MAR ATHANASIUS COLLEGE OF ENGINEERING
(Affiliated to APJ Abdul Kalam Technological University, TVM)
KOTHAMANGALAM

Department of Computer Applications

Main Project Report

Age and Gender Detection

Done by

Shirin Safwana S

Reg No: MAC20MCA-2019

Under the guidance of
Prof. Merin Joy M

2020-2022

MAR ATHANASIUS COLLEGE OF ENGINEERING
(Affiliated to APJ Abdul Kalam Technological University, TVM)
KOTHAMANGALAM

CERTIFICATE

Age and Gender Detection

Certified that this is the bonafide record of project work done by

Shirin Safwana S
Reg No: MAC20MCA-2019

During the academic year 2020-2022, in partial fulfilment of requirements for award of the degree,

Master of Computer Applications
of
APJ Abdul Kalam Technological University
Thiruvananthapuram

Faculty Guide
Prof. Merin Joy M

Head of the Department
Prof. Biju Skaria

Project Coordinator
Prof. Biju Skaria

External Examiner

ACKNOWLEDGEMENT

First and foremost, I thank God Almighty for his divine grace and blessings in making all this possible. May he continue to lead me in the years to come.

I am also grateful to Prof. Biju Skaria, Head of the Department, Department of Computer Applications and also our project coordinator ,for his valuable guidance as well as timely advice which helped me a lot during preparation of the project.

I would like to express my special gratitude and thanks to my Main project guide Prof. Merin Joy M, Assistant Professor, Department of Computer Applications for her guidanceand constant supervision as well as for providing necessary information regarding the Main project & also for her support.

I profusely thank other Professors in the department and all other staffs of MACE, for their guidance and inspirations throughout my course of study. No words can express my humble gratitude to my beloved parents who have been guiding me in all walks of my journey. My thanks and appreciations also go to my friends and people who have willingly helped me out with their abilities.

ABSTRACT

Automatic prediction of age and gender from face images has drawn a lot of attention recently, due it is wide applications in various facial analysis problems. To build a gender and age detector that can guess the gender and approximate age of the person in a picture using deep learning .The predicted gender may be either male or female and predicted age may be one of the following ranges (0-2),(4-6),(8-12),(15-20),(25-32) ,(38-43),(48-53),(60-100).

The proposed architecture in the paper is Convolutional Neural Network. But after a brief study, I felt that the latest CNN architecture ResNet50 is more useful and advantagable to this task. It has several advantages over other architectures like Lenet-5, VGG , AlexNet etc. . Age and gender information are very important for various real world applications, such as social understanding, biometrics, identity verification, video surveillance , human-computer interaction, electronic customer, crowd behaviour analysis, online advertisement, item recommendation, and many more.

In this project I'm using 2 datasets. The dataset we using for age is Audience Benchmark dataset. It has 12196 images of 2284 subjects. It has the features age in classes mentioned above and gender (f or m). The Audience set consists of images automatically uploaded to Flickr from smartphone devices. Because these images were uploaded without prior manual filtering, as is typically the case on media webpages or social websites, viewing conditions in these images are highly unconstrained, reflecting many of the real-world challenges of faces appearing in Internet images. Audience images therefore capture extreme variations in head pose, lightning conditions quality, and more. The dataset using for gender classification is of cropped images of male and female . It is split into training and validation directory. Training contains 23,000 images of each class and validation directory contains 5,500 images of each class.

Dataset Url: https://talhassner.github.io/home/projects/Adience/Adience-data.html
https://www.kaggle.com/datasets/cashutosh/gender-classification-dataset

LIST OF TABLES

LIST OF FIGURES

CONTENTS

1. INTRODUCTION

Automatic prediction of age and gender from face images has drawn a lot of attention recently, due it is wide applications in various facial analysis problems. To build a gender and age detector that can guess the gender and approximate age of the person in a picture using deep learning .The predicted gender may be either male or female and predicted age may be one of the following ranges (0-2),(4-6),(8-12),(15-20),(25-32) ,(38-43),(48-53),(60-100).

The dataset we using for age is Audience Benchmark dataset. It has 12596 images of 2284 subjects. It has the features age in classes mentioned above and gender (f or m). The Audience set consists of images automatically uploaded to Flickr from smartphone devices. The dataset using for gender classification is of cropped images of male and female .It is called gender classification dataset. It is split into training and validation directory. Training contains 23,000 images of each class and validation directory contains 5,500 images of each class.

The proposed architecture in the paper is Convolutional Neural Network. But after a brief study, I felt that the latest CNN architecture ResNet50 is more useful and advantagable to this task. The task of gender and age detection just from an image is not an easy task even for us humans because it is totally based on looks and sometimes it is not easy to guess it. People of the same age can look very different from what we can guess. We can create solutions better than this using ResNet50 CNN (convolutional neural networks) which has emerged as the most preferred model for computer vision tasks.

2. SUPPORTING LITERATURE

2.1. Literature Review

Paper 1:- Aryan Saxena, Prabhangad Singh , Shailendra Narayan Singh(2021) , IEEE , Age and gender detection using deep learning.

The paramount objective of this paper is to build a gender and age detector that can guess the gender and approximate age of the face of an individual in a picture using deep learning on the audience dataset. This technology can be used to our benefit and look at the huge spectrum where it can be implemented: ranging from security services, CCTV surveillance and policing to dating applications, matrimonial sites.

And it also discusses the great success of deep learning models in various computer vision problems in the past decade, the more recent works on age and gender predictions are mostly shifted toward deep neural networks based models. In this work, we propose a deep learning framework to jointly predict the age and gender from face images. Given the intuition that some local regions of the face have more clear signals about the age and gender of an individual (such as beard and mustache for male, and wrinkles around eyes and mouth for age). Computer Vision enables the computers and enables them to look at , figure out and identify digital images and videos as a human would. It involves acquiring , processing, analyzing and understanding digital images to extract data from the real world in order to generate symbolic or numerical information that it uses to make decisions on. This process includes different practices like object recognition ,tracking a video, motion estimation and image restoration . Convolutional Neural Network is a deep neural network widely used for purpose of image recognition and processing and NLP. Also known as ConvNet. CNN configuration generally consist of 4 layers. Convolution layer, ReLU layer, Pooling layer , Fully connected layer. Deep CNN is used here to classify the age and gender of the face of a person.

The tentative process that told in this paper is detection of facial area, classification into the gender(Male/Female), classification of age into one of the 9 age groups(0-2, 4-6, 8-12, 15-20, 21-24, 25-32, 38-43, 48-53, 60-100) and then display the resultant on the image using the facebox.

This paper suggests, the webcam facility also, if no argument has been passed for an image , then the model automatically prompts to get access of the webcam on the computer in order to do live face detection.

Paper 2:- Insha Rafique, Awais Hamid, Sheraz Naseer (2019) , IEEE , Age and gender detection using deep convolutional neural network .

This paper proposes to build a age and gender detector . The classification of age was done by calculation ration between different features of face like nose, eyes, mouth, chin etc. After localizing calculating their sizes and distances, ratio between them are calculated in order to predict age by using conventional methods. There are various application of Convolutional Neural Network (CNN) are present like human pose estimation, face parsing, facial key point detection, and speech recognition and action Classification. On unconstraint photo this is their first application according to our knowledge. The system we proposed works perfectly fine with experiments in classification for age and gender. Classification of age on dataset requires to differentiate between eight classes and for two genders.

The objective here is to make a system that would be efficient enough to predict age and gender of a person without breaching his security and other bypassing any other security. Such efficient systems are helpful in variety of ways in performing different activities. The main objective focuses on train a model which can predict age and gender in most efficient way. It is also aimed to use in age specific content access limitation by which system can detect age and gender and allows/deny user to access that content. Talking about the model, five network architecture layers are used in this model. Two of them are fully connected layers and three convolutional layers are used.

Using deep CNN, model is trained to an extent that accuracy of Age and Gender become 79% using HAAR cascading. Its accuracy could be increased more using more efficient algorithms and more precise architecture of CNN so that it could have been used more in different platforms.

Paper 3:- Mohammed Kamel Benkaddour , Sara Lahlali, Maroua Trabelsi (2020), IEEE, Human age and gender classification using convolutional neural network.

In this research, I analyzed the implementation of deep convolutionnal neural network for human age and gender prediction using CNN. During this study various design was developed for this task, age and gender classification is one of the key segments of research in the biometric as social applications with the goal that the future forecast and the information disclosure about the particular individual should be possible adequately.

The goal of this work is to develop a gender prediction and age estimation system based on convolutional neural networks for a face image or a real-time video. In this paper, three CNN network models were created with different architecture (the number of filters, the number of convolution layers...) validated on IMDB and WIKI dataset, the results obtained showed that CNN networks greatly improve the performance of the system as well as the accuracy of the recognition. Finally, as a perspective, an extension of this work can be envisaged by creating a face detection and recognition system based on CNNs as a feature extractor and the machine vector support as a classifier, another perspective would be the tests our approach on other facial databases showing strong variations in lighting and pose.

There are 3 methods for preprocessing step used in this work, face detection, cropping, and resizing. In face detection methods, here it uses the Haar-Cascade technique. It attempts to eliminate background and non-face areas and then to cut the region of the face. The proposed method aimed to automate a system for gender prediction and age estimation by using CNN and deep learning techniques, first, it build three models: CNN1, CNN2, CNN3. The CNNs models were trained for 1500 epochs, after every epoch the accuracy was calculated, which is the count of predictions where the predicted value is equal to the true value, it is typically expressed as a percentage. The input is passed through a pile of convolutional and maxpooling layer, the non-linear activation function (ReLu) was used, in output result , applied a sigmoid function , for all models , RMSpro was used as an optimizer.

2.2. Findings and Proposals

From the three papers I referred , It is clear that the paper discusses about basic CNN and deep CNN for the age and gender detection. Yes, CNN is best for image classification. But today there are several architectures for CNN like LeNet, AlexNet, VGGNet, ResNet etc. Also they have advantages and disadvantages as per new technological inventions. In these papers it is clear that gender classification is easier than age classification. Because gender classification has 2 classes, but age classification has several classes. There are more marked differences exists between genders than between many age groups. In gender classification it has 84% accuracy and for age classification it has 44.64% accuracy. From researches I understand that deeper networks is equal to more expressive and better performance. So I went for ResNet50 model. The main benefit of a very deep network is that it can represent very complex functions. It can also learn features at many different levels of abstraction, from edges (at the lower layers) to very complex features (at the deeper layers). ResNet-50 (Residual Networks) is a deep neural network that is used as a backbone for many computer vision applications like object detection, image segmentation, etc. Convolutional Neural Networks have a major disadvantage — 'Vanishing Gradient Problem'. During backpropagation, the value of gradient decreases significantly, thus hardly any change comes to weights. To overcome this, ResNet is used. It make use of **"**SKIP CONNECTION**".** The popular ResNet50 contained 49 convolution layers and 1 fully connected layer at the end of the network. Resnets are a kind of CNNs called Residual Networks they are very deep compared to Alexnet and VGG, and Resnet 50 refers to a 50 layers Resnet.

So I uses ResNet50 for my model and I separately build model for both age and gender classification. But to run ResNet50, it needs higher system configurations. So I use transfer learning with keras ie, a phenomenon which allows you to transfer what state-of-the-art machine learning models have learnt, and you use it for your custom problem. So in short, transfer learning allows us to reduce massive time and space complexity by using what other state-of-the-art models have learnt.

3. SYSTEM ANALYSIS

3.1. Analysis of Dataset

3.1.1. About the Dataset

I collect my datasets from few open source libraries, Kaggle and Git towards datascience.

❖ https://www.kaggle.com/datasets/cashutosh/gender-classification-dataset

❖ https://talhassner.github.io/home/projects/Adience/Adience-data.htm

The dataset used for gender classification is gender classification dataset which is taken from Kaggle.com . Training contains ~23,000 images of each class and validation directory contains ~5,500 images of each class. The data set is of cropped images of male and female. It has training and validation directory that contains two class folders ie, male and female. The data was collected from the different sources in the internet including scraping the web. The data mostly consist of the image from IMDB dataset. The IMDB dataset has almost 200k data which was carefully cleaned and seperated in the two different classes. After separating the data, the faces from the data were cropped and saved to different directory.

The dataset we are going to use to train our age model is the audience benchmark age dataset. This dataset contains various images in various real-world conditions with different lighting and noise levels .It contains 12596 images of 2284 subjects of different age group. This dataset is used for age classification. And it is a image dataset having training and validation folder containing 8 age groups , they are (0-2),(4-6),(8-12),(15-20),(25-32) ,(38-43),(48-53),(60-100). The Audience set consists of images automatically uploaded to Flickr from smartphone devices. Because these images were uploaded without prior manual filtering, as is typically the case on media webpages or social websites, viewing conditions in these images are highly unconstrained, reflecting many of the real-world challenges of faces appearing in Internet images. Audience images therefore capture extreme variations in head pose, lightning conditions quality, and more. The data included in this collection is intended to be as true as possible to the challenges of real-world imaging conditions. In particular, it attempts to capture all the variations in appearance, noise, pose, lighting and more, that can be expected of images taken without careful preparation or posing. And also I added some images(about 300) from google after preprocessing step and manually add it to the dataset.

3.1.2. Explore the Dataset

The gender classification dataset has train and valid folder. In these directories there are male and female classes(sub folders) that contains cropped images. Training folder has 23243 female images and 23766 male images. Validation folder has 5841 female images and 5808 male images.

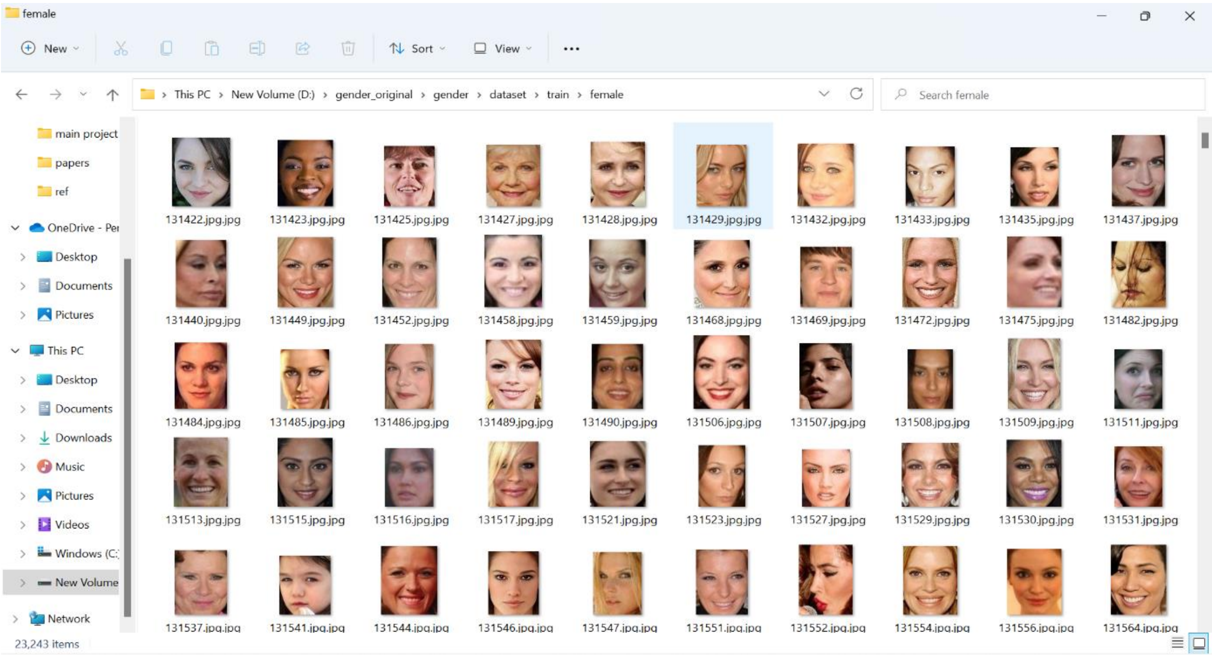


Fig 3.1. Images of Female

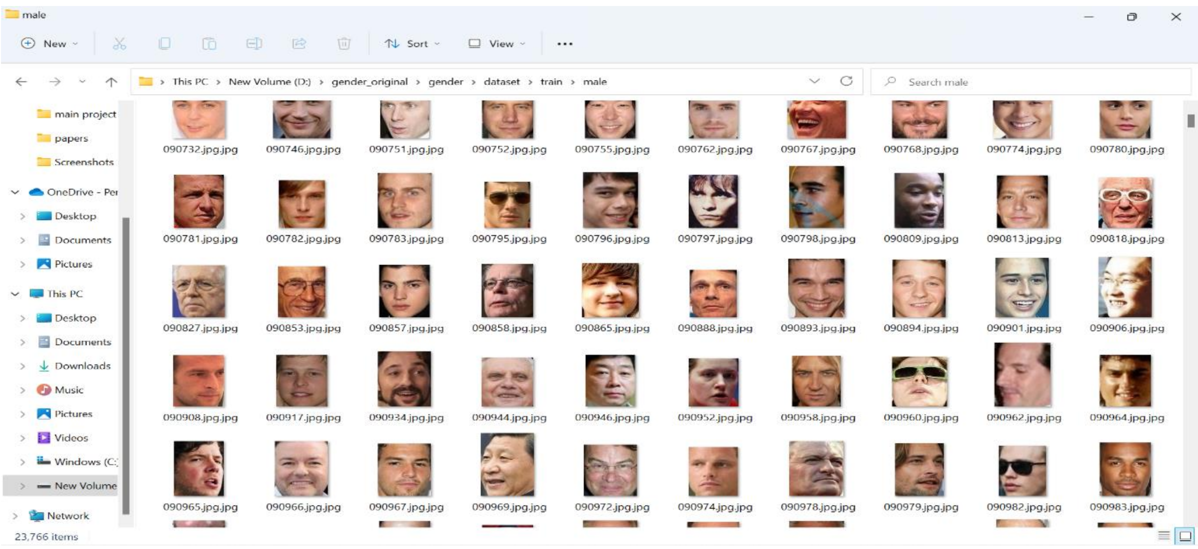


Fig 3.2. Images of Male

The audience benchmark age dataset has also train and valid folder and that contains 8 sub folders(classes).Train folder has 10196 images and valid folder has 2400 images.

