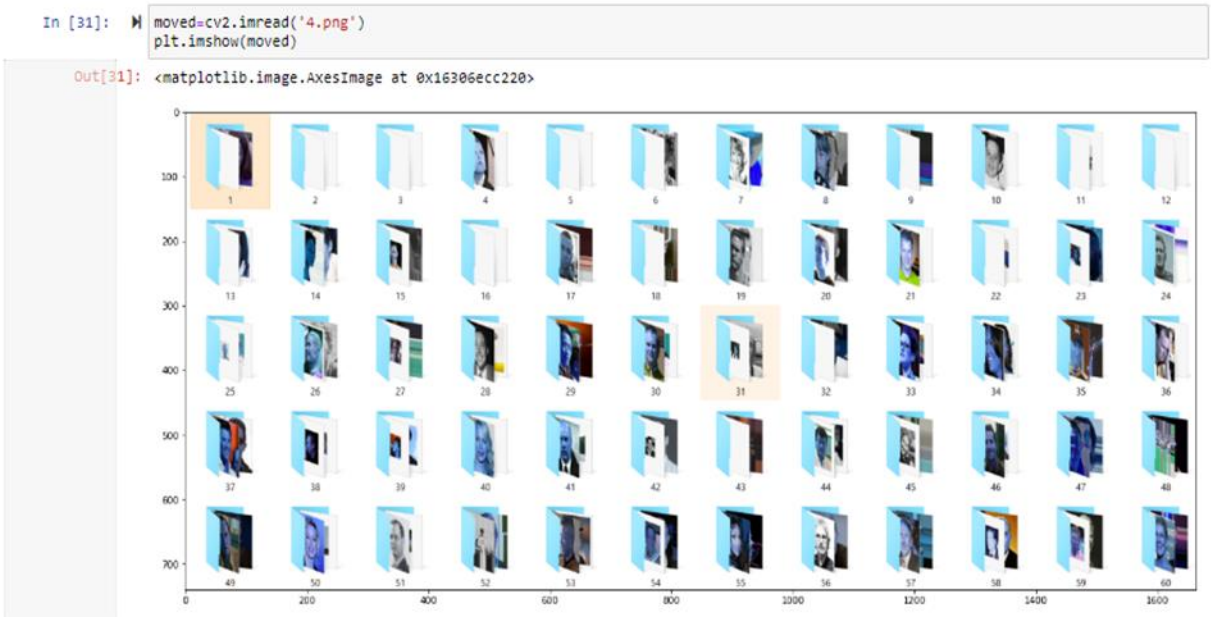

Figure 4. 5 Update Folders

4.5.2. Dataset Cleaning

The first step of cleaning dataset is to remove the noisy images present in the dataset. Noisy images are those images in dataset which are not needed and their matrix shaped are too small in size like

1x1 etc. In dataset these images are not needed, these images are we can defective, so we have to remove them as shown in figure below some examples of noisy images:

