

GENDER, AGE AND ETHNICITY DETECTION

Project report submitted in partial fulfillment of the requirement for
the degree of Bachelor of Technology

in

Computer Science and Engineering

By

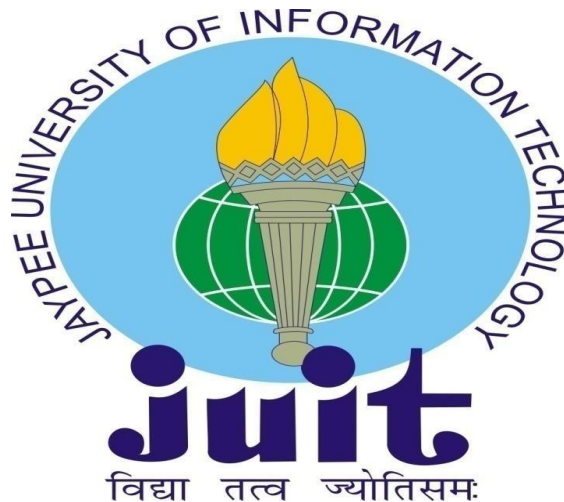
Akshat Agrawal (181207)

Sahil Thakur (181221)

Under the supervision of

Dr. Amol Vasudeva

to



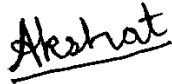
Department of Computer Science & Engineering and Information
Technology

**Jaypee University of Information Technology Waknaghat,
Solani-173234, Himachal Pradesh**

CERTIFICATE

Candidate's Declaration

We hereby declare that the work presented in this report entitled “ **Gender and Age Detection**” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2021 to December 2021 under the supervision of **Dr. Amol Vasudeva** (Assistant Professor (Senior Grade) , Department of Computer Science & Engineering and Information Technology Jaypee University of Information Technology). The matter embodied in the report has not been submitted for the award of any other degree or diploma.



Akshat Agrawal, 181207.



Sahil Thakur, 181221.

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Amol Vasudeva
Designation
Assistant Professor (Senior Grade)
Department of Computer Science & Engineering and Information Technology
Dated:- 29-05-2022

ACKNOWLEDGEMENT

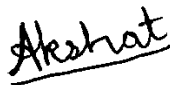
Firstly, we express my heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully.

We are really grateful and wish my profound my indebtedness to Supervisor Dr. Amol Vasudeva, Assistant Professor (Senior Grade), Department of CSE Jaypee University of Information Technology, Wakhnaghat. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to express my heartiest gratitude to Dr. Amol Vasudeva, Department of CSE, for his kind help to finish my project.

We would also generously welcome each one of those individuals who have helped me straight forwardly or in a roundabout way in making this project a win. In this unique situation, we might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, we must acknowledge with due respect the constant support and patients of my parents.



Akshat Agrawal
(181207)



Sahil Thakur
(181221)

TABLE OF CONTENTS

CERTIFICATE	I
ACKNOWLEDGEMENT	II
TABLE OF CONTENTS.....	III
LIST OF ABBREVIATIONS	IV
LIST OF FIGURES	V
LIST OF TABLES	VII
ABSTRACT	VIII
CHAPTER-01: INTRODUCTION	1
1.1 Introduction:	1
1.2 Problem Statement:	2
1.3 Objectives:	2
1.4 Methodology:	2
1.5 Organization:.....	3
CHAPTER-02: LITERATURE SURVEY	5
2.1 Literature Survey	5
CHAPTER-03: SYSTEM DEVELOPMENT	12
3.1 Analysis/Design/Development/Algorithm.....	12
3.2 Model Development	16
CHAPTER-04: PERFORMANCE ANALYSIS	28
4.1 Analysis of system developed.....	28
4.2 Output at various stages	29
4.3 Comparison of above results by at least two methods and justification for the differences or error in with theory or earlier published results.....	39
CHAPTER-05: CONCLUSIONS	40
5.1 Conclusions	40
5.2 Future Scope.....	40
5.3 Applications	41
REFERENCES.....	42

LIST OF ABBREVIATIONS

- 1) ML: Machine Learning
- 2) CNN: Convolutional Neural Network
- 3) DL: Deep Learning
- 4) CSV: Comma Separated Values
- 5) API: Application Program Interface
- 6) FC: Fully connected
- 7) CLI: Command Line Interface
- 8) DNN: Deep Neural Network
- 9) XML: Extensible Markup Language
- 10) PCA: Principal Component Analysis
- 11) MAE: Mean Absolute Error