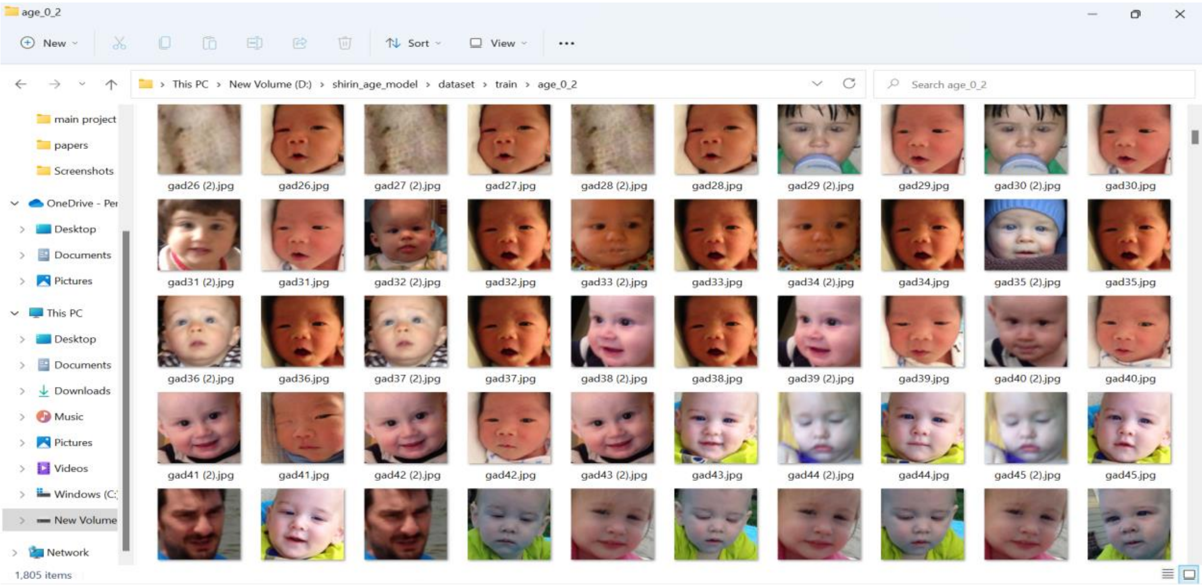


Fig 3.3. Images of 0-2 .

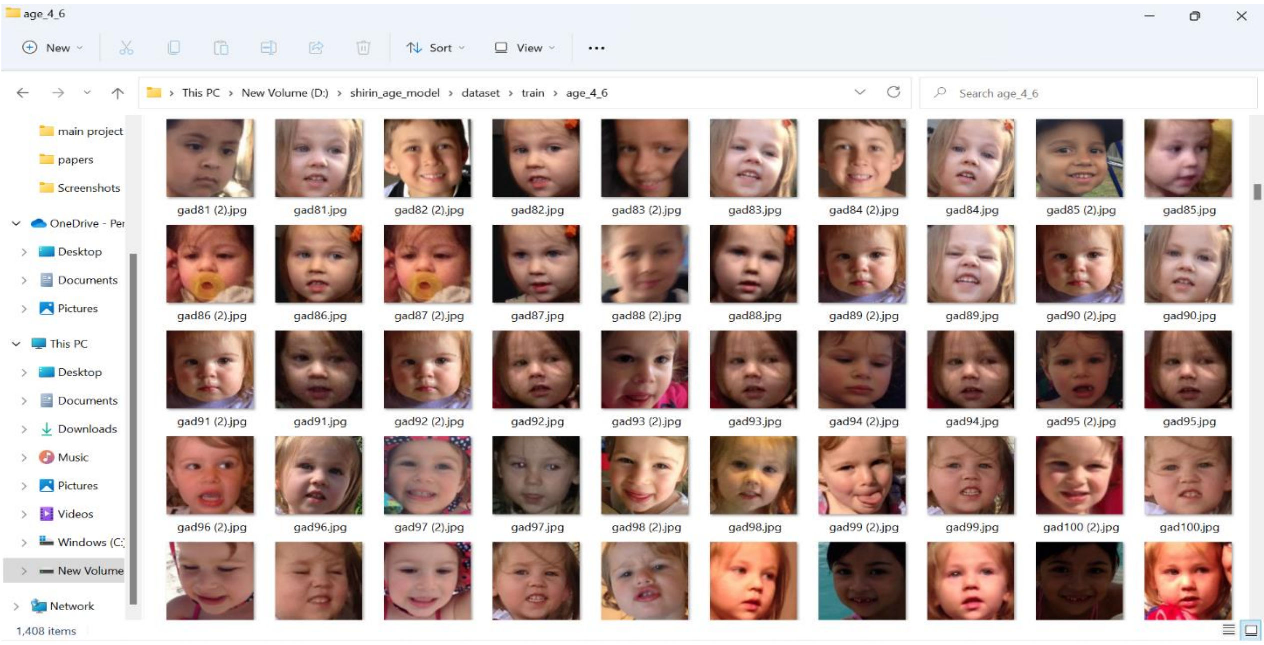


Fig 3.4. Images of 4-6.

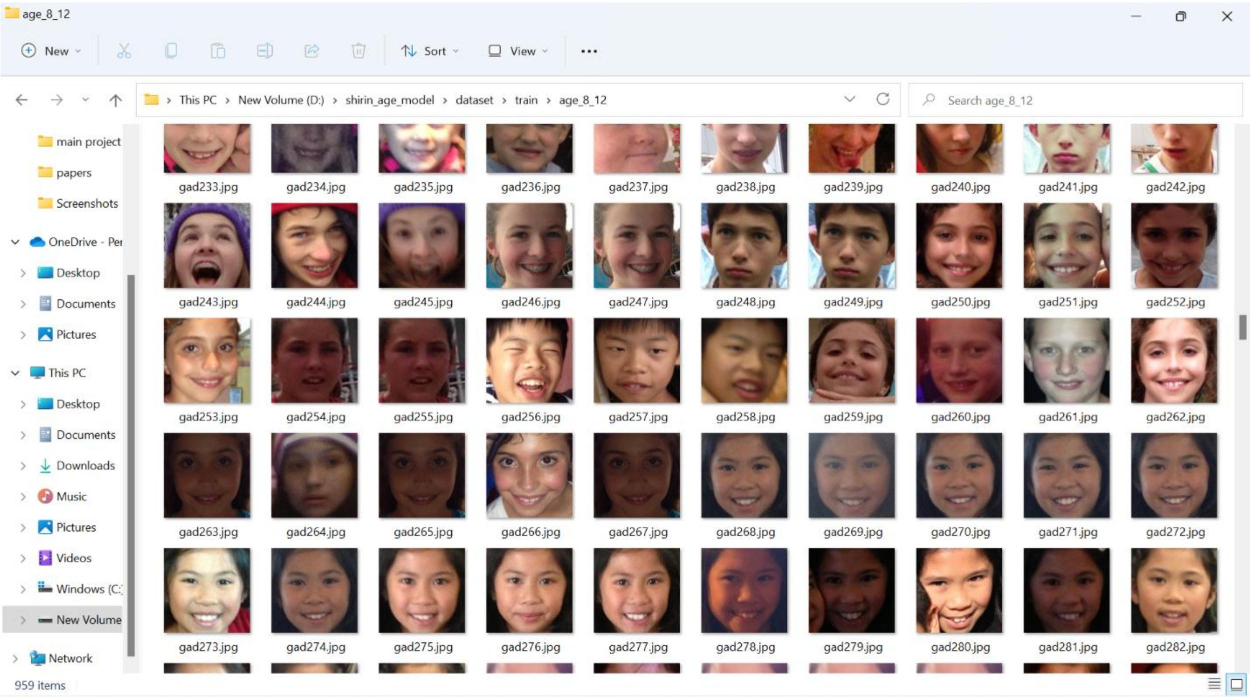


Fig 3.5. Images of 8-12.

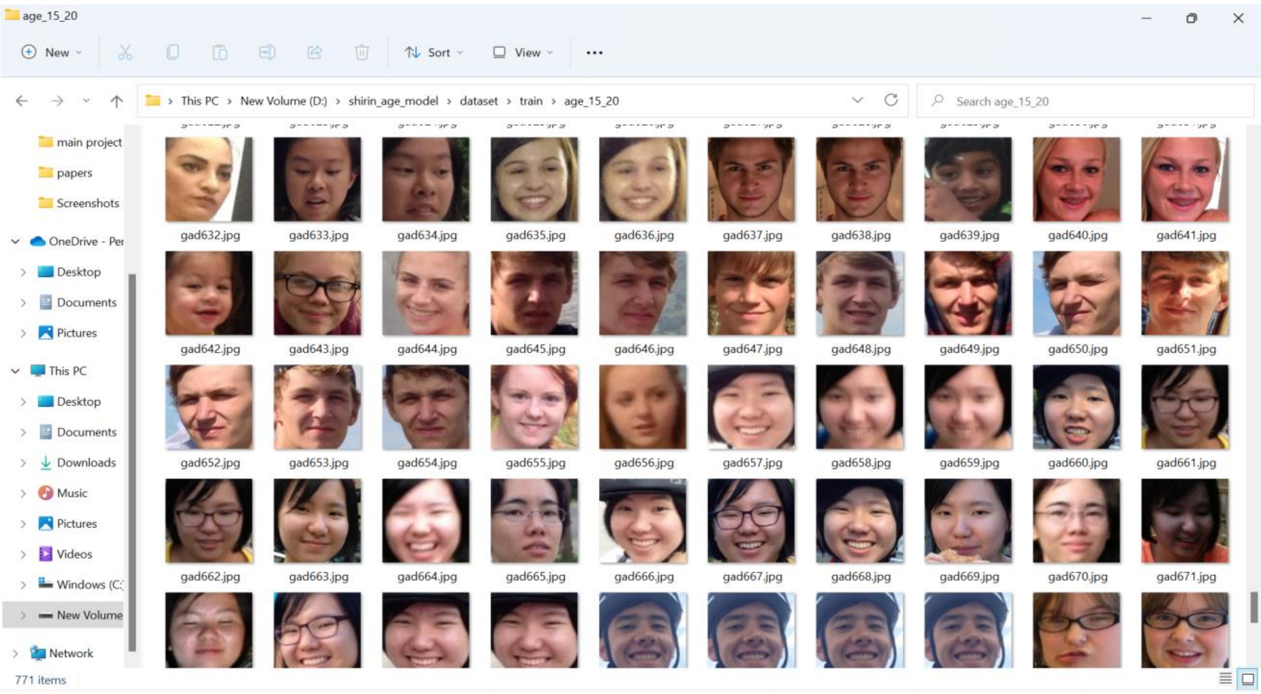


Fig 3.6. Images of 15-20.

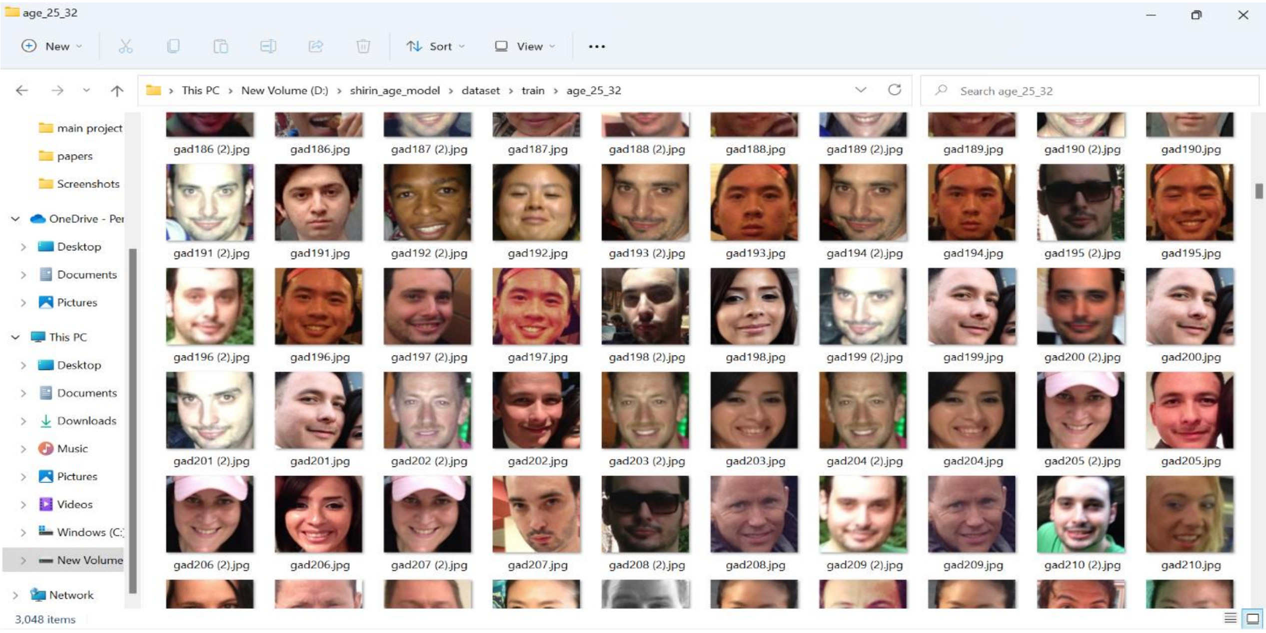


Fig 3.7 Images of 25-32.

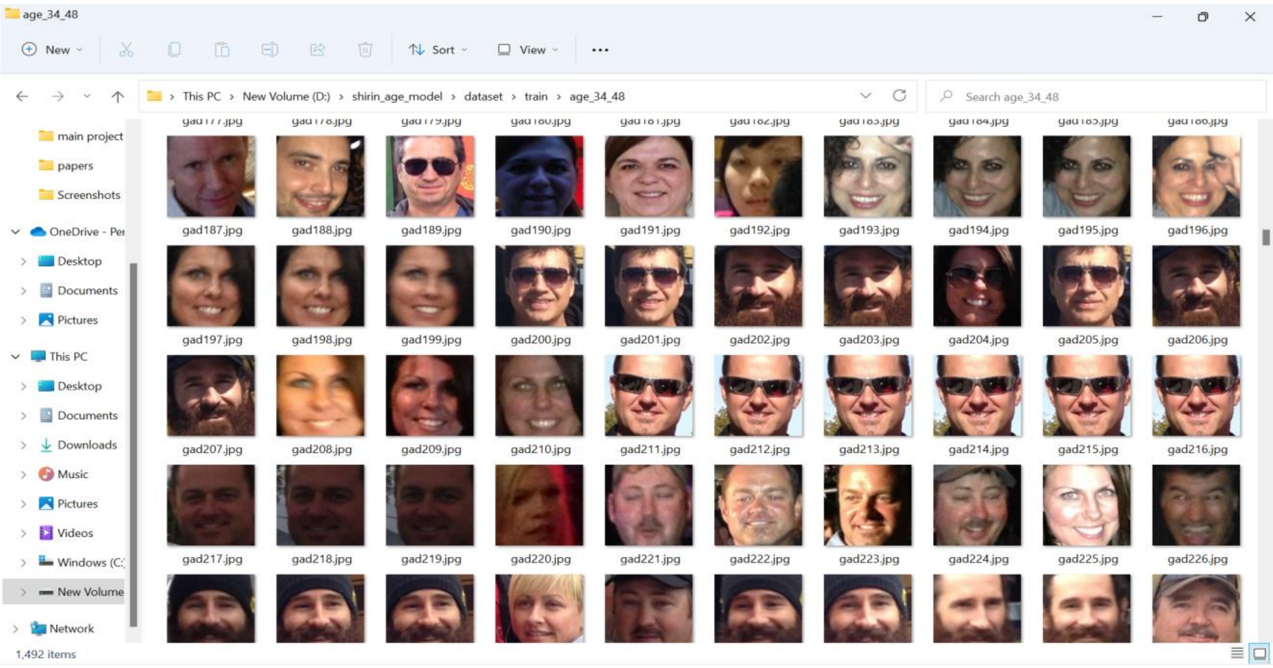


Fig 3.8. Images of 34-43.

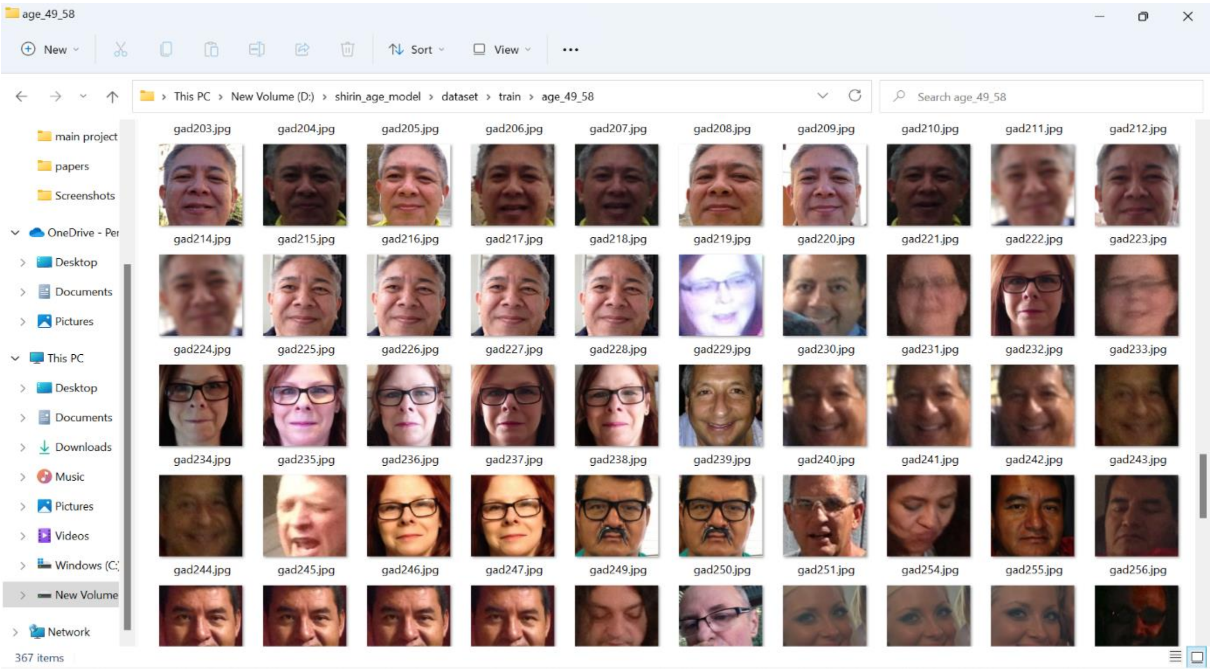


Fig 3.9. Images of 48-53.