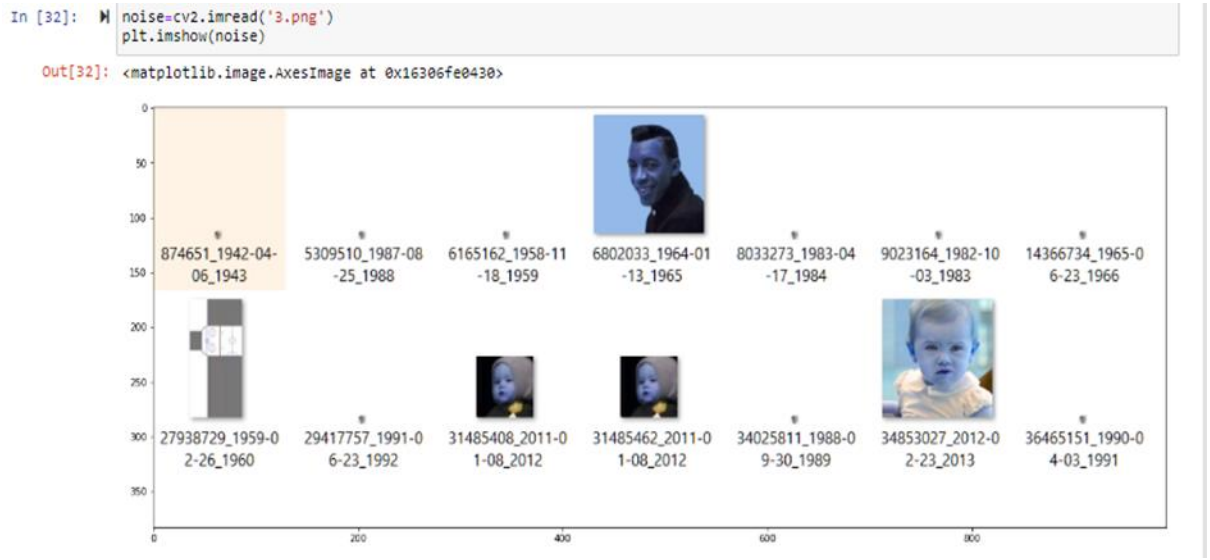

Figure 4. 6 Noisy Images

So, for the removal of the noisy images, we simply send these images to other location/ folder from the dataset as mentioned in below figures:

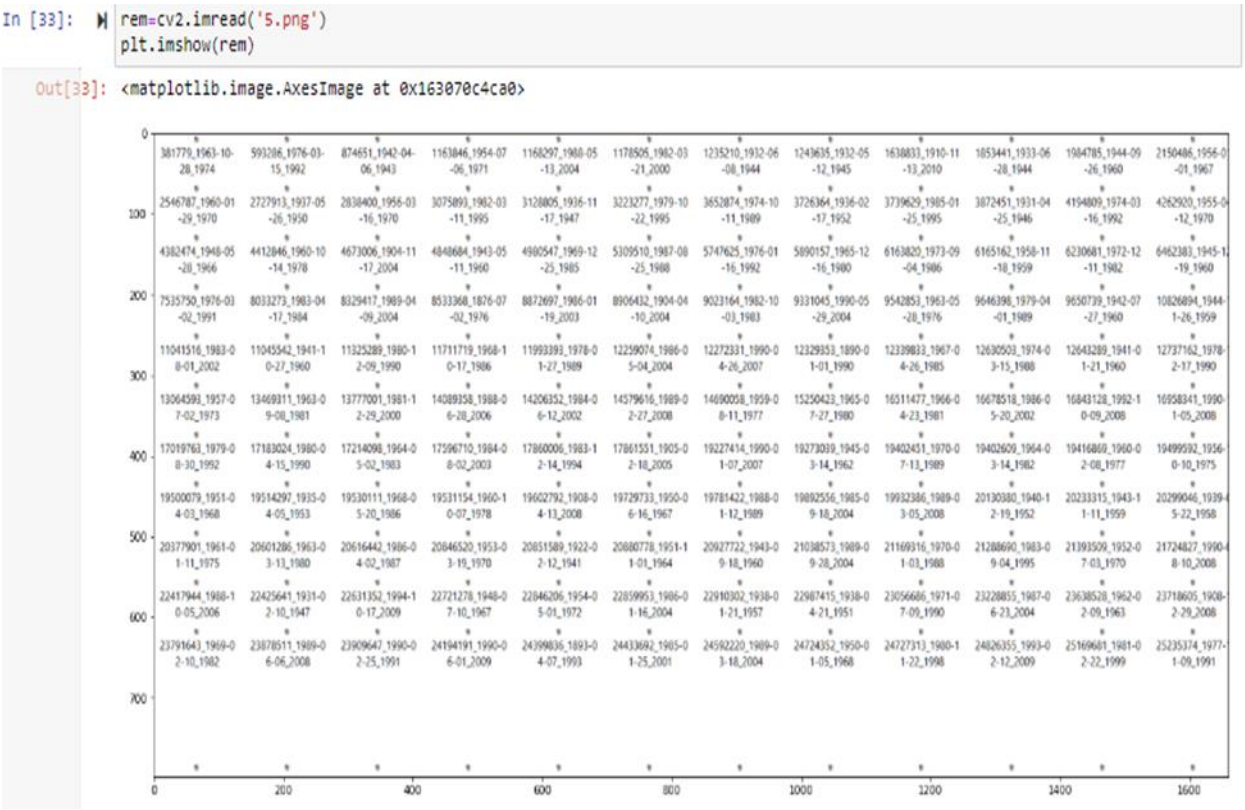

Figure 4. 7 Noisy Images Folder

4.5.3. Convolutional Neural Network

After cleaning process is done, we apply CNN on training data and validation data. First it extracts features after that it goes through different layers of CNN and at the end it shows the results on both datasets. The features extraction is mentioned in the figure below;

```
import seaborn as sns
import warnings
from tqdm.notebook import tqdm
warnings.filterwarnings('ignore')
%matplotlib inline

import tensorflow as tf
from keras.preprocessing.image import load_img
from keras.models import Sequential, Model
from keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D, Input
from PIL import Image
from tqdm.notebook import tqdm
```

In [36]:
```
image_dir = Path('D:/age data/New folder/age')
```

In [37]:
```
filepaths = pd.Series(list(image_dir.glob(r'**/*.jpg')), name='Filepath').astype(str)
ages = pd.Series(filepaths.apply(lambda x: os.path.split(os.path.split(x)[0])[1]), name='Age').astype(np.int)

images = pd.concat([filepaths, ages], axis=1).sample(frac=1.0, random_state=1).reset_index(drop=True)
```

In [39]:
```
def extract_features(images):
    features = []
    for image in tqdm(images):
        img = load_img(image, grayscale=True)
        img = img.resize((128, 128), Image.ANTIALIAS)
        img = np.array(img)
        features.append(img)

    features = np.array(features)
    features = features.reshape(len(features), 128, 128, 1)
    return features
```

In [40]:
```
X = extract_features(images['Filepath'])
```

Figure 4. 8 Features Extraction

The below image shows different layers of CNN by which the dataset is go through which includes convolutional layer. This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter. Second layer is Pooling Layer,

The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and

independently operates on each feature map. Depending upon method used, there are several types of Pooling operations. It basically summarizes the features generated by a convolution layer. The next layer is fully connected layer, The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.

```
In [45]: input_shape = (128, 128, 1)

In [46]: inputs = Input((input_shape))
         # convolutional layers
         conv_1 = Conv2D(32, kernel_size=(3, 3), activation='relu') (inputs)
         maxp_1 = MaxPooling2D(pool_size=(2, 2)) (conv_1)
         conv_2 = Conv2D(64, kernel_size=(3, 3), activation='relu') (maxp_1)
         maxp_2 = MaxPooling2D(pool_size=(2, 2)) (conv_2)
         conv_3 = Conv2D(128, kernel_size=(3, 3), activation='relu') (maxp_2)
         maxp_3 = MaxPooling2D(pool_size=(2, 2)) (conv_3)
         conv_4 = Conv2D(256, kernel_size=(3, 3), activation='relu') (maxp_3)
         maxp_4 = MaxPooling2D(pool_size=(2, 2)) (conv_4)

         flatten = Flatten() (maxp_4)

         # fully connected layers
         #dense_1 = Dense(256, activation='relu') (flatten)
         dense_2 = Dense(256, activation='relu') (flatten)

         #dropout_1 = Dropout(0.3) (dense_1)
         dropout_2 = Dropout(0.3) (dense_2)

         #output_1 = Dense(1, activation='sigmoid', name='gender_out') (dropout_1)
         output_2 = Dense(1, activation='relu', name='age_out') (dropout_2)

         model = Model(inputs=[inputs], outputs=[ output_2])

         model.compile(loss=[ 'mae'], optimizer='adam', metrics=['accuracy'])

In [ ]: history = model.fit(X=X,y=y_age, batch_size=32, epochs=10,validation_split=0.2 )

        Epoch 1/10
```

Figure 4. 9 Layers of CNN

Finally, one of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network.