LIST OF FIGURES

1) Fig.1 Flowchart of Eigenfaces method	Pg-07
2) Fig.2 Block Diagram of CNN method for prediction	Pg-09
3) Fig.3 Development and process flow	Pg-12
4) Fig.4 Architecture showing web service and endpoints	Pg-16
5) Fig.5 UTKFace Dataset	Pg-17
6) Fig.6 CNN Architecture	Pg-19
7) Fig.7 Convolutional Layer	Pg-19
8) Fig.8 Pooling Layer	Pg-20
9) Fig.9 Flattening Layer	Pg-20
10) Fig.10 Relu function	Pg-21
11) Fig.11 Sigmoid function	Pg-22
12) Fig.12 Softmax function	Pg-22
13) Fig.13 Flowchart of Process	Pg-27
14) Fig.14 Face Detection	Pg-29
15) Fig.15 Count for unique ages	Pg-29
16) Fig.16 Age Distribution	Pg-29
17) Fig.17 Age Distribution Bar Graph	Pg-30
18) Fig.18 Gender Distribution Bar Graph	Pg-30
19) Fig.19 Age Bins Bar Graph	Pg-30
20) Fig.20 Ethnicity Distribution Bar Graph	Pg-31
21) Fig.21 Sample images with age, gender, bin	Pg-31
22) Fig.22 Converting Variables to Categorical	Pg-31
23) Fig.23 Creating test and train data	Pg-32
24) Fig.24 Model-1 Summary	Pg-32
25) Fig.25 Model-2 Summary	Pg-33
26) Fig.26 Training Model	Pg-33
27) Fig.27 Model-1 Evaluation	Pg-34
28) Fig.28 Model-2 Evaluation	Pg-34
29) Fig.29 Model-1 Accuracy and Loss vs Epoch	Pg-35

30) Fig.30 Model-1 Predicted vs Actual Age	Pg-35
31) Fig.31 Model-2 Age Accuracy and Loss vs Epoch	Pg-36
32) Fig.32 Model-2 Gender Accuracy and Loss vs Epoch	Pg-37
33) Fig.33 Model-2 Ethnicity Accuracy and Loss vs Epoch	Pg-37
34) Fig.34 Model-1 Sample Prediction	Pg-38
35) Fig.35 Model-2 Sample Prediction	Pg-38
36) Fig.36 Heroku deployed website and results	Pg-39

LIST OF TABLES

- | | |
|---|-------|
| 1) Table.1 CNN Model-1 | Pg-23 |
| 2) Table.2 CNN Model-2 | Pg-25 |
| 3) Table.3 Analysis of system developed | Pg-28 |

ABSTRACT

There is a lot of interest in facial analysis since it can assist with a range of challenges, including better targeted advertising, strong content recommendation engines, and more. It is essential to determine one's 'age, gender and ethnicity' in order to carry out facial analysis, and this is the most crucial phase in the process. The project's goal is to create a model that can estimate a person's 'age, gender and ethnicity' based on their appearance and provide a working live webapp for the same. Organizations need to know the 'age, gender and ethnicity' range of their consumers in order to understand their demands and build strategies to give better services to them. In order to collect data from their customers, these organizations rely heavily on their enterprise architecture, and forms play a significant role in this process. There may be a certain format for this information, and it must adhere to it. As a result, application form validation and recommendations were created. There is a wide range of data types that can be validated by form validators, according to previous research. Auto-detection of a user's 'age, gender and ethnicity' are currently unavailable. The age range of the user is not discovered or checked based on the user's picture. People's age must be clearly visible on their photo-id. As part of this study, we present a unique approach to sexuality and target age group prediction based on a picture. A Convolutional Neural Network (CNN) is used to detect the 'age, gender and ethnicity' of a single person's picture, receiving input from a form that includes the user's photo. Finally, Heroku is used to host a website that facilitates the predictive model.