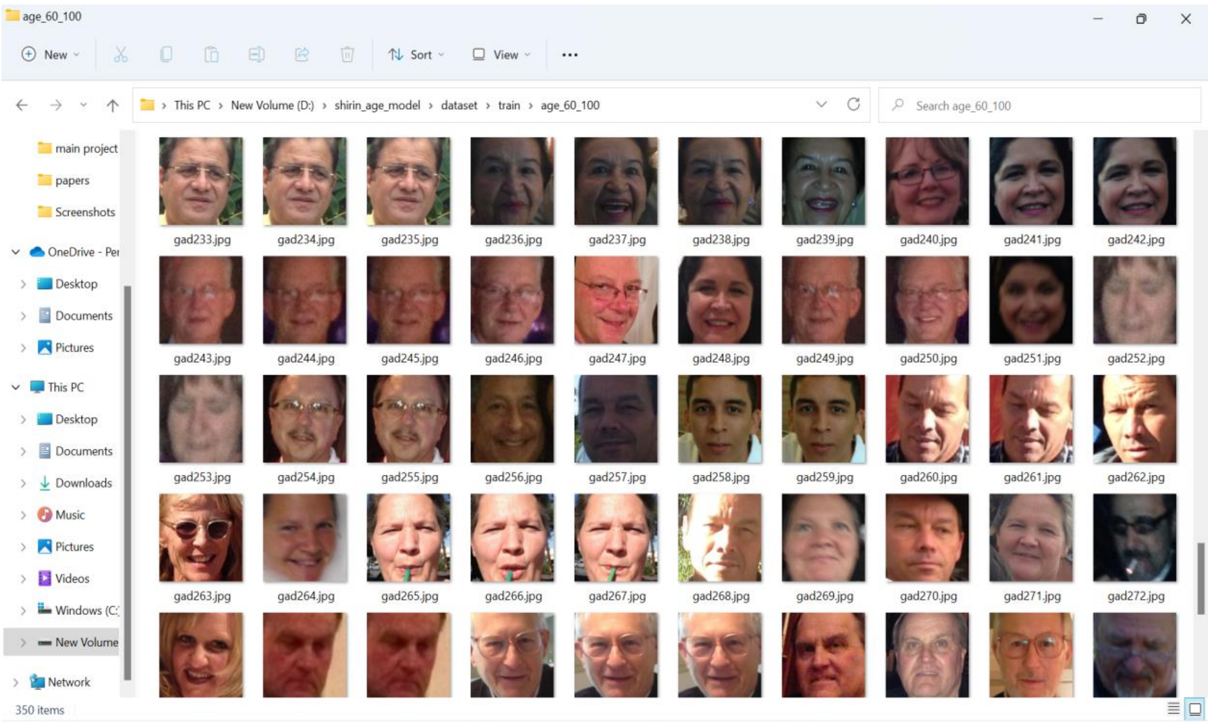


Fig 3.10. Images of 60-100.

3.2. Data Preprocessing

3.2.1. Data Cleaning

The preprocessing step is used for audience benchmark age dataset because we have to detect and extract only faces from the image. So it is easier for image classification.

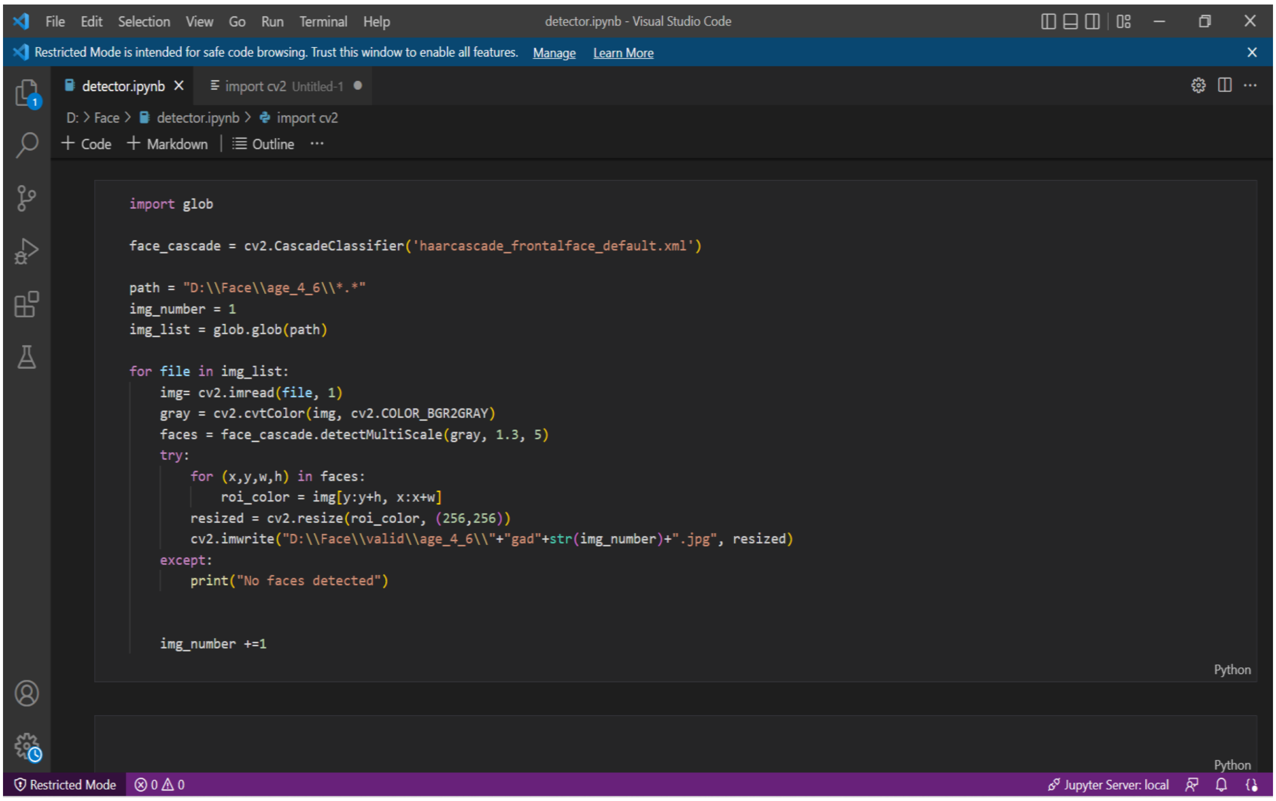


Fig 3.11: Preprocessing code

3.2.2. Analysis of Feature Variables

Both datasets are image dataset. Considering the features, it has age and gender . Features of image include properties like corners, edges, regions of interest points, ridges, etc.

3.2.3. Analysis of Class Variables

Classes are sometimes called as targets/labels or categories. Class of the dataset is the category to which the input will be classified to. That means the final result of an application. In my work, age has 8 classes such as (0-2), (4-6), (8-12), (15-20), (25-32), (38-43) ,(48-53),(60-100) and gender has 2 classes male and female.

3.3. Data Visualization

Data visualization is the graphical representation of information and data in a pictorial or graphical format (Example: charts, graphs, and maps). Data visualization tools provide an accessible way to see and understand trends, patterns in data, and outliers. Data visualization tools and technologies are essential to analysing massive amounts of informationand making data-driven decisions. The concept of using pictures is to understand data that hasbeen used for centuries. General types of data visualization are Charts, Tables, Graphs, Maps, and Dashboards.

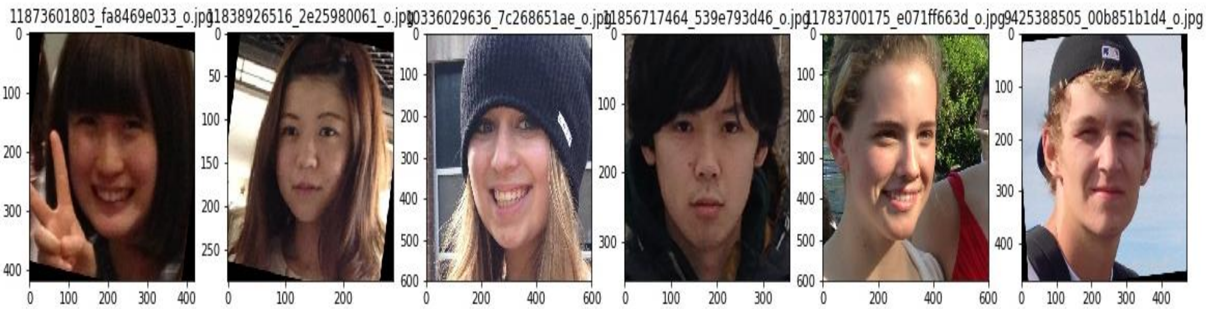


Fig 3.12: Visualization of image

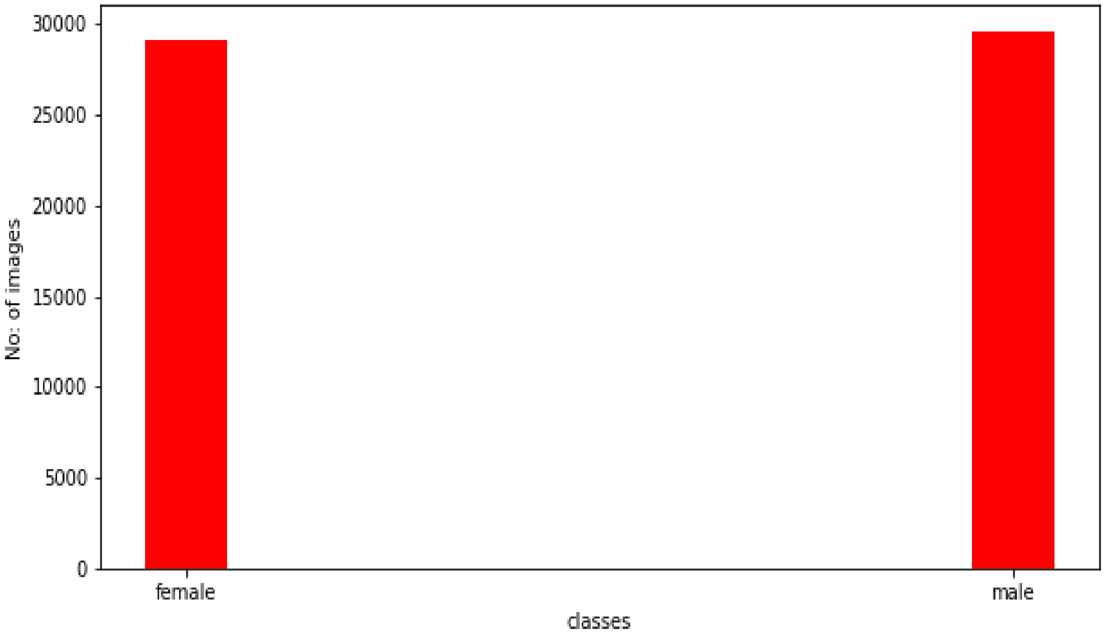


Fig 3.13: Visualization of Gender Classification Dataset

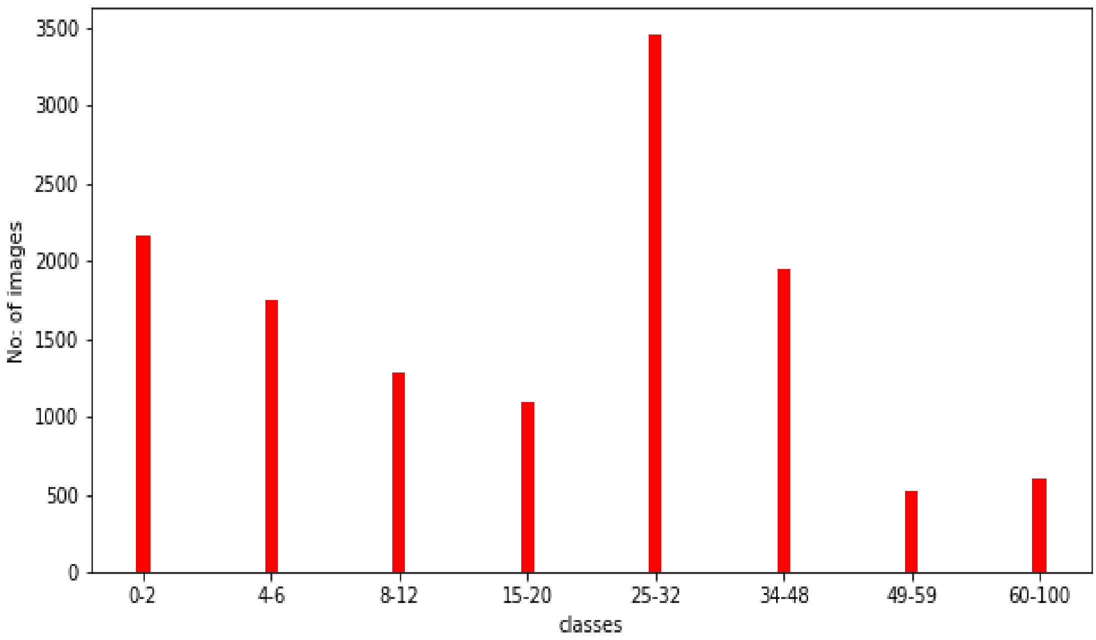
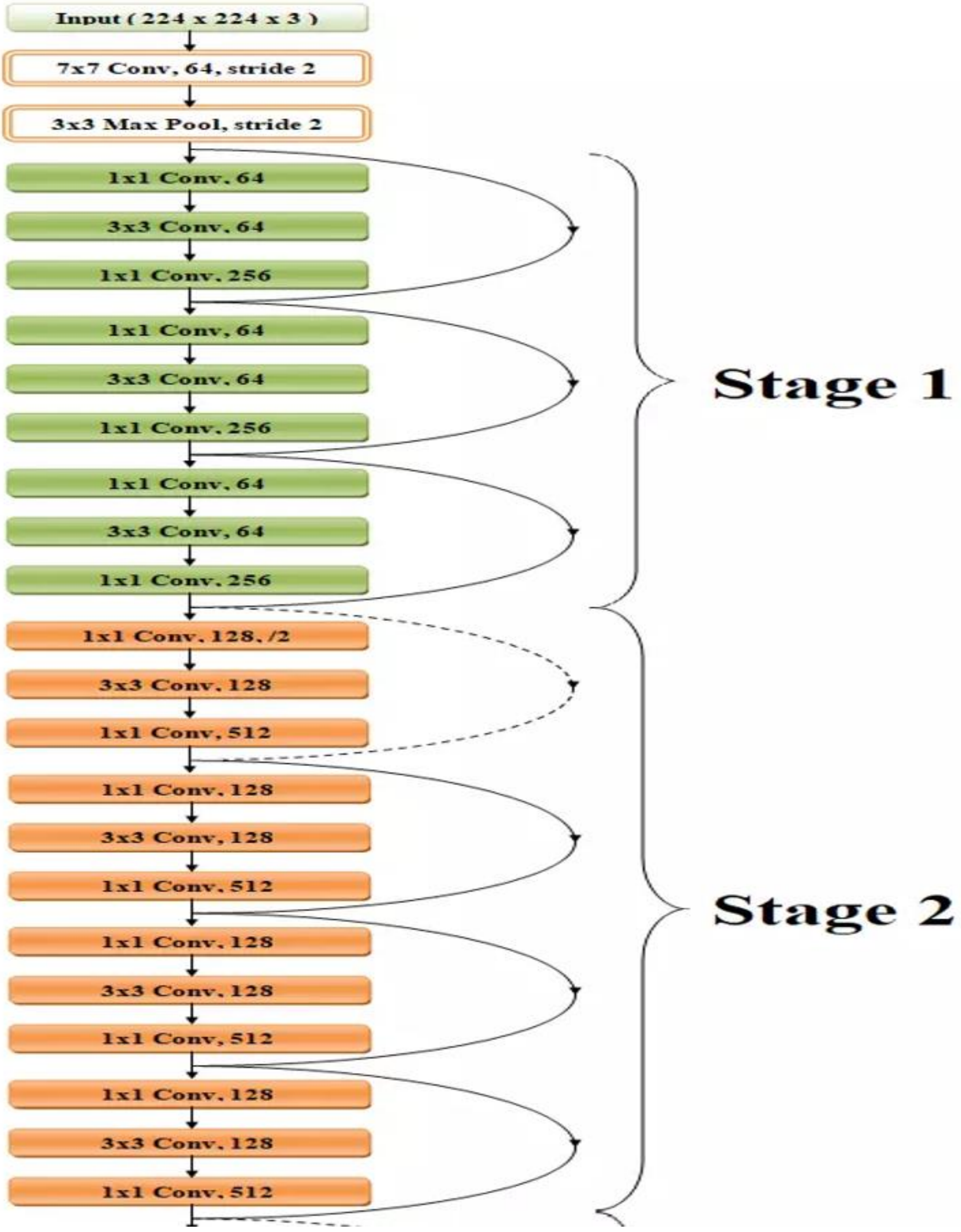