4.5.4. Result

Table 4.1 shows the results after applying CNN. We set the batch size to 32 and run 20 epochs and as mentioned below in the table that from start the loss is heigh but at the end the loss became very less and same happens with val_loss which is at the start very heigh but at the end it became low.

Epochs	Loss	Val_loss
1.	247.123	206.244
2.	193.404	166.464
3.	166.890	161.316
4.	148.715	171.201
5.	135.285	142.626
6.	124.055	129.024
7.	113.353	130.450
8.	101.071	130.710
9.	90.652	135.734
10.	78.410	139.968
11.	67.954	137.437
12.	59.966	137.885
13.	53.638	139.262
14.	48.504	143.181
15.	44.283	143.979
16.	41.985	142.473
17.	38.804	150.482
18.	37.838	145.480
19.	35.001	146.798
20.	33.239	145.159

Table 3. 1 Results

Chapter 5 Testing and Analysis

5.1 Testing Procedures

Every system needs its proper testing when it's been developed and it is very important to perform proper testing to check whether the system is working fine or not. So, we applied different testing cases to our system and we emphases on the system's datasets increasing and decreasing the number of datasets, and increasing or decreasing the training amount of the system to check its performance which is discussed in the chapter below.

5.2 System Test Scenarios and Analysis

We perform different tests on the system and the results are shown below. The first test is shown in figure 5.1and which shows the accurate response of the system in the real-time frame.