CHAPTER-01: INTRODUCTION

1.1 Introduction:

In computer vision, determining a person's age or gender is one of the most difficult tasks. You may use it in a broad variety of ways: demographic segmentation; consumer analytics; visual monitoring; and age progression. It is essential to identify a person's 'age, gender and ethnicity' by extracting characteristics from collected photographs [1]. Since recently, CNNs, a deep learning technique, have helped researchers access this information. In order to accurately anticipate a human's 'age, gender and ethnicity' in an ever-changing external environment, this technique suggests employing deep learning. A CNN network topology and training algorithm, as well as a performance evaluation, are built to do this.

By checking and correcting user input, form validators help to ensure that business systems get accurate data from their users. The user photo validation procedure is highly influenced by face, age, and gender identification algorithms. For the sake of this project, let's have a look at some of the most important algorithms. In order to accurately identify 'age, gender and ethnicity', face detection is a crucial step. In a shot, it can tell the difference between human faces and other objects [2]. This job may be accomplished using a number of different methods, including the Haar Cascade, Deep neural networks, Eigenface methods, Fisherface methods, and the Convolutional Neural Network, each having its own set of applications, advantages, and disadvantages. Using pictures, the CNN is one of the most often used deep learning methods [3]. This method is particularly effective in our situation since it treats 'age, gender and ethnicity' range' prediction as a classification problem, with gender classes "Male" and "Female" and many age range classes. This work aims to build a model that can predict 'age, gender and ethnicity' range' from user images, based on an automated age/gender and ethnicity classification approach that employs CNN, and assess it using a dataset of real people's photos.

1.2 Problem Statement:

When it comes to collecting data from users, forms are vital in business systems, and form validators aid in the collection of trustworthy data by verifying submitted data and deleting user input mistakes. User picture validation algorithms that recognize face, age, and gender are heavily influencing the 'age, gender and ethnicity' displayed by the process of validating a user's photo. The purpose of this research is to develop a model that can estimate a person's 'age, gender and ethnicity' based on their use of the CNN. It was decided to employ the current standard face recognition technology, which includes an algorithm for extracting different features from photographs of faces and an identification model that uses those traits to determine who the person is [5]. An eye, nose, mouth, contour, and contrast information, as well as information on their relative 'sizes, shapes, and correlations', are all used by the content face recognition approach to identify whether or not a face is present. In spite of the fact that this method is simple and easy to use, it is not affected by the position or orientation of the face in issue. But due of variations in the brightness of the light, the background or an object that is close the skin tone may be misread, resulting in loss of the color and texture information of the facial features. There is also a risk that it may not be able to recognize the unique characteristics of the face, depending on how much the face is tilted. As for deep learning-based face recognition systems, they build their feature sets by evaluating massive volumes of face data gathered in various contexts and then performing extraction of features and classification algorithm learning from scratch. Therefore, it has the advantage of having extremely strong face recognition performance in the wild as a result.

1.3 Objectives:

The goal of this research is to develop a model that can estimate a person's 'age, gender and ethnicity' based on information from CNN. The next step is to pack the model and transport it to a site where it may be used to demonstrate the model in action.

1.4 Methodology:

Many different approaches were used to categorize the 'age, gender and ethnicity' of the people in the images. A machine learning technology called convolution neural network

CNN was used in this study to construct a model that could be used to predict the outcome of a game. Just a few of the techniques that were employed to execute this project are listed below. One of the difficulties in automatic gender categorization is taking into consideration the impact of posture, lighting, and ambient noise on the results. In order to resolve these challenges, practical frameworks must be trustworthy enough. It is assumed that pre-aligned frontal photographs of faces are accessible and free of distracting background noise in the vast majority of gender classification research studies.

The means available for the proposed strategy are as follows:

- Detect faces from images in dataset using pretrained 'opencv' function.
- Delete unwanted and outlier images with incorrect face detections and poor resolution.
- Save files with image name containing 'age, gender and ethnicity'.
- Read pixels from images.
- Develop CNN model.
- Create training data and testing data from variables.
- Train model and testing it.
- Finalizing and packing model for deployment.
- Deploy model on Heroku that predicts age gender of person from image provided in form.

1.5 Organization:

This report is divided into five modules, and a full description of each module of this project has been provided for the sake of clarity and comprehension.

Module 1: This module serves as a formal introduction to the project and contains all of the necessary information. In this section, we introduce the reader to the project's numerous terms while also outlining the problem or motive for undertaking the project in the first place. Along with this, we're also getting started on our project's purpose and the methods that will be used to carry it out.