Fig 3.14: Visualization of Age Classification Dataset

3.4. Analysis of Architecture

3.4.1. Block Diagram



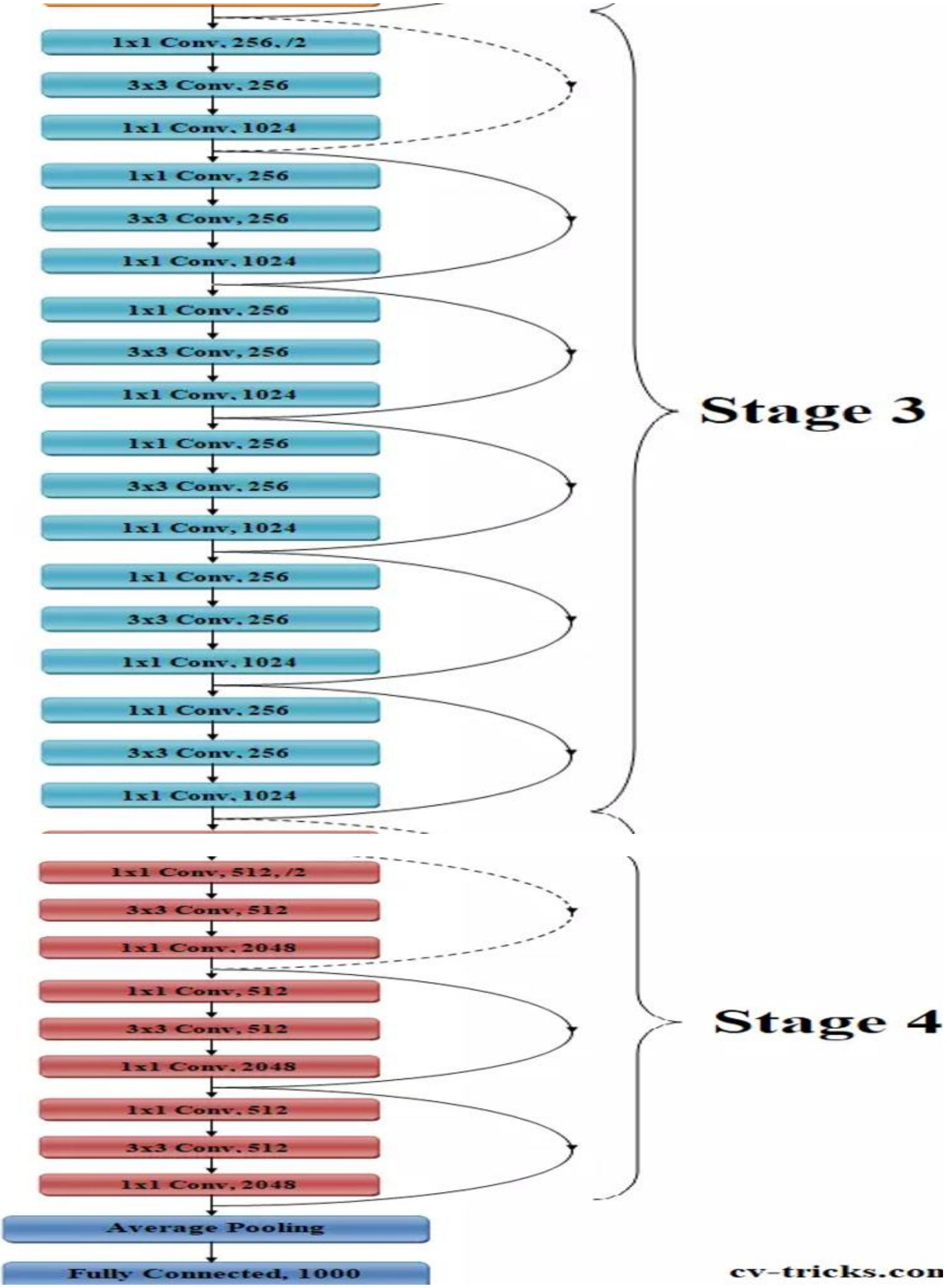
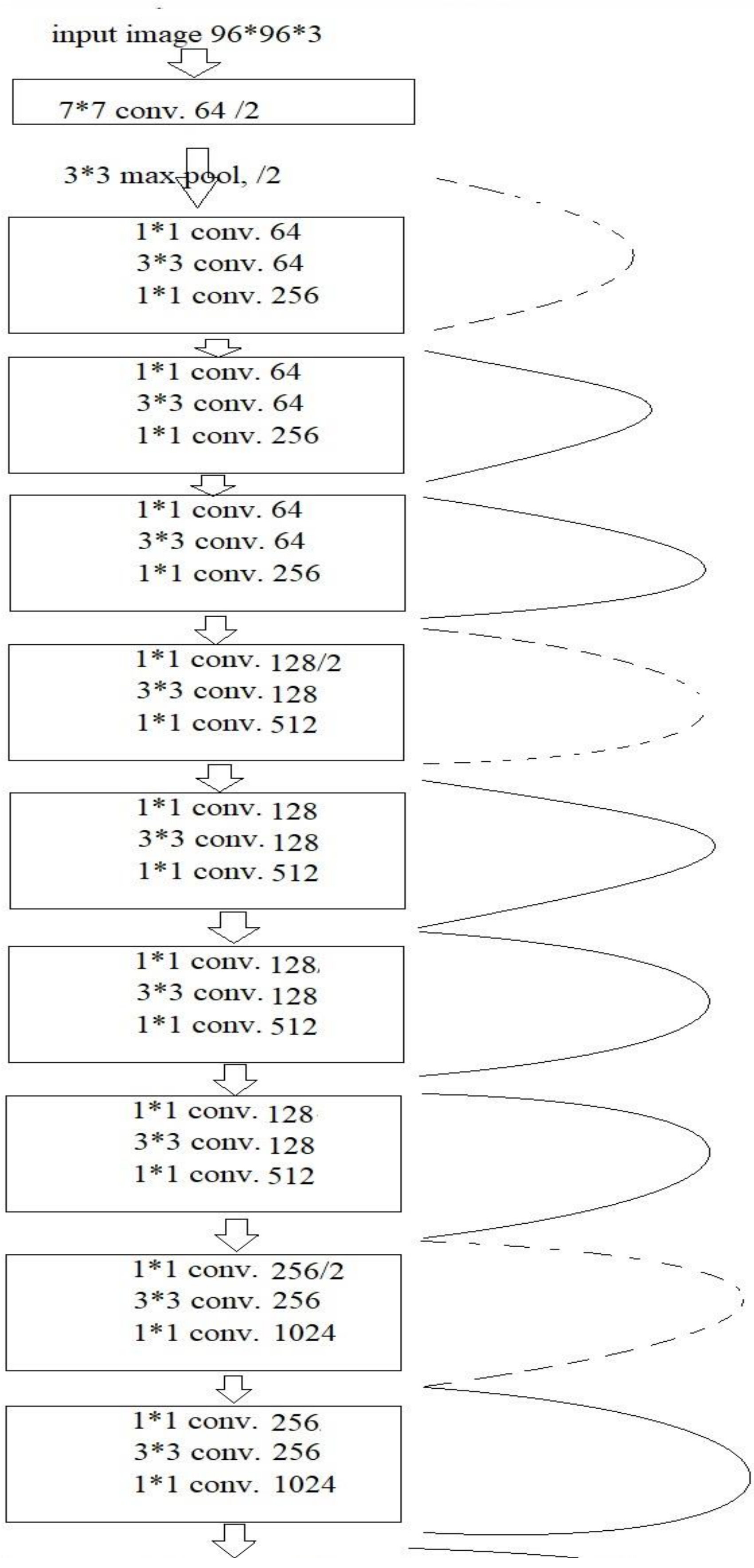


Fig3.15: Architecture diagram of ResNet50


input image 96*96*3
7*7 conv. 64 /2
3*3 max-pool, /2
1*1 conv. 64
3*3 conv. 64
1*1 conv. 256
1*1 conv. 64
3*3 conv. 64
1*1 conv. 256
1*1 conv. 64
3*3 conv. 64
1*1 conv. 256
1*1 conv. 128/2
3*3 conv. 128
1*1 conv. 512
1*1 conv. 128
3*3 conv. 128
1*1 conv. 512
1*1 conv. 128
3*3 conv. 128
1*1 conv. 512
1*1 conv. 128
3*3 conv. 128
1*1 conv. 512
1*1 conv. 256/2
3*3 conv. 256
1*1 conv. 1024
1*1 conv. 256
3*3 conv. 256
1*1 conv. 1024

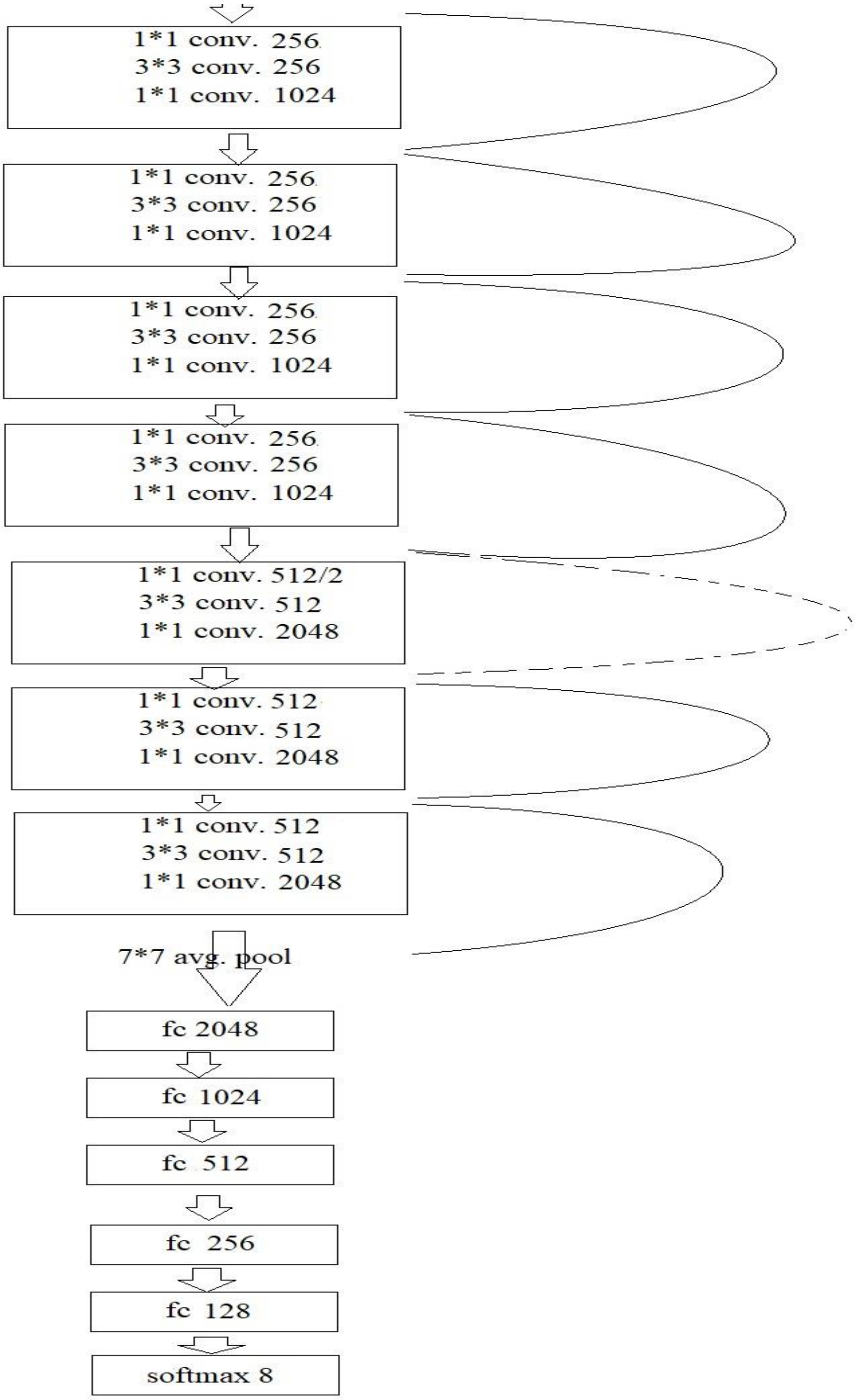


Fig 3.16: Block diagram of ResNet50 Age Model

ResNet is one of the most powerful deep neural networks which has achieved fantabulous performance results in the ILSVRC 2015 classification challenge. ResNet has achieved excellent generalization performance on other recognition tasks and won the first place on ImageNet detection, ImageNet localization, COCO detection and COCO segmentation in ILSVRC and COCO 2015 competitions. There are many variants of ResNet architecture i.e. same concept but with a different number of layers. We have ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-110, ResNet-152, ResNet-164, ResNet-1202 etc. The name ResNet followed by a two or more digit number simply implies the ResNet architecture with a certain number of neural network layers.

Deep Residual Network is almost similar to the networks which have convolution, pooling, activation and fully-connected layers stacked one over the other. The only construction to the simple network to make it a residual network is the identity connection between the layers.

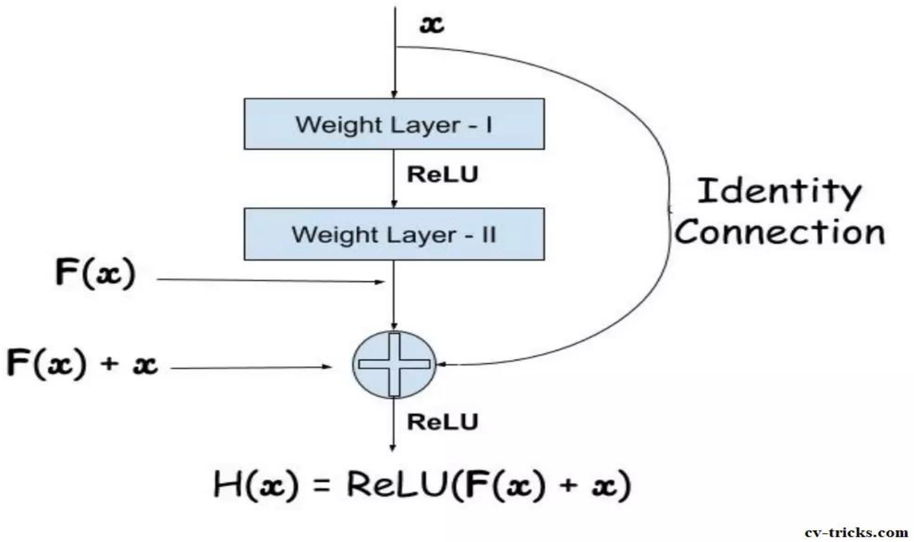


Fig 3.17: A residual block of deep residual network

The architecture of ResNet50 has ***4 stages*** as shown in the diagram below**.** The network can take the input image having height, width as multiples of 32 and 3 as channel width. For the sake of explanation, we will consider the input size as 96 x 96 x 3. Every ResNet architecture performs the initial convolution and max-pooling using 7×7 and 3×3 kernel sizes respectively. Afterward, Stage 1 of the network starts and it has 3 Residual blocks containing 3 layers each. The size of kernels used to perform the convolution operation in all 3 layers of the block of stage 1 are 64, 64 and 128 respectively. The curved arrows refer to the identity connection. The dashed connected arrow represents that the convolution operation in the Residual Block is performed with stride 2, hence, the size of input will be reduced to half in terms of height and width but the channel width will be doubled. As we progress from one stage to another, the channel width is doubled and the size of the input is reduced to half.

For deeper networks like ResNet50, ResNet152, etc, bottleneck design is used. For each residual function F**,** 3 layers are stacked one over the other. The three layers are 1×1, 3×3, 1×1 convolutions. The 1×1 convolution layers are responsible for reducing and then restoring the dimensions. The 3×3 layer is left as a bottleneck with smaller input/output dimensions.

Finally, the network has an Average Pooling layer followed by a fully connected layer having 1000 neurons (ImageNet class output).But our model 2048 neurons then 1024 neurons then 512 neurons and last a softmax layer of 8 classes. For gender classification softmax layer of 2 classes.

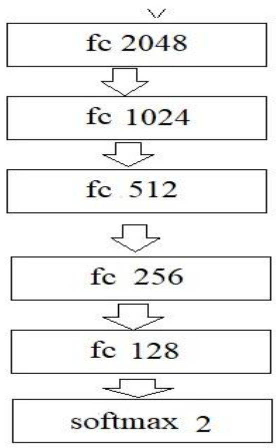


Fig 3.18: Block Diagram Remaining Portion of Gender model

Key Features of ResNet

- ResNet uses Batch Normalization at its core. The Batch Normalization adjusts the input layer to increase the performance of the network. The problem of covariate shift is mitigated.
- ResNet makes use of the Identity Connection, which helps to protect the network from vanishing gradient problem.
- Deep Residual Network uses bottleneck residual block design to increase the performance of the network.