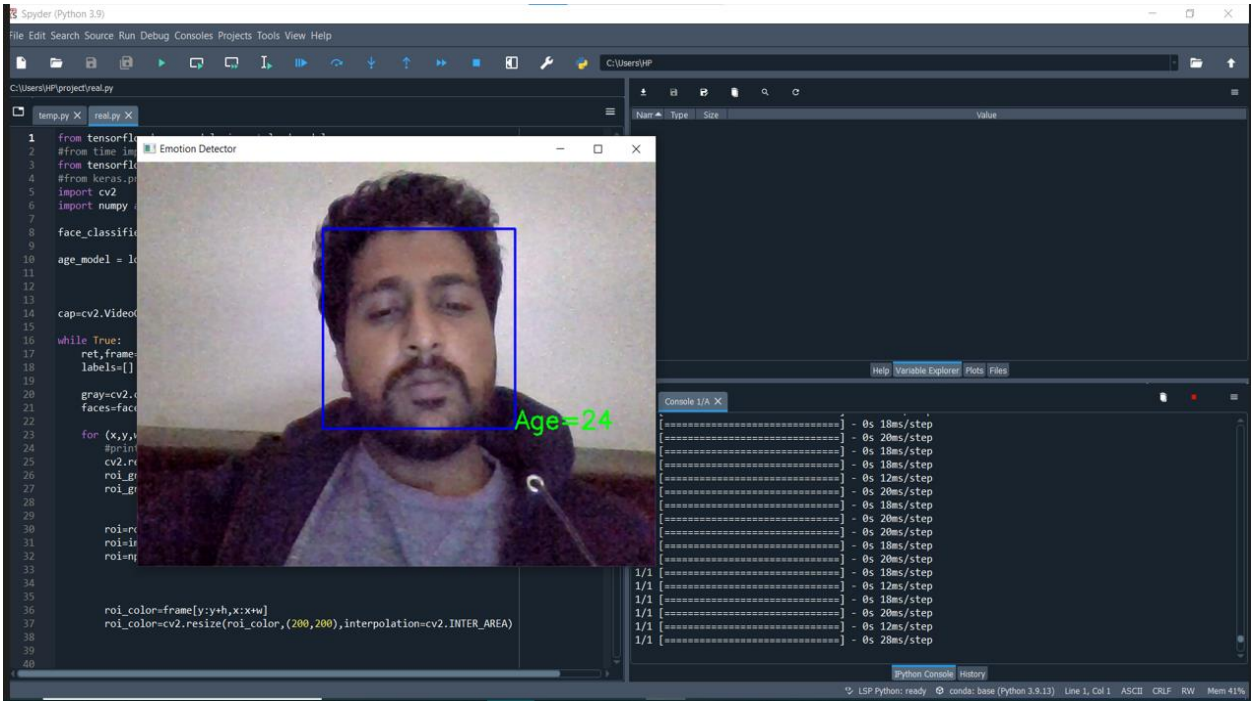

Figure 5. 1 Test Case 1

The second test is shown below which also predicts the accurate age group on the live test from a dataset image of a person as shown in figure 5.2. The figure shows that the actual age of the person in the image is 66 and the system predicted it is 59 which is close and highly accurate.

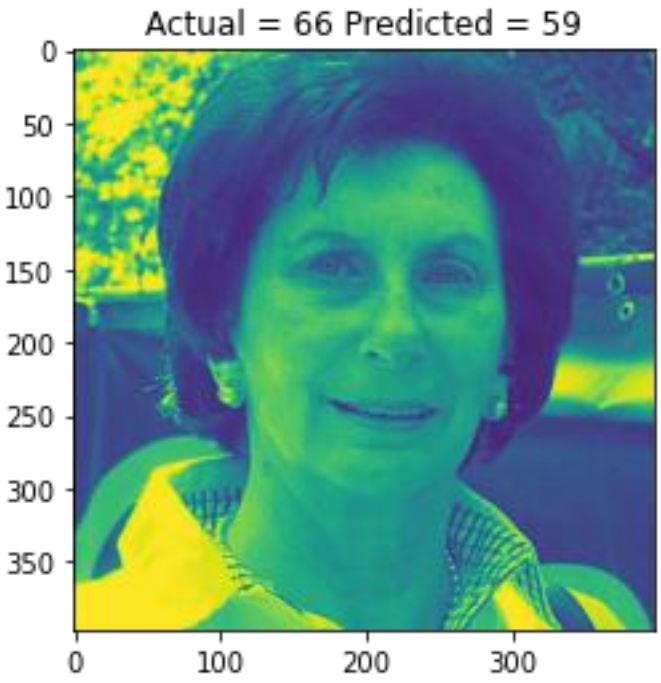

Figure 5. 2 Test Case 2

The third test scenario is performed on the young person's image data and the system predicts accurate results as shown below in figure 5.3. It shows that the actual age of a person in a figure is 41 and the system predicted it as 36 which is also close and mostly accurate.