Module 2: This module contains a collection of current research studies that have been conducted in connection with our project. In this section, we place a greater emphasis on the methodology that the articles employed. In addition to this, we're keeping an eye on the results of their various initiatives as well.

Module 3: We will go through the many stages of our development in this module, and we will learn about the design and implementation of our algorithm. In this section, we will also construct the model and attempt to represent it from a variety of perspectives, including analytical, computational, experimental, mathematical, and statistical perspectives, among others.

Module 4: In this module, we will examine the performance of our project and provide recommendations for improvement.

Module 5: This will be our last module, in which we will discuss the outcome of our research as well as examine our findings and conclusions. In addition to this, we will examine the project's future scope as well as any enhancements that may be feasible in the near future. In addition, we will describe some of the applications in which the system might be beneficial.

CHAPTER-02: LITERATURE SURVEY

2.1 Literature Survey

We have a lot of alternatives for dealing with this situation. While there are ways for face recognition and feature relation approaches, such as "Fisherfaces" and "Eigenface," they do not perform as well as they could, owing to a number of factors. CNN, the most popular model for face recognition work, may be used to get better results. For picture datasets, they've shown to be the most successful, and they form the basis of most machine learning algorithmic frameworks.

1.)

Title- “‘Age, gender and ethnicity’ Prediction and Validation Through Single User Images Using CNN”

Author- Abdullah M. Abu Nada, Eman Alajrami, Ahemd A. Al-Saqqa, Samy S. Abu-Naser

Date- 8, August, 2020

Published In- “International Journal of Academic Engineering Research (IJAER)”

They created a suggested strategy for predicting user ‘age, gender and ethnicity’ by employing the most recent OpenCV library as well as the ‘Caffe deep learning framework’ to forecast user ‘‘age, gender and ethnicity’’. As testing data, a collection of photographs taken by ‘University of Palestine’ undergraduates was selected. In this subset, students are divided into two groups according to their gender, which are labelled as 'Male' and 'Female.' These groups are utilized for testing and taking into account factors such as the student's ‘age, gender and ethnicity’. Age, gender, and the date of a photograph's taking have lately become crucial pieces of information for a number of businesses and government organizations for a variety of objectives, including commercial, identity, security, and other concerns. Furthermore, because this information was gathered from individuals through an enterprise system, form validators were recommended in order to reduce the number of data entry errors made by users. By estimating ‘‘age, gender and ethnicity’’ from a single person picture and comparing it to the real gender and time stamp obtained, this study was

intended to give a unique method for confirming these data [6]. After that, they evaluated it and determined that it worked well in terms of gender predicting but poorly in terms of age forecasting.

2.)

Title- "A Face Recognition System Based on Eigenfaces Method"

Author- 'Müge Çarıkçı, Figen Özen'

Date- 23, February, 2012

Published In- 'Haliç University, Electrical and Electronics Engineering Department, Şişli, Istanbul, Turkey'

In the Eigenfaces technique, the "Principal Component Analysis" (also known as "PCA") serves as the basis. The "Eigenfaces and PCA algorithms" were used by 'Sirovich and Kirby' to efficiently represent facial pictures [7]. They started with a collection of original pictures and worked their way up to determining the best vector technique for image fusion. 'Turk and Pentland' used the Eigenfaces to the problem of face identification after that, and they were successful. A vector projection method known as Principal Component Analysis (PCA) is extensively employed in pattern categorization. PCAs attempt to replace highly correlated vectors of high dimensions with statistically independent vectors of lower dimensions in order to improve their performance. Another purpose is to construct a basis for the set of data that has been collected. The most significant advantages of PCA are that it is least sensitive to loudness, that it requires less memory and storage capacity, and that it increases productivity by allowing for execution in a little amount of area. According to this approach, the typical features of the face are extracted first, and then the face in question is represented as a weighted sum of the so-called "eigenfaces" that were acquired from the feature extraction stage. Recognizing a face involves projecting it into the area created by the template matching and comparing the results. A link is established based on the 'Euclidian distance' between the eigenvectors of the eigenfaces and the eigenface of the relevant picture in topic (see figure). If the distance between the two points is narrow enough, the user can be highlighted. Though the distance between the two grows too

considerable, the picture is regarded as if it belonged to a user who has to be taught how to use the system.