Skip Connection

In ResNet architecture, a “shortcut” or a “skip connection” allows the gradient to be directly backpropagated to earlier layers:

-

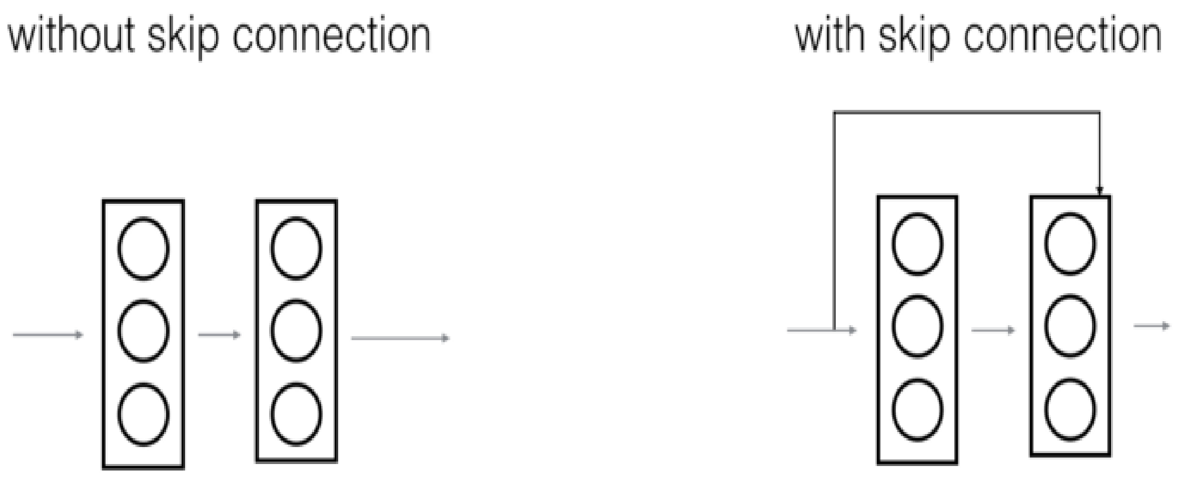


Fig 3.19: Skip Connection

The image on the top shows the “main path” through the network. The image on the bottom adds a shortcut to the main path. By stacking these ResNet blocks on top of each other, you can form a very deep network. There are two main types of blocks are used in a ResNet, depending mainly on whether the input/output dimensions are the same or different.

3.4.2. Diagrams and Details of Each Layer

1. **Identity Block -** The identity block is the standard block used in ResNets and corresponds to the case where the input activation has the same dimension as the output activation.

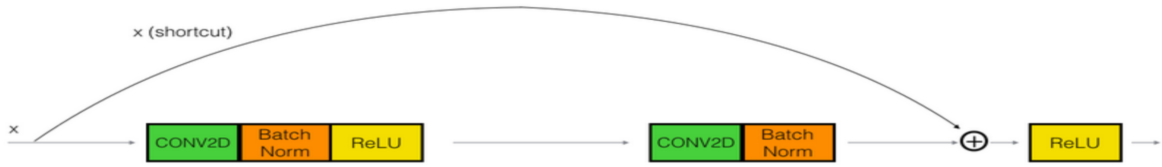


Fig 3.20: Identity block

2. **Convolutional Block –** We can use this type of block when the input and output dimensions don't match up. The difference with the identity block is that there is a CONV layer in the shortcut path.

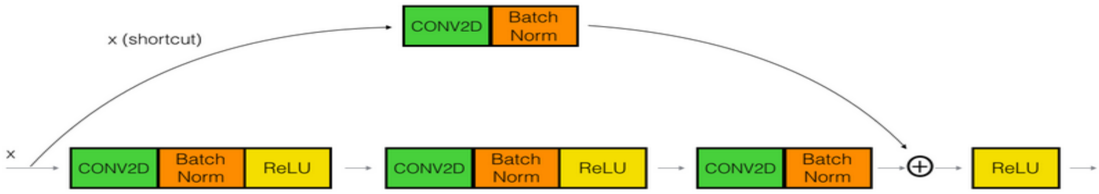


Fig 3.21: Convolution Block

1. 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 MxM. 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 (MxM).

The output is termed as the Feature map which gives us information about the image suchas the corners and edges. Later, this feature map is fed to other layers to learn several other featuresof the input image.

2. Pooling Layer

In most cases, a Convolutional Layer is followed by a 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.

In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually servesas a bridge between the Convolutional Layer and the FC Layer.

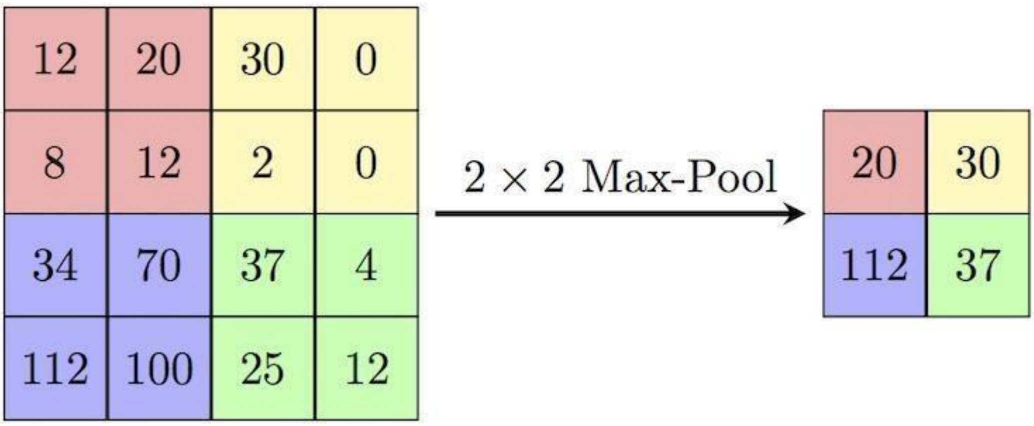


Fig 3.22: Process in max pooling layer

3. Fully Connected Layer

The Fully Connected (FC) layer consists of the weights and biases along with the neuronsand 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 this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place.

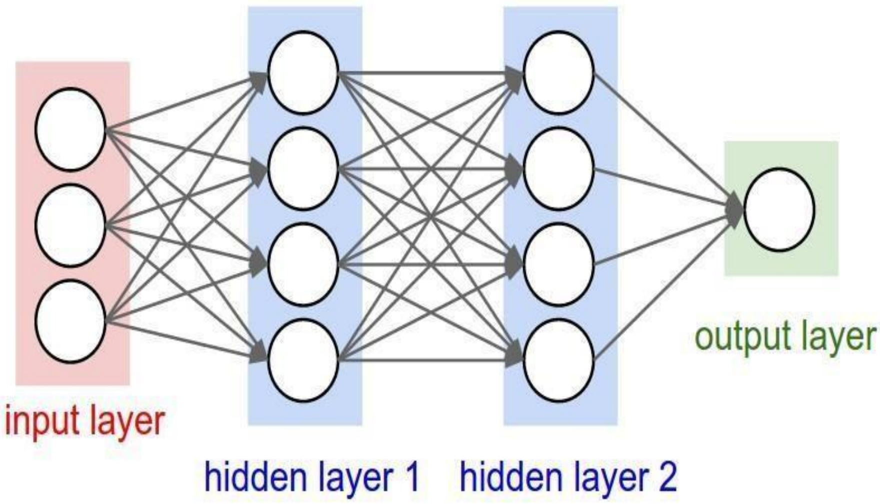
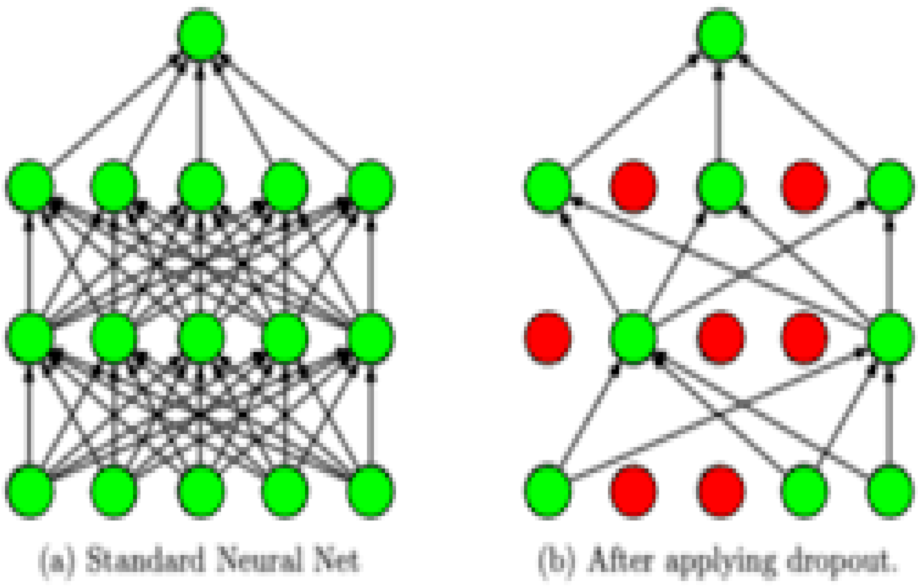


Fig 3.23 : Fully Connected Layer

4. Dropout

Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model's performance when used on a new data.

To overcome this problem, a dropout layer is utilized wherein a few neurons are dropped from theneural network during training process resulting in reduced size of the model. On passing a dropoutof 0.3, 30% of the nodes are dropped out randomly from the neural network.



5. Activation Functions

Finally, one of the most important parameters of the CNN model is the activationfunction. 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 ofthe model should fire in the forward direction and which ones should not at the end of the network.

It adds non-linearity to the network. There are several commonly used activation functionssuch as the ReLU, Softmax, tanH and the Sigmoid functions. Each of these functions have a specific usage. For a binary classification CNN model, sigmoid and softmax functions are preferred an for a multi-class classification, generally softmax us used.

Before learning each layer, there are two parameters which are important in the working ofconvolutional neural network layers, stride and padding

Stride

Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and soon. Stride is a component of convolutional neural networks, or neural networks tuned for the compression of images and video data. Stride is a parameter of the neural network's filter that modifies the amount of movement over the image or video. For example, if a neural network's stride is set to 1, the filter will move one pixel, or unit, at a time. The size of the filter affects the encoded output volume, so stride is often set to a whole integer, rather than a fraction or decimal.

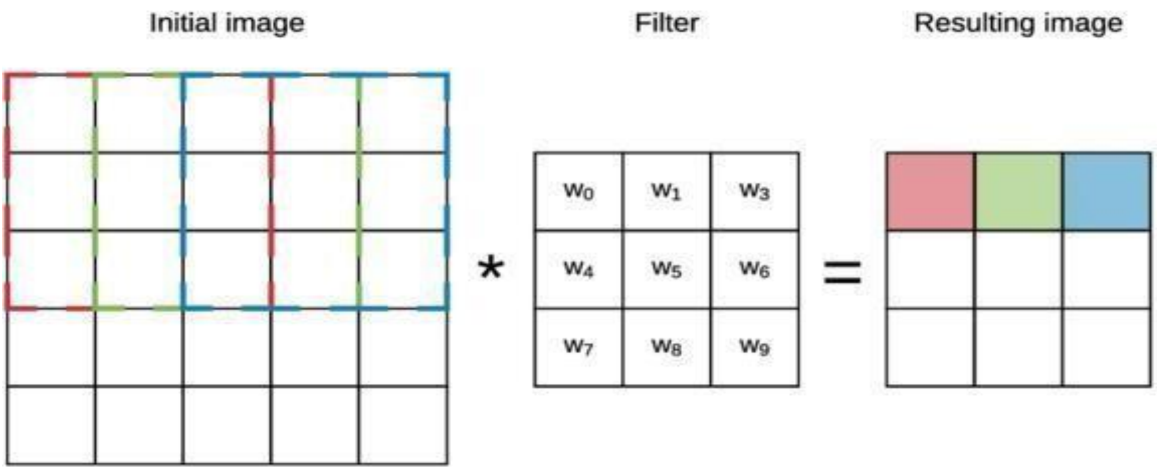


Figure 3.24 : Convolution layer working with a stride of 1.

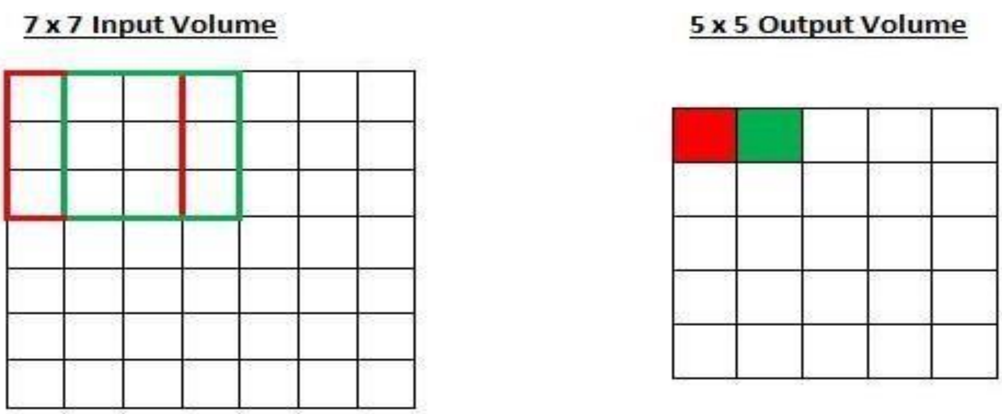


Figure 3.25 : Working of stride

Imagine a convolutional neural network is taking an image and analyzing the content. If the filter size is 3x3 pixels, the contained nine pixels will be converted down to 1 pixel in the output layer. Naturally, as the stride, or movement, is increased, the resulting output willbe smaller. Stride is a parameter that works in conjunction with padding, the feature that adds blank, or empty pixels to the frame of the image to allow for a minimized reduction of size in the output layer. Roughly, it is a way of increasing the size of an image, to counteractthe fact that stride reduces the size. Padding and stride are the foundational parameters ofany convolutional neural network.

Padding

Sometimes filter does not perfectly the input image. we have two options :

- Pad the picture with zeros (zero-padding) so that it fits.
- Drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image.

Padding is a term relevant to convolutional neural networks as it refers to the amount of pixels added to an image when it is being processed by the kernel of a CNN. For example, if the padding in a CNN is set to zero, then every pixel value that is added will be of value zero. If, however, the zero padding is set to one, there will be a one pixel border added to the image with a pixel value of zero.

Padding works by extending the area of which a convolutional neural network processes an image. The kernel is the neural networks filter which moves across the image, scanning each pixel and converting the data into a smaller, or sometimes larger, format. In order to assist the kernel with processing the image, padding is added to the frame of the image to allow for more space for the kernel to cover the image. Adding padding to an imageprocessed by a CNN allows for more accurate analysis of images.

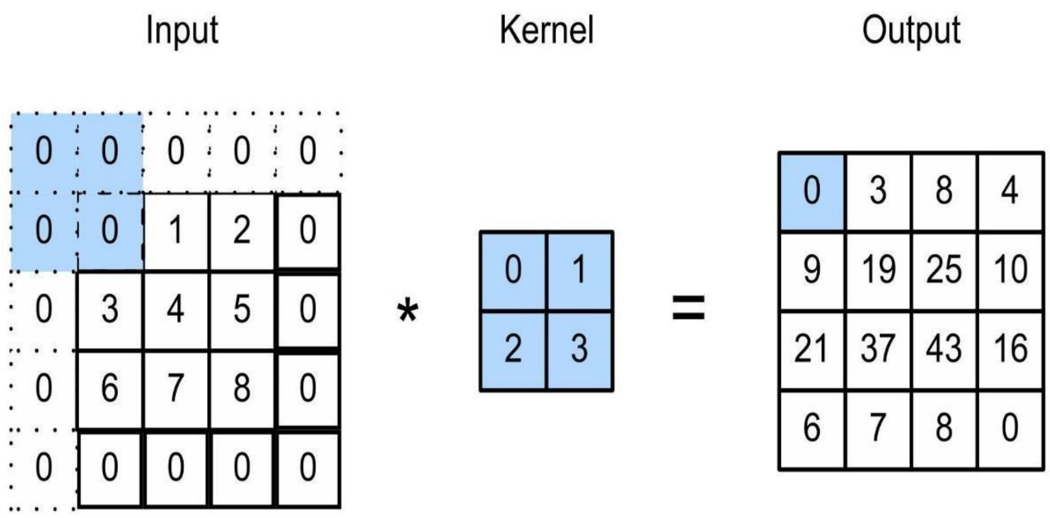


Figure 3.26 : padding

The key building block in a convolutional neural network is the convolutional layer. We can visualize a convolutional layer as many small square templates, called convolutional kernels, which slide over the image and look for patterns. Where that part of the image matches the kernel's pattern, the kernel returns a large positive value, and when there is no match, the kernel returns zero or a smaller value.

Convolution layers used trainable kernels or filters to perform convolution operations, sometimes including an optional trainable bias for each kernel. These convolution operations involved moving the kernels over the input in steps called strides. Generally, the larger the stride was, the more spaces the kernels skipped between each convolution. This led to less overall convolutions and more miniature output size. For each placement of a given kernel, a multiplication operation was performed between the input section and the kernel, with the bias summed to the result. This produced a feature map containing the convolved result. Thefeature maps were typically passed through an activation function to provide input for the subsequent layer.

Size of the feature map = [(input_size − kernel_size + 2 × padding) / stride] + 1.

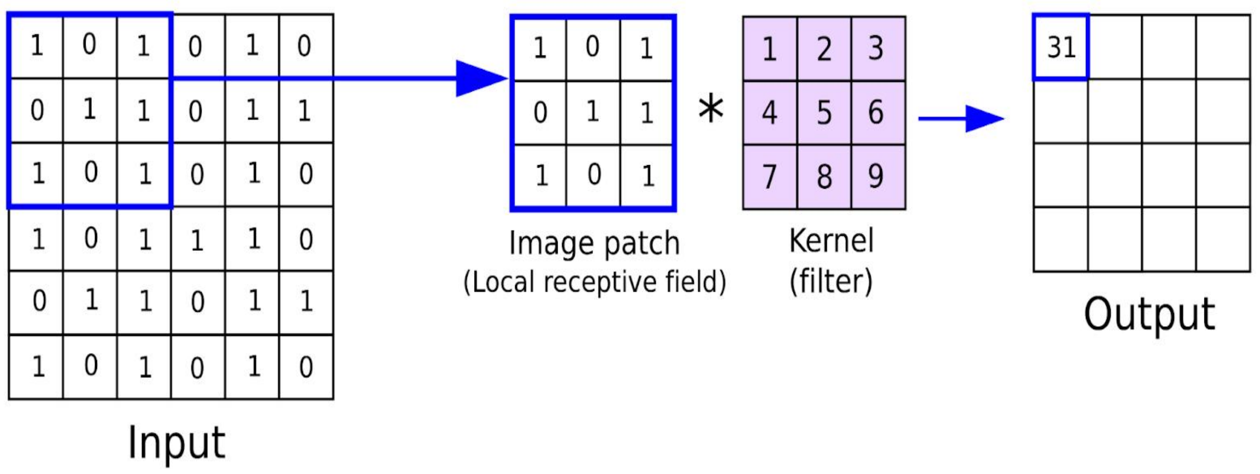


Figure 3.27 : Working of convolutional layer.