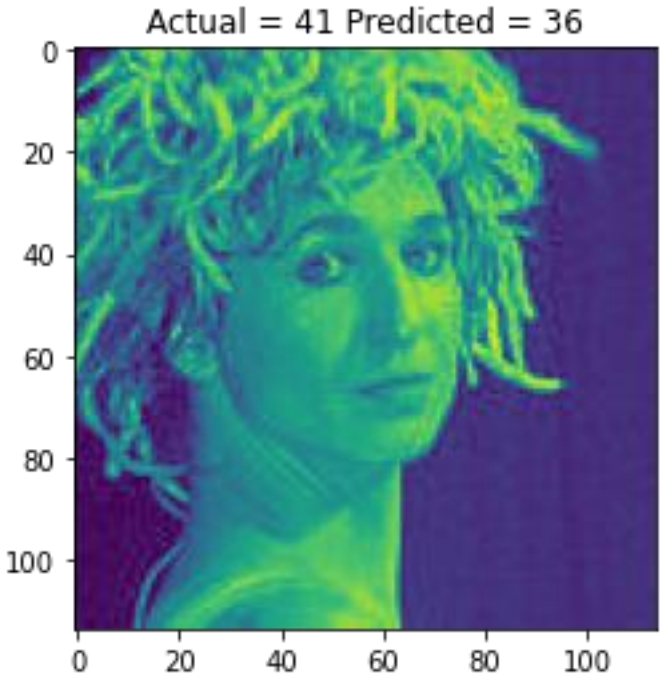

Figure 5. 3 Test Case 3

In the next image that we take from our dataset, we perform a test and it also predicts accurately. The figure 5.4 shows that the actual age of the person in the image is 21 and the system predicts it as 24 which is also close and highly accurate.

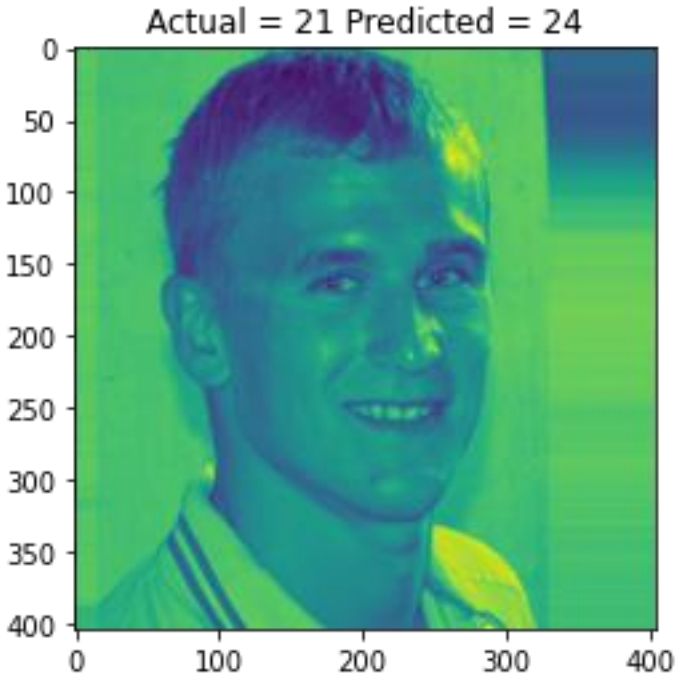

Figure 5. 4 Test Case 4

5.3 Summary

This project is developed to build such a system in which we can detect the human age in a real-time camera. We use Deep learning with computer vision technologies which enables us to predict the human age in real-time from a live camera. The image dataset used for this system is larger in numbers which helps us to get more accuracy from the system. The system is stable and performs more accurately and is also user-friendly.

Chapter 6 Conclusion and Future Work

6.1 Conclusion

The system we build to detect human age in real-time is stable and accurate. The algorithms used for this system are working accurately. Deep learning and computer vision serve the purpose of this project. The facial image dataset used in this system is bigger in numbers. We perform many tests on the system and the system predicts accurate results for these tests which shows that the system is accurate and reliable.

6.2 Improvements and Future Work

With the technological advancement in the field of AI and the improvements made in the computer vision field, there is a lot of work will be done in the field of human age detection in real-time. As more and more algorithms are developed the system can improve and as a result, we get more options in our hands for improvements. We can increase the training of data, or we can use different algos to build a more accurate, and advanced system.