The Eigenfaces technique is employed on a very large dataset of 3040 photographs, which is used to demonstrate its effectiveness. These aspects, which include the backdrop, sunglasses, beard and moustache, have been discussed in some detail. Failure is a possibility, according to the simulation results, but only on rare occasions. 94.74 percent of the attempts are successful, according to the estimates. The eigenfaces technique can be complemented with other information, such as the face pyramid, to increase the likelihood of a successful match. It is intended that the next study will focus on increasing the success rate for really large datasets.

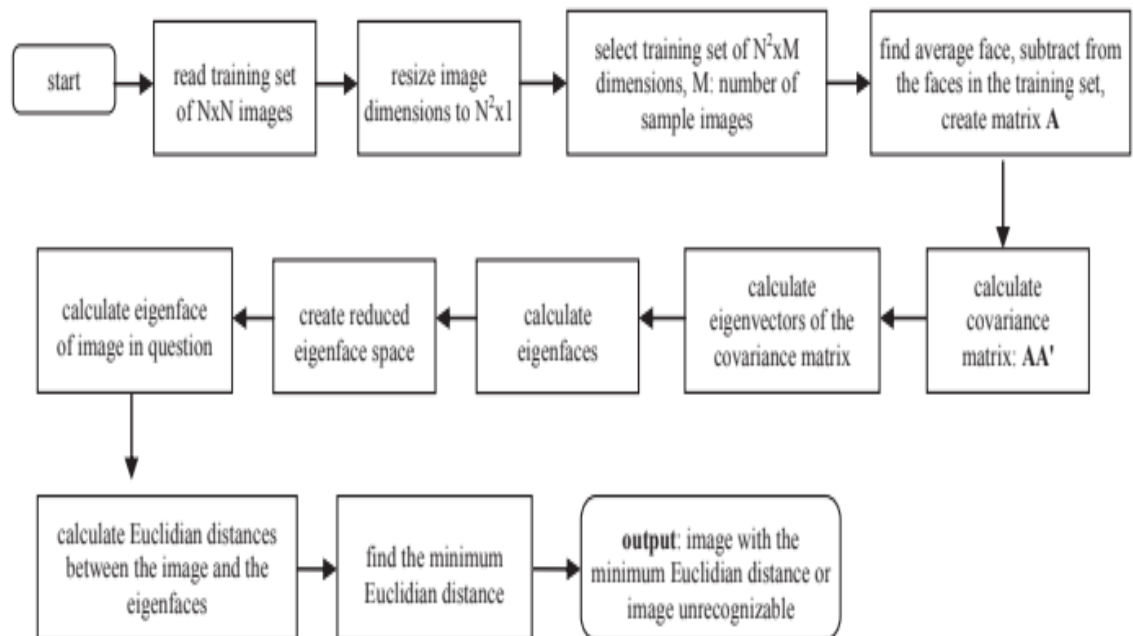


Fig.1 Flowchart of Eigenfaces method

3.)

Title- "Face Recognition Using Fisherface Method"

Author- "Mustamin Anggo1 and La Arapu1"

Date- 2018

Published In- “2nd International Conference on Statistics, Mathematics, Teaching, and Research, Department of Mathematics, Universitas Cenderawasih, Jayapura, Indonesia.”

The 'fisherface' approach of face recognition is meant to recognize a face picture by comparing the results of its feature extraction with the results of the face picture being recognized. According to its design, the system should be capable of determining whether or not the image to be examined has been correctly identified [8]. In this study, up to 200 facial images in "*.bmp format" of 50 students were used, with the maximum number of photographs being 200. Each of the children has five various expressions on their faces. An automated face recognition system that employs 'fisherface' approaches is capable of accurately detecting the picture of face screening with 100 percent accuracy whenever the test image is from the training sample, but only with 93 percent accuracy when the input image is dissimilar from the training set. In order to complete an introduction to face identification using the 'fisherface' approach, it is necessary to test face pictures with different colored components of the input pattern and a copy of the source image, as well as to complete an introductory to face identification using the 'fisherface' approach. The fading effect and noise-induced photos are also resistant to this kind of processing. The majority of failed identification images are influenced by a variety of characteristics, including scale issues and poses, among others. In order to ease the first component, enhanced picture scaling may be employed; nevertheless, the problem associated with posture might be solved by giving extra training photographs in various situations.

4.)