ReLU :

The rectified linear activation function or ReLU for short is a linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance. Here I am using Leaky ReLU.

<u>Difference between ReLU and Leaky ReLU</u>

- Leaky ReLU is a type of activation function that helps to prevent the function from becoming saturated at 0. It has a small slope instead of the standard ReLU which has an infinite slope
- Leaky ReLU is a modification of the ReLU activation function. It has the same form as the ReLU, but it will leak some positive values to 0 if they are close enough to zero.
- it is a variant of the ReLU activation function. It uses leaky values to avoid dividing by zero when the input value is negative, which can happen with standard ReLU when training neural networks with gradient descent.

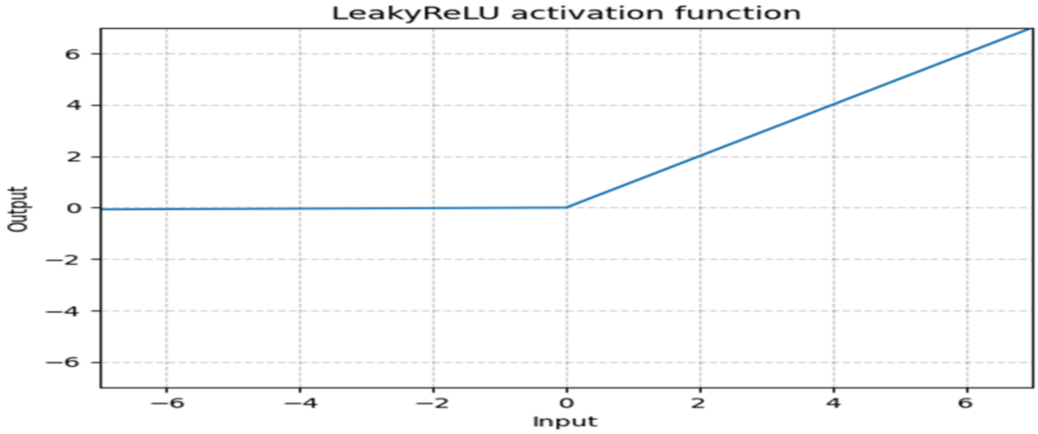


Fig 3.28: Graph of Leaky ReLU

Softmax is often used as the activation for the last layer of a classification network because the result could be interpreted as a probability distribution.

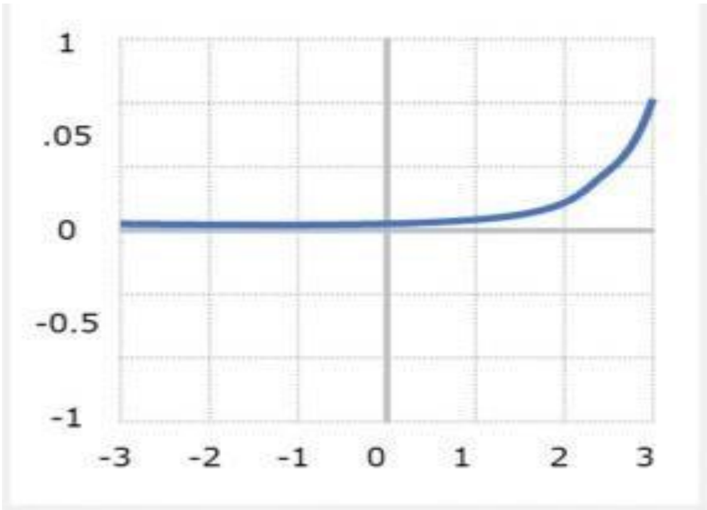


Figure 3.29 : Graph of Softmax

3.4.3. Dimension Table

	KERNEL SIZE	FILTERS	STRIDE	OUTPUT SIZE
CONV1(224*224)	7*7	64	2	112*112
MAX_POOL	3*3		2	56*56
CONV2_X	1*1 3*3 *3 1*1	64 64 *3 256	-	56*56
CONV3_X	1*1 3*3 *4 1*1	128 128 * 4 512	-	28*28
CONV4_X	1*1 3*3 *6 1*1	256 256 *6 1024	-	14*14
CONV5_X	1*1 3*3 *3 1*1	512 512 *3 2048	-	7*7
AVG_POOL+ FC	-	-	-	1*1 ,2048

Table 3.1: Dimension Table of ResNet50

3.5. Project Pipeline

Project pipeline explains the project flow. The below figure explains the project flow during the developing stage of the project. First we collect the data from different sources, then preprocess the data after that divide it into training and testing dataset. But I have training and testing folder separate. So I preprocess it and then pass it to transfer learning keras ResNet50 architecture then classify to the appropriate classe. Here we have age and gender model separately. So it classify separately. It is combined in the case of interface. ie, interface shows both the age and gender classification combinely.

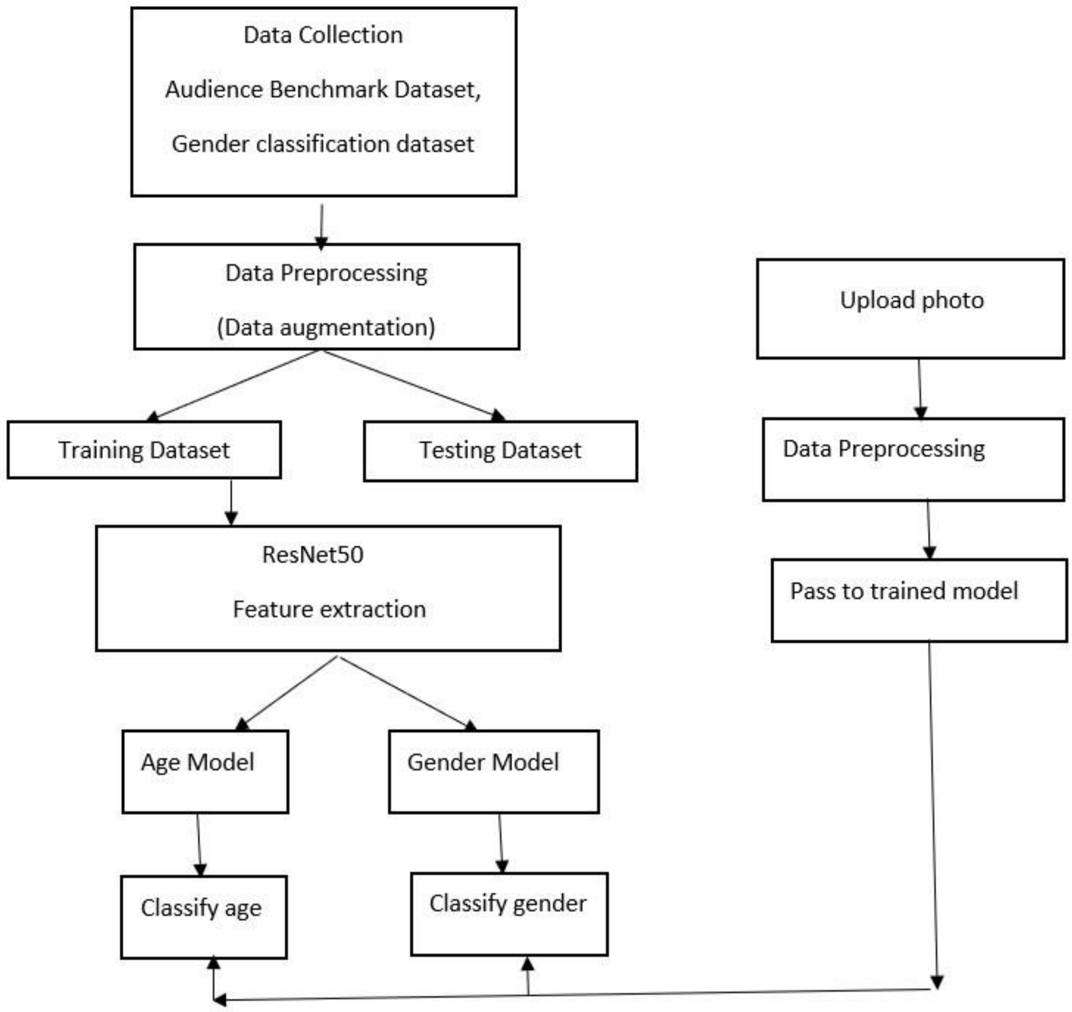


Figure 3.30 : Project pipeline

3.6. Feasibility Analysis

A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing system or proposed system, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success.

Evaluated the feasibility of the system in terms of the following categories:

- Technical Feasibility
- Economical Feasibility
- Operational Feasibility

3.6.1. Technical Feasibility

Proposed system is technically feasible since all the required tools are easily available. Technical issues involved are the necessary technology existence, technical guarantees of accuracy, reliability, ease of access, data security, and aspects of future expansion. The application is technically feasible because all the technical resources required for the development and working of the application is easily available and reliable. The project is implemented in Python. Since Python supports a various libraries and packages that make the project development easier, the project was technically feasible. The codes are written in Google Colab, therefore all the libraries will be available, no need to install or import each of those. These requirements are easily available, reliable, and will make the system more time saving and require less manpower.

3.6.2. Economic Feasibility

In our proposed system ”Age and Gender Detection”, the development cost of the application is optimum. The system requires only a computer for working. The code is working on Google Colab and Jupyter notebook , So the colab consumes an amount of internet. The development of the system will not need a huge amount of money. It will be economically feasible. But in Jupyter Notebook it needs high memory and time. But it doesn’t need internet.

3.6.3. Operational Feasibility

Operational feasibility assesses the extent to which the required system performs a series of steps to solve business problems and user requirements. Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. The developed system is completely driven and user friendly. Since the code is written on Google Colab, no need for worrying about importing or installing the libraries required. There is no need of skill for a new user to open this application and use it. The interface contain only a file upload option and a submit button. Users also need to be aware of the application initially. Then they can use it easily. So it is feasible. But sometimes it have a GPU issues. If we use jupyter notebook, then we must need a GPU in our system to get a faster user input and output.

3.7. System Environment

System environment specifies the hardware and software configuration of the new system. Regardless of how the requirement phase proceeds, it ultimately ends with the software requirement specification. A good SRS contains all the system requirements to a level of detail sufficient to enable designers to design a system that satisfies those requirements. The system specified in the SRS will assist the potential users to determine if the system meets theirneeds or how the system must be modified to meet their needs.

3.7.1. Software Environment

Various software used for the development of this application are the following :

- Python

Python is a high-level programming language that lets developers work quickly and integrate systems more efficiently. This model is developed by using many of the Python libraries and packages such as :

- Pandas :

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library.

- Matplotlib :

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals.

In this application, its used for plotting the graph.

- Numpy :

NumPy is a Python library used for working with arrays. NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays.

In this application, its used for handling arrays.

- Tensorflow :

TensorFlow is an open-source library developed by Google primarily for deep learning applications.

In this application, its used for creating and handling the model.

- Keras

Keras is a powerful and easy-to-use free open source Python library for developing and evaluating deep learning models. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code.

In this application, its used for creating and handling the model.

- OS :

The OS module in Python provides functions for interacting with the operating system. OS comes under Python’s standard utility modules. This module provides a portable way of using operating system-dependent functionality.

In this application, its used for saving the model.

- **Google Colab**

Colab is a free Jupyter notebook environment that runs entirely in the cloud. We can write and execute code in Python. Colab supports many machine learning libraries which can be easily loaded in the colab notebook.

- **Jupyter Notebook**

The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience. he Jupyter Notebook App is a server-client application that allows editing and running notebook documents via a web browser. The Jupyter Notebook App can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet.

In addition to displaying/editing/running notebook documents, the Jupyter Notebook App has a “Dashboard” (Notebook Dashboard), a “control panel” showing local files and allowing to open notebook documents or shutting down their kernels.

- **Visual Studio Code**

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE

- **HTML** and **CSS**

Hyper Text Markup Language is used for creating web pages.HTML describes the structure of the web page. Here, the user interface of my project is done using HTML. Cascading Style Sheet is used with HTML to style the web pages.

- **PHP**

Here php is used to connect to the backend. PHP is an acronym for "PHP: Hypertext Preprocessor" PHP is a widely-used, open source scripting language. PHP scripts are executed on the server. PHP is free to download and use.

- **Github**

Git is an open-source version control system that was started by Linus Torvalds. Git is similar to other version control systems Subversion, CVS, and Mercurial to name a few. Version control systems keep these revisions straight, storing the modifications in a central repository. This allows developers to easily collaborate, as they can download a new version of the software, make changes, and upload the newest revision. Every developer can see these new changes, download them, and contribute. Git is the preferred version control system of most developers, since it has multiple advantages over the other systems available.It stores file changes more efficiently and ensures file integrity better.

The social networking aspect of GitHub is probably its most powerful feature, allowing projects to grow more than just about any of the other features offered. Project revisions can be discussed publicly, so a mass of experts can contribute knowledge and collaborate to advance a project forward.

3.7.2. Hardware Environment

Selection of hardware configuration is very important task related to the software development.

Computer -

Processor : 2 GHz or faster (dual-core or quad-core will be much faster)

Memory : 4 GB RAM or greater

Disk space : 40 GB or greater

Good internet connectivity

4. SYSTEM DESIGN

4.1. Model Building

4.1.1. Model Planning

The below figure is the architecture of the model used in this application.