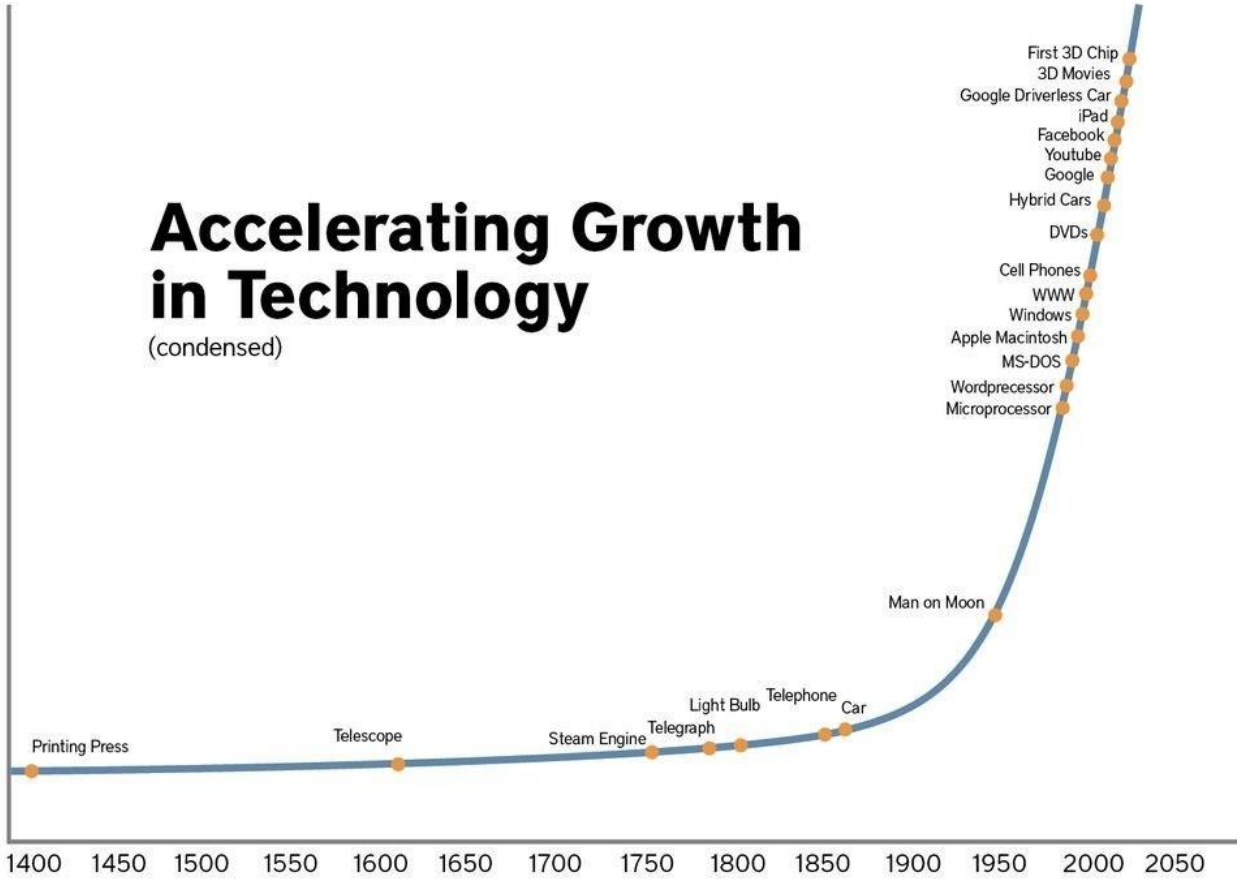

Figure 6. 1 Advancement in Technology

References

[1] A. Ekmekji, "Convolutional Neural Networks for Human age Classification".

[2] Amirali Abdolrashidi, "Age and Gender Prediction From Face Images Using Convolutional Neural Newtork".

[3] "Deeply Learned Classifiers for Human age Predictions of unclassified faces".

[4] T. Hassner, "Age and Gender Classification using Convolutional Neural Networks," The Open University of Israel.

[5] Singla, "Human age Detection Using Deep Learning," Data Science Blogathon, july 27, 2021.

[6] d. Silva, "Human age Classification – A Proposed System," Department of Computer Science and Technology, Canada.