Age Model:-

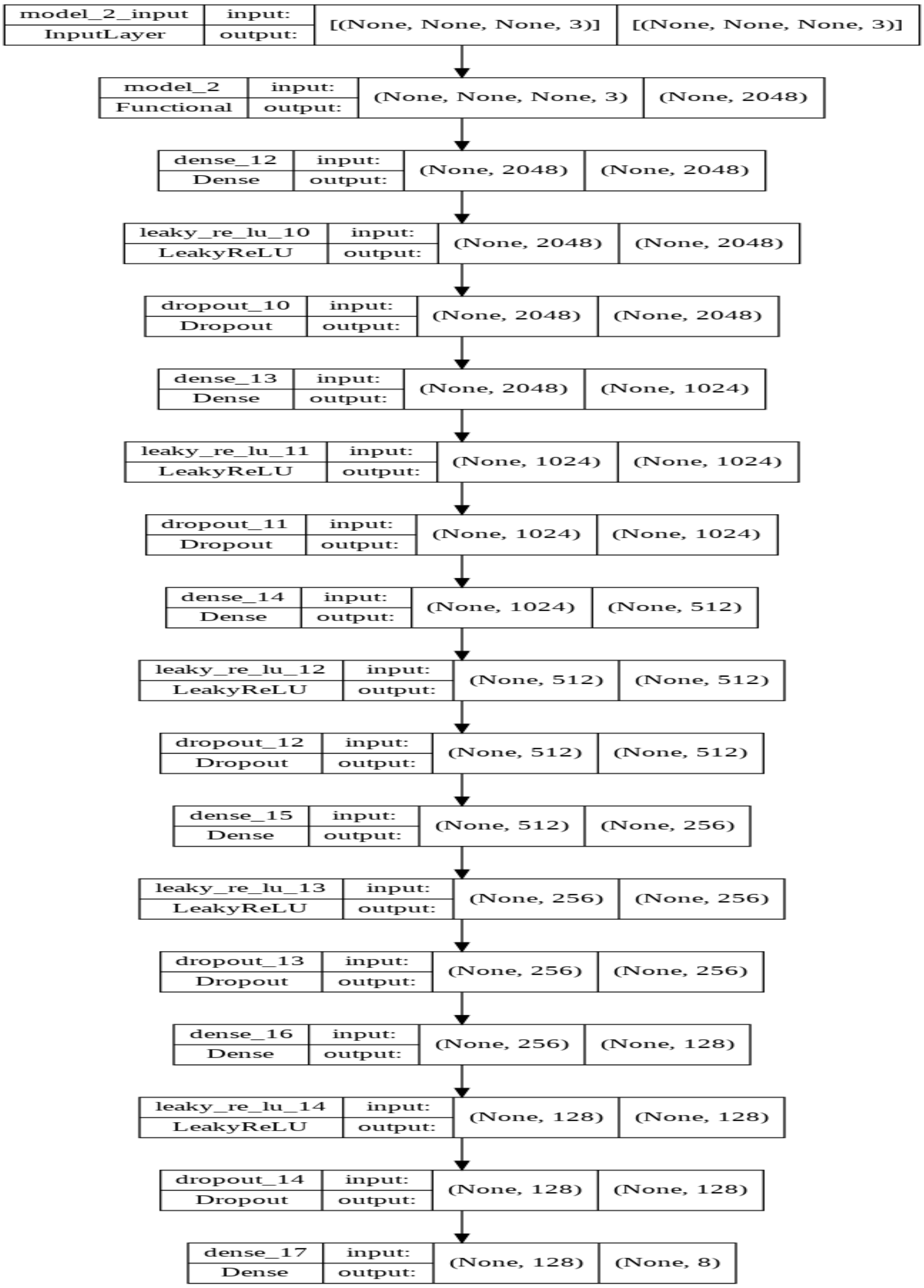


Fig 4.1 : Age Model

Gender Model:-

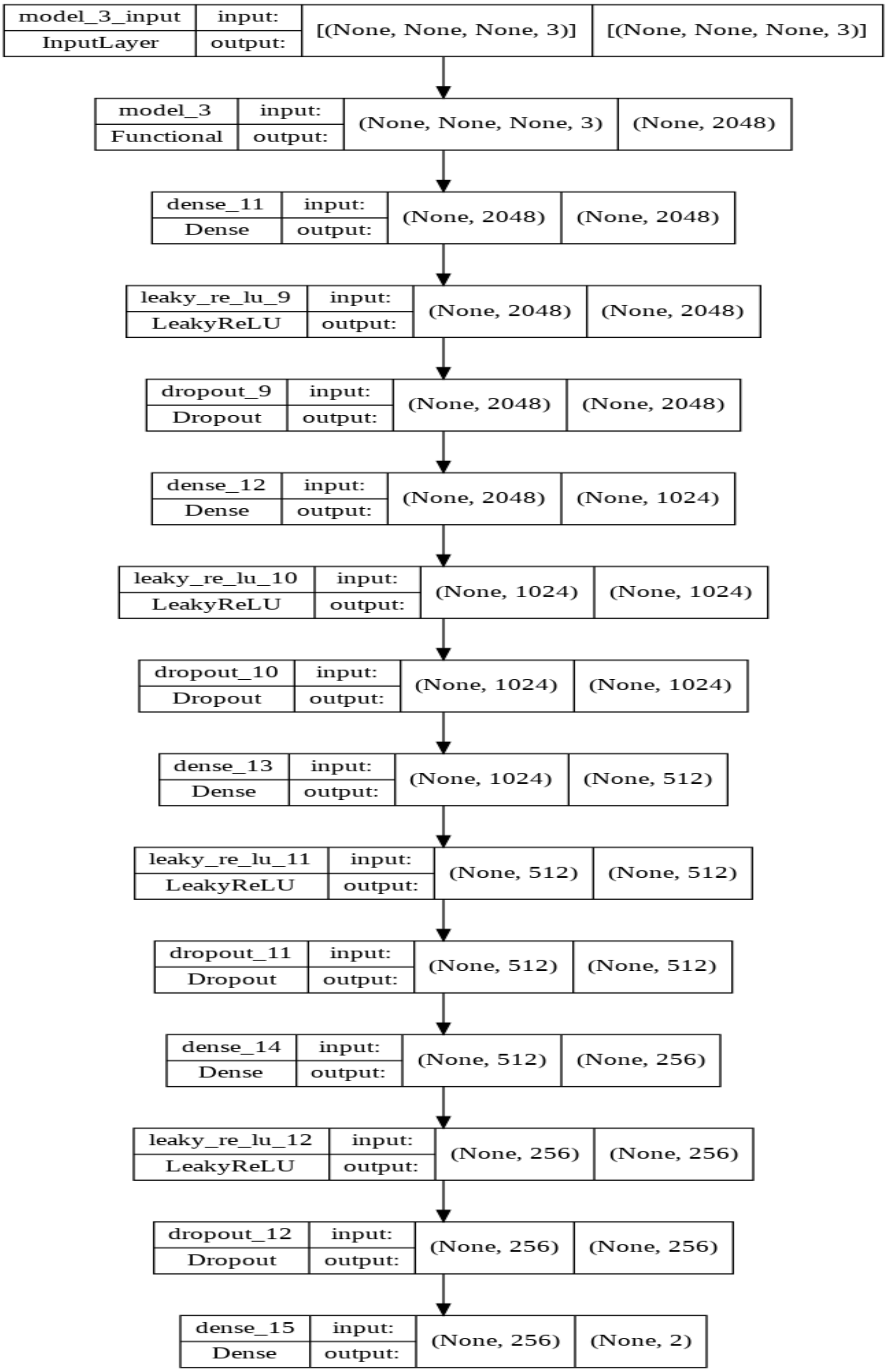


Fig 4.2: Gender Model

4.1.2. Training

```
Epoch 42/200
20/20 [==============================] - 16s 803ms/step - loss: 0.4488 - accuracy: 0.8481 - val_loss: 0.6552 - val_accura
Epoch 43/200
20/20 [==============================] - 14s 696ms/step - loss: 0.3847 - accuracy: 0.8763 - val_loss: 0.4870 - val_accura
Epoch 44/200
20/20 [==============================] - 13s 681ms/step - loss: 0.4386 - accuracy: 0.8631 - val_loss: 0.5851 - val_accura
Epoch 45/200
20/20 [==============================] - 13s 638ms/step - loss: 0.3692 - accuracy: 0.8844 - val_loss: 0.4986 - val_accura
Epoch 46/200
20/20 [==============================] - 13s 673ms/step - loss: 0.3733 - accuracy: 0.8925 - val_loss: 0.6741 - val_accura
Epoch 47/200
20/20 [==============================] - 11s 544ms/step - loss: 0.3385 - accuracy: 0.8938 - val_loss: 0.8314 - val_accura
Epoch 48/200
20/20 [==============================] - 13s 663ms/step - loss: 0.3678 - accuracy: 0.8895 - val_loss: 0.5610 - val_accura
Epoch 49/200
20/20 [==============================] - 11s 564ms/step - loss: 0.3661 - accuracy: 0.8869 - val_loss: 0.7964 - val_accura
Epoch 50/200
20/20 [==============================] - 11s 573ms/step - loss: 0.2712 - accuracy: 0.9144 - val_loss: 0.6468 - val_accura
Epoch 51/200
20/20 [==============================] - 11s 540ms/step - loss: 0.3996 - accuracy: 0.8800 - val_loss: 0.6224 - val_accura
Epoch 52/200
20/20 [==============================] - 10s 477ms/step - loss: 0.3119 - accuracy: 0.9056 - val_loss: 0.5836 - val_accura
```

Fig 4.3: Epochs of Age Model

```
Epoch 150/500
10/10 [==============================] - 26s 3s/step - loss: 0.9322 - acc: 0.7350 - val_loss: 0.5992 - val_acc: 0.6500
Epoch 151/500
10/10 [==============================] - 26s 3s/step - loss: 0.9789 - acc: 0.6900 - val_loss: 1.2089 - val_acc: 0.8500
Epoch 152/500
10/10 [==============================] - 26s 3s/step - loss: 0.6910 - acc: 0.6400 - val_loss: 0.6279 - val_acc: 0.6000
Epoch 153/500
10/10 [==============================] - 26s 3s/step - loss: 0.7638 - acc: 0.6500 - val_loss: 0.5240 - val_acc: 0.7500
Epoch 154/500
10/10 [==============================] - 26s 3s/step - loss: 0.8300 - acc: 0.6850 - val_loss: 1.4305 - val_acc: 0.5000
Epoch 155/500
10/10 [==============================] - 26s 3s/step - loss: 0.9384 - acc: 0.7400 - val_loss: 0.6337 - val_acc: 0.7500
Epoch 156/500
10/10 [==============================] - 26s 3s/step - loss: 0.9889 - acc: 0.7300 - val_loss: 0.6187 - val_acc: 0.7500
Epoch 157/500
10/10 [==============================] - 26s 3s/step - loss: 0.8136 - acc: 0.7300 - val_loss: 1.3841 - val_acc: 0.6000
Epoch 158/500
10/10 [==============================] - 26s 3s/step - loss: 0.8410 - acc: 0.7400 - val_loss: 0.4836 - val_acc: 0.9000
```

Fig 4.4: Epochs of Gender Model

4.1.3. Testing

```
Found 877 images belonging to 8 classes.
877/877 [==============================] - 33s 37ms/step
[0 5 5 0 0 0 0 5 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 5 0 0 0 0 5 0 2 7 0 0 5
 0 0 0 0 0 0 0 0 2 0 0 5 5 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 5 5 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 5 0 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0
 0 0 0 0 0 0 0 0 0 2 2 0 5 0 0 0 0 5 0 0 0 2 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0
 0 0 0 2 0 0 5 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 2 2 2 2 2 2 2 2 2
 5 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 2 2 2 7 2 2 2 2 2 2 2 2 2 7 7 2 2 2 2 5 2
 2 2 2 2 2 5 2 5 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 0 2 2 0 2 2 2 5 2 2 2 5 0 2
 2 7 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2
 2 0 2 2 2 2 2 2 2 0 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 5 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 7 2 2 2 2 2 2 2 2 2 2 2 2 2 2 7 2 2 2 0 2 2 2
 2 2 2 2 2 2 2 2 5 2 2 2 2 2 2 2 7 2 2 2 5 5 5 2 2 7 2 2 2 2 5 7 7 0 0 2 2
 2 2 2 2 2 7 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 5 5 2
 0 0 5 5 2 2 2 2 5 2 0 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 5 2 2 2 2 2 2 2 0 2
 2 2 2 2 2 2 2 0 2 2 2 0 2 2 2 0 0 2 0 2 2 2 0 2 2 2 2 2 0 2 2 2 0 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 0 2 0 0 5 2 2 2 0 0 2 5 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 0 2 0 2 2 2 5 2 2 2 5 2 0 2 2 2 2
 2 2 2 2 2 2 0 0 5 0 5 0 5 0 0 2 0 0 0 0 0 0 0 0 5 2 0 0 0 5 0 0 5 5 0 0 2
 0 2 0 5 5 7 5 0 5 5 5 5 2 5 2 0 0 2 7 5 2 5 0 0 0 0 7 5 7 5 0 0 5 5 5 5 5
 5 5 0 0 0 0 2 5 5 0 7 5 5 2 7 5 5 0 2 5 0 0 5 0 0 5 5 5 5 5 0 0 5 0 0 2 2
 5 5 7 5 7 7 5 0 5 5 5 0 5 0 5 2 2 2 2 2 2 2 0 2 2 5 2 2 2 2 2 0 2 2 2 2 2
 2 5 2 2 2 2 2 2 0 5 2 2 0 0 2 5 5 5 2 0 0 2 0 5 2 2 5 5 0 5 5 5 5 2 7 2 2
 0 2 2 2 2 7 2 2 2 2 5 0 5 0 5 0 5 5 5 5 2 7 5 0 0 2 5 5 7 7 0 0 0 5 2 5 5
```

Fig 4.5: Testing of Age Model

```
Found 11649 images belonging to 2 classes.
11649/11649 [==============================] - 740s 64ms/step
[0 0 0 ... 0 1 1]
```

Fig 4.6: Testing of Gender Model

5. RESULTS AND DISCUSSION

The accuracy is the metrics used in the training of the dataset. Accuracy is a measurement of observational error. It defines how close or far off a given set of measurementare to their true value.

First with gender model , for the 1st epoch the accuracy is 0.65. For 142thth one it is increased to 0.70. For 500thepoch it is 0.90.

For age model ,the 1st epoch has .17 accuracy , the 20th has 0.57 accuracy , the 56th has 0.85 accuracy ,the 105th epoch has 0.89 accuracy.

This shows that by increasing the number of epochs in training, we can improve the quality of prediction. It is like, more trained the system is, more correct the prediction will be.

Confusion Matrix

Gender Model:-

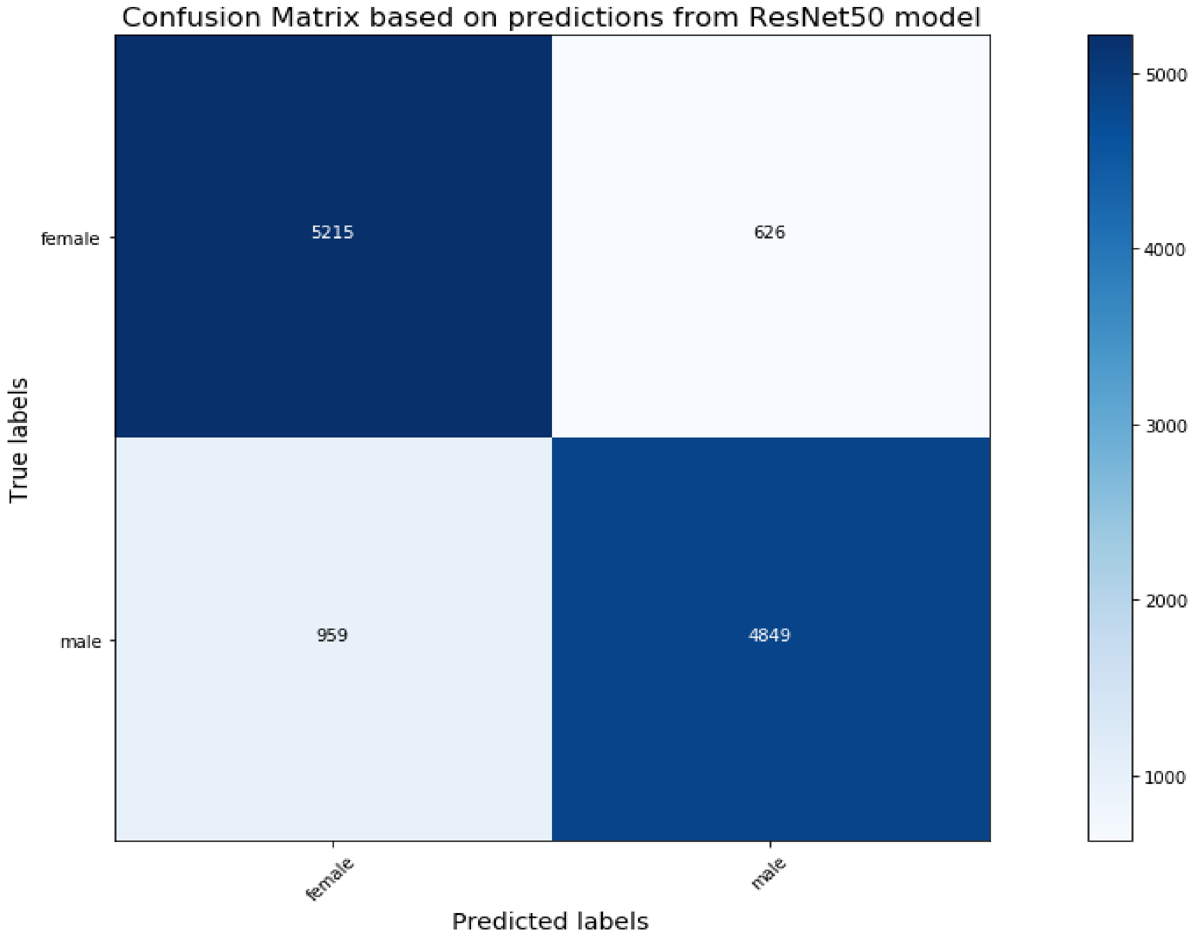


Fig 5.1 : Confusion Matrix of Gender Model

Age Model:-

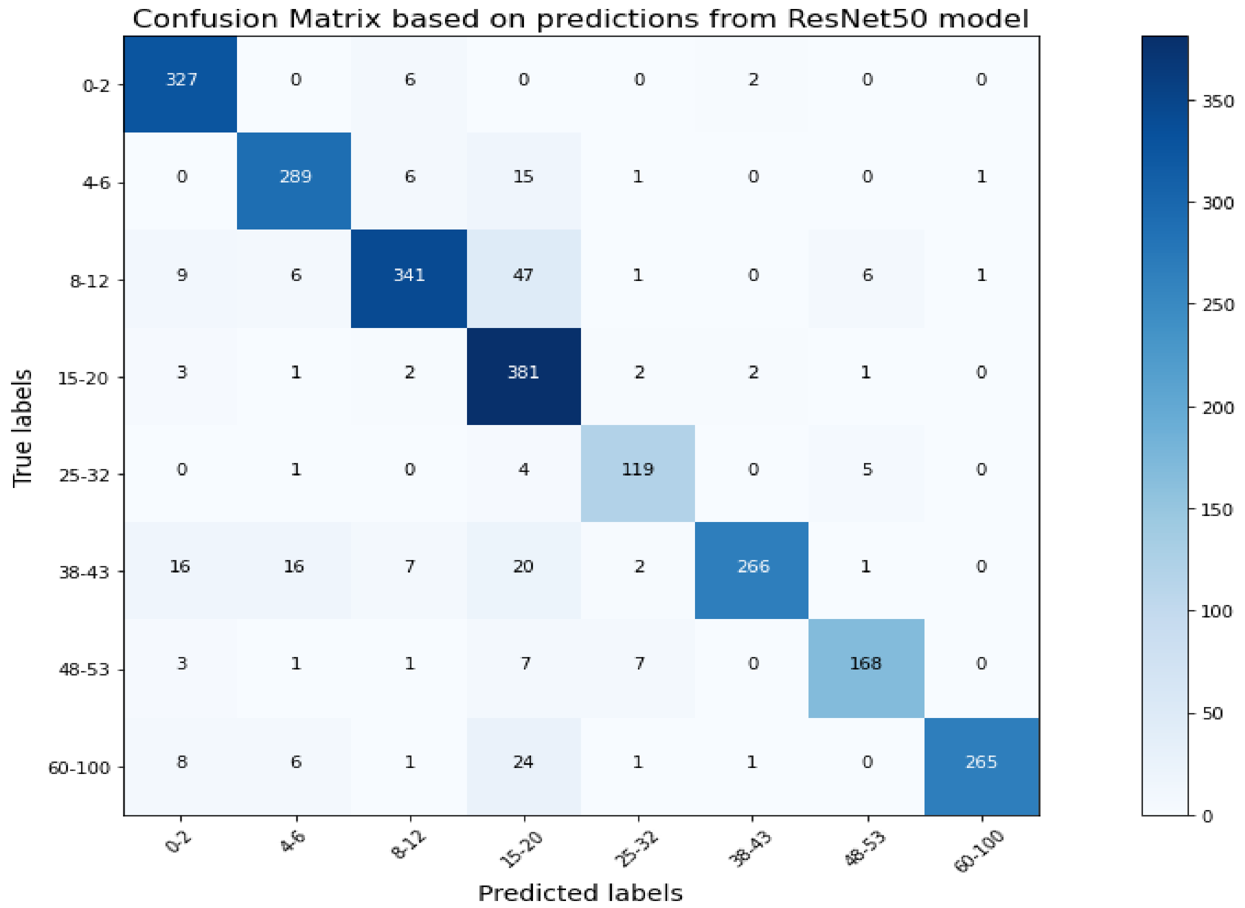


Fig 5.2: Confusion Matrix of Age Model

Classification Report

Age Model:-

	precision	recall	f1-score	support
0	0.89	0.98	0.93	335
1	0.90	0.93	0.91	312
2	0.94	0.83	0.88	411
3	0.77	0.97	0.86	392
4	0.89	0.92	0.91	129
5	0.98	0.81	0.89	328
6	0.93	0.90	0.91	187
7	0.99	0.87	0.92	306
accuracy			0.90	2400
macro avg	0.91	0.90	0.90	2400
weighted avg	0.91	0.90	0.90	2400

Gender Model:-

```
print(classification_report(ac_label,pc_label))
```

	precision	recall	f1-score	support
0	0.84	0.89	0.87	5841
1	0.89	0.83	0.86	5808
avg / total	0.87	0.86	0.86	11649

Accuracy and Loss in epochs

Age Model:-

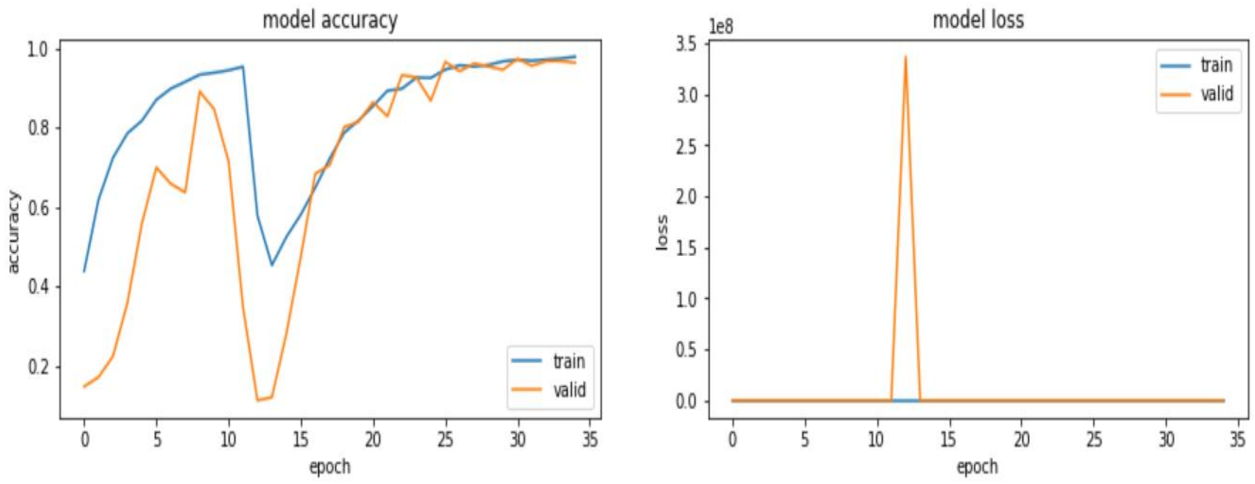


Fig 5.3 : Accuracy-Loss Graph of Age Model

Gender Model:-

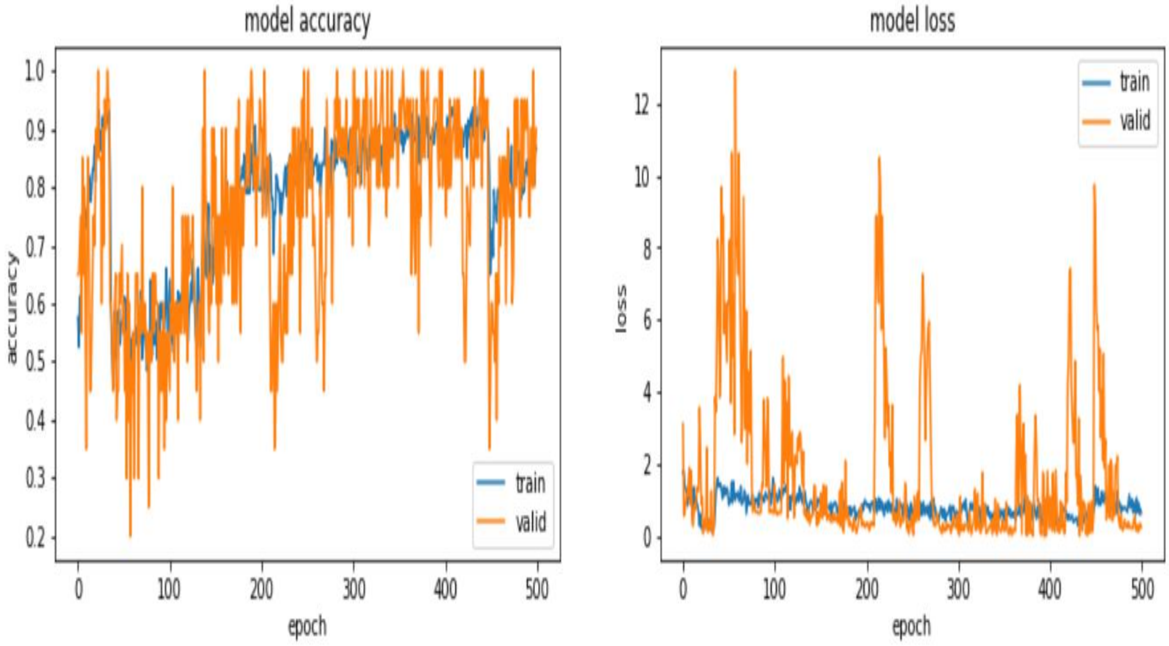


Fig 5.4 Accuracy-Loss Graph of Gender Model

6. MODEL DEPLOYMENT

This figure shows the user interface of this application. The interface is very simple and easy to understand. There are only some elements displayed on the screen. There is a file upload option provided. The user can choose an image file. Then there is a submit button. On click of the button, the image uploaded will be given to the model. After processing for few seconds, the image we given and the prediction by the model will be displayed as output as shown below :

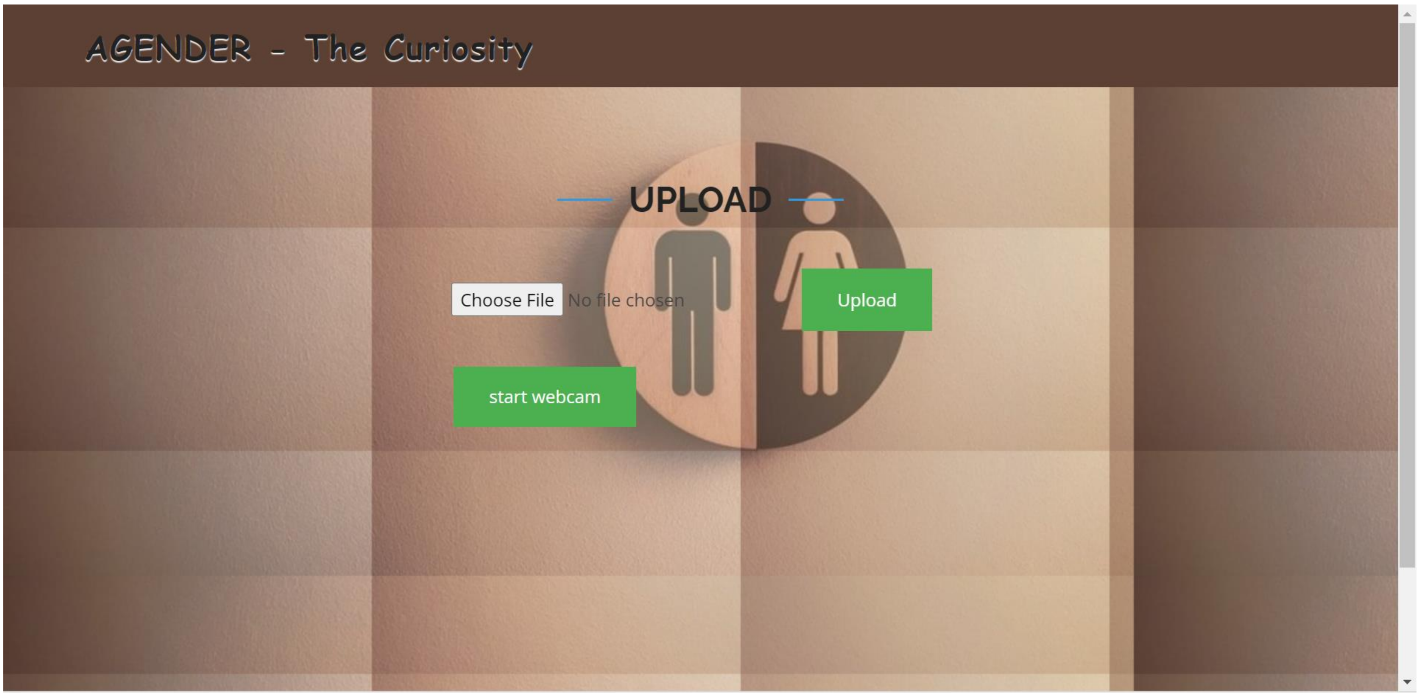


Fig 6.1: UI Design

Figure 6.2: Result of Image Feed

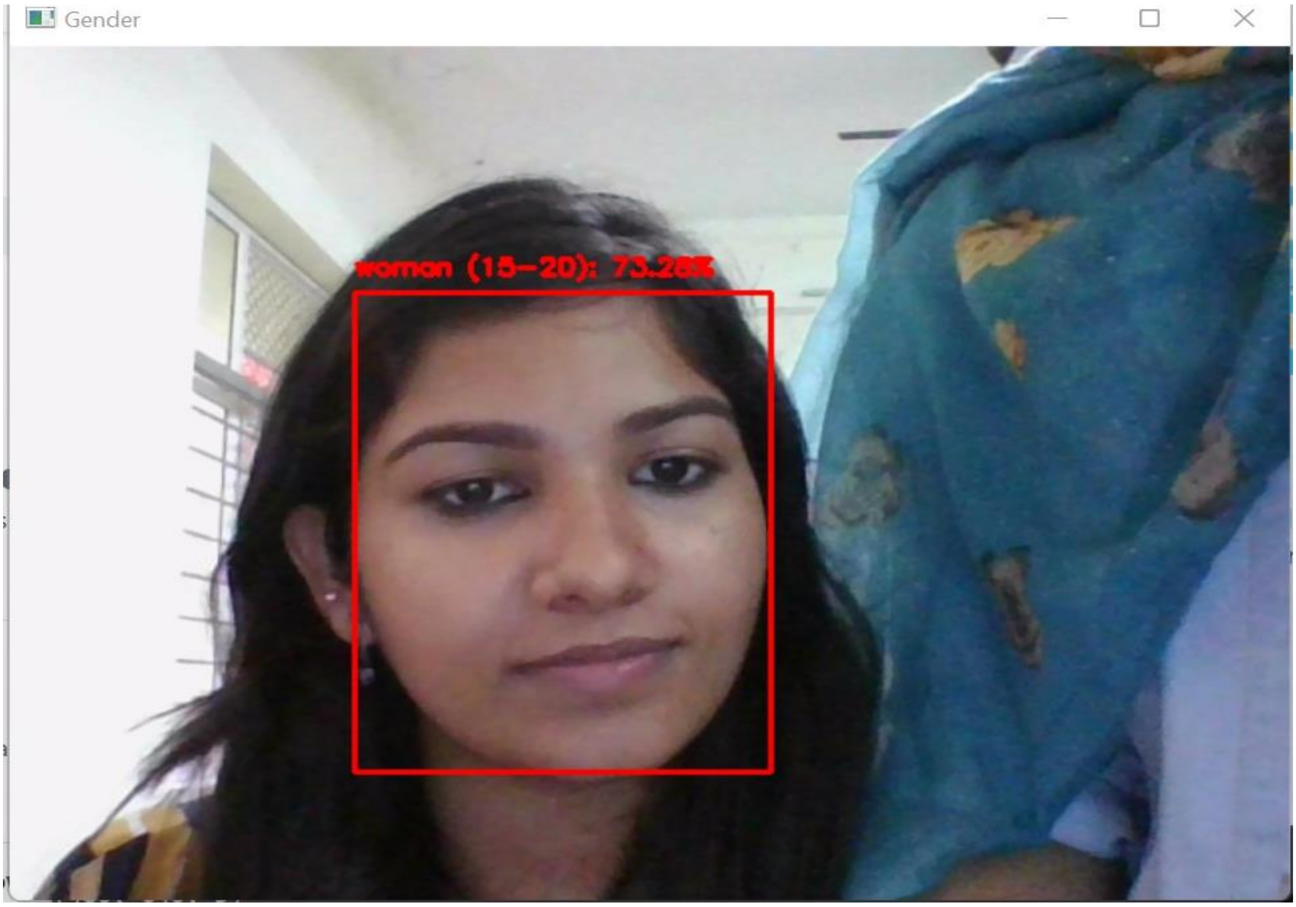


Fig 6.3: Result of Live Feed

7. GIT HISTORY

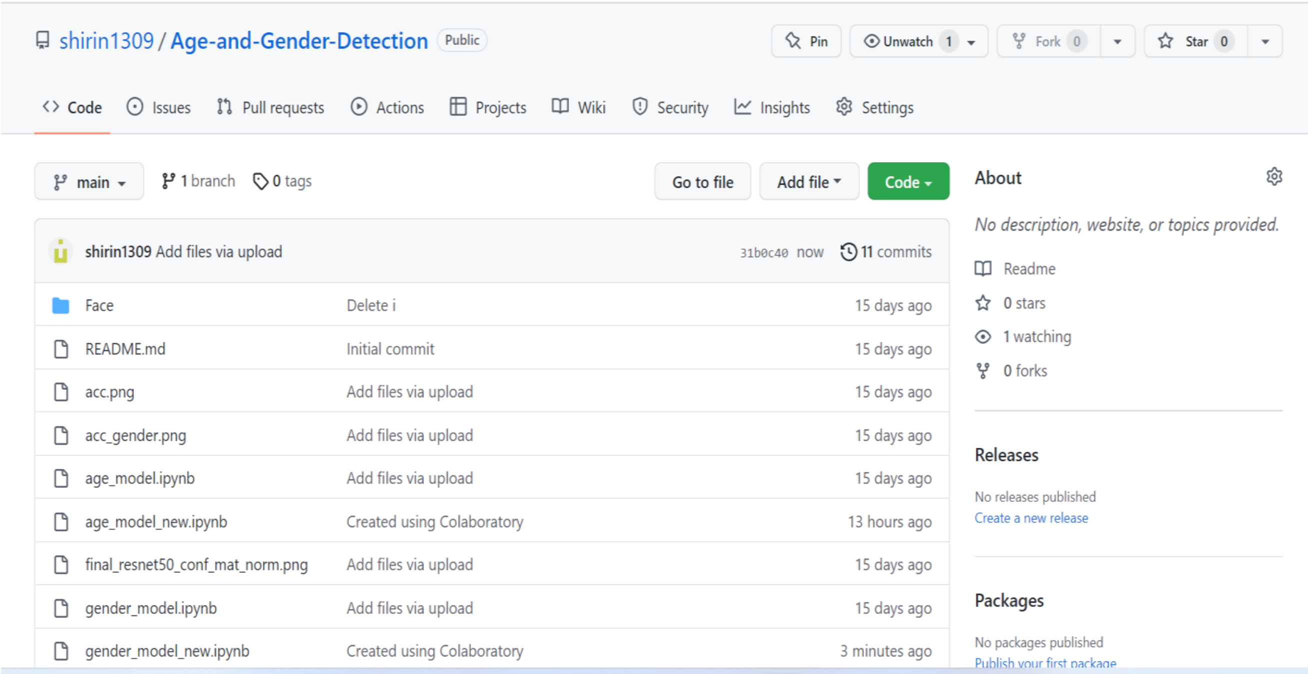


Fig 7.1 Git History

8. CONCLUSIONS

This project is a deep learning project, which aims to detect age and gender from the image. The idea to develop this application as my academic project came to my mind after reading an IEEE paper. For getting more understanding I referred other papers on the same topic also. As the paper suggested it is done using CNN. So I use most recent CNN architecture ResNet50 for model building. And also I made a real time age and gender detector, but because of real life constraints and my webcam , it doesn’t shows an accurate result.

There are 2 classes for gender and 8 classes for age. I use age classification rather than age prediction. Because it is a tedious task to predict correct age. So I use classification rather than prediction. The gender model has 90% accuracy and age model has 89% accuracy as the better one.

I used Google Colab and Jupyter Notebook for developing this application. It made the work so easy and efficient.

9. FUTURE WORK

The Age and Gender Detection system can be implemented in following sectors:

1) Security Control, as it can control minors from purchasing alcohol and other prohibited objects under the underage acts.
2) Human-Computer Interaction, as the system can show only the content based on the age of the person.
3) Law Enforcement, as the automatic age and gender detection, can determine potential suspects very efficiently based on age and gender. This system can be broadly deployed for security purposes, as in airports and police checkpoints.

10. APPENDIX

10.1. Minimum Software Requirements

Software: Jupyter Notebook, Google Colab

Operating System : Windows

10.2. Minimum Hardware Requirements

Hardware capacity	:	256 GB (minimum)
RAM	:	4 GB
Processor	:	Intel Core i3 preferred
Display	:	1366 * 768

11. REFERENCES

- https://colab.research.google.com
- https://iq.opengenus.org/resnet50-architecture/
- https://cv-tricks.com/keras/understand-implement-resnets/
- https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/
- https://iq.opengenus.org/relu-activation/
- https://ieeexplore.ieee.org/search/advanced/citation
- https://talhassner.github.io/home/projects/Adience/Adience-data.html
- https://www.kaggle.com/datasets/cashutosh/gender-classification-dataset
- https://chroniclesofai.com/transfer-learning-with-keras-resnet-50/
- https://codepen.io/TheLukasWeb/pen/kOrJmR
- https://machinelearningknowledge.ai/keras-implementation-of-resnet-50-architecture-from-scratch/
- Aryan Saxena, Prabhangad Singh , Shailendra Narayan Singh(2021),IEEE , Age and gender detection using deep learning.
- Insha Rafique, Awais Hamid, Sheraz Naseer (2019) , IEEE, Age and gender detection using deep convolutional neural network .
- Mohammed Kamel Benkaddour , Sara Lahlali, Maroua Trabelsi(2020), IEEE, Human age and gender classification using convolutional neural network.