Title- “Estimation of ‘age, gender and ethnicity’ using CNN-based face recognition algorithm”

Author- ‘Sooyeon Lim’

Date- 9, June, 2020

Published In- School of Game, Dongyang University, Korea, International Journal of Advanced Smart Convergence Vol.9 No.2 203-211 (2020)

This work is a suggestion for using deep learning-based modeling to identify the ‘age, gender and ethnicity’ of a realistic facial picture in response to changes in the external environment.

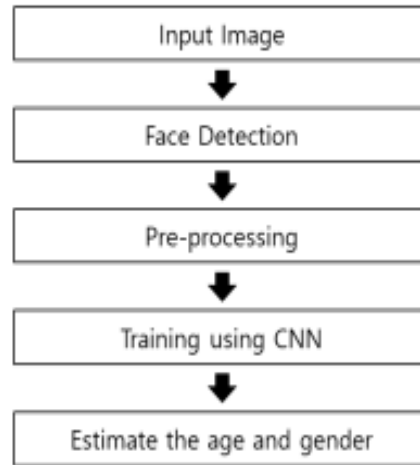


Fig.2 Block Diagram of CNN method for prediction

A face recognition algorithm based on CNN-based facial recognition is presented in the study effort. As a result, the proposed algorithm's validity may be improved by examining how the updated CNN network topology and learning process are implemented. Using this strategy, we may say that the findings are satisfactory. Even with a simple CNN architecture, mature male/female categorization is reasonably easy and shows great performance, but it is more difficult to identify the more precise groupings depending on age. While classifying six categories, there was a significant propensity for children, women, and elderly women in this research to make incorrect classifications. There must be fresh data sets, as well as period commencing algorithms and artificial neural, to improve this. Researchers believe that real age recognition might be achieved in the future if they do more study into alternative system designs and activation functions.

5.)

Title- “Comparing classical and deep approaches for face recognition in a smart-gym application”

Author- ‘Gonzalo Benito’

Date- September, 2017

Published In- Universitat de Barcelona and Computer Vision Center

Author compared various face detection models that could be used for the problem statement. Majorly, Eigenface, Fisherface and CNN were compared and the outcome was that out of three CNN gave best performance [9].

The Eigenface method had the following drawbacks:

- Firstly, the approach is scale-sensitive, necessitating additional preprocessing to normalize the scale.
- Second, its identification rate drops with changing stance and lighting.
- Moreover, the eigenface technique is not resilient when working with large fluctuations in stance, emotion, and mask.
- Finally, the approach may necessitate a consistent background, that isn't always possible in natural landscapes.

The eigenfaces have been surpassed by more superior, more resilient techniques. As a result, it is now only utilized as an introduction approach.

When it comes to calculating the projection of image space, the 'Fisherface' is more difficult than the 'Eigenface.' It takes a long time to calculate the relationship between between-class scatter and within-class scatter. Furthermore, due to the requirement for improved categorization, the depth of representation in face space was not as compact as it was in Eigenface, resulting in a larger amount of face storage and a longer processing time for recognition. Because there was a greater amount of data in the first phase, Fisherfaces performed better. In contrast, its behavior altered throughout the second part of the experiment. This algorithm was typically the more difficult of the two processes to implement. In addition, it was quite imprecise in the second phase. So far, CNN's algorithm has proven to be the most effective. When it came to the first phase, it properly recognized all of the photographs; when it came to phase 2, it correctly identified 5 and 10 items. When it came to the second segment, which consisted of 15 items, there were just a few typos. These errors were minimized as the amount of training data increased. In comparison to

the other participants who had trouble identifying the photos, CNN was only moderately influenced by the change in the environment in which they were working.

From these data, we can conclude that employing a convolutional neural network for facial recognition software is more efficient than using a statistical tool or looking for trends. As previously established, a CNN does not have a set structure and may have its structure altered at any time [10].

CHAPTER-03: SYSTEM DEVELOPMENT

3.1 Analysis/Design/Development/Algorithm

Using a framework for ‘age, gender and ethnicity’ detection, this project aims to provide a webapp that links gender, age, and picture inputs and validates every one of them according on their connection. For example, the gender given by the customer will be evaluated to the gender determined from the user picture, for example. In addition, the user photo's age range will be examined depending on the user's stated age. The basic steps may be used to perform this task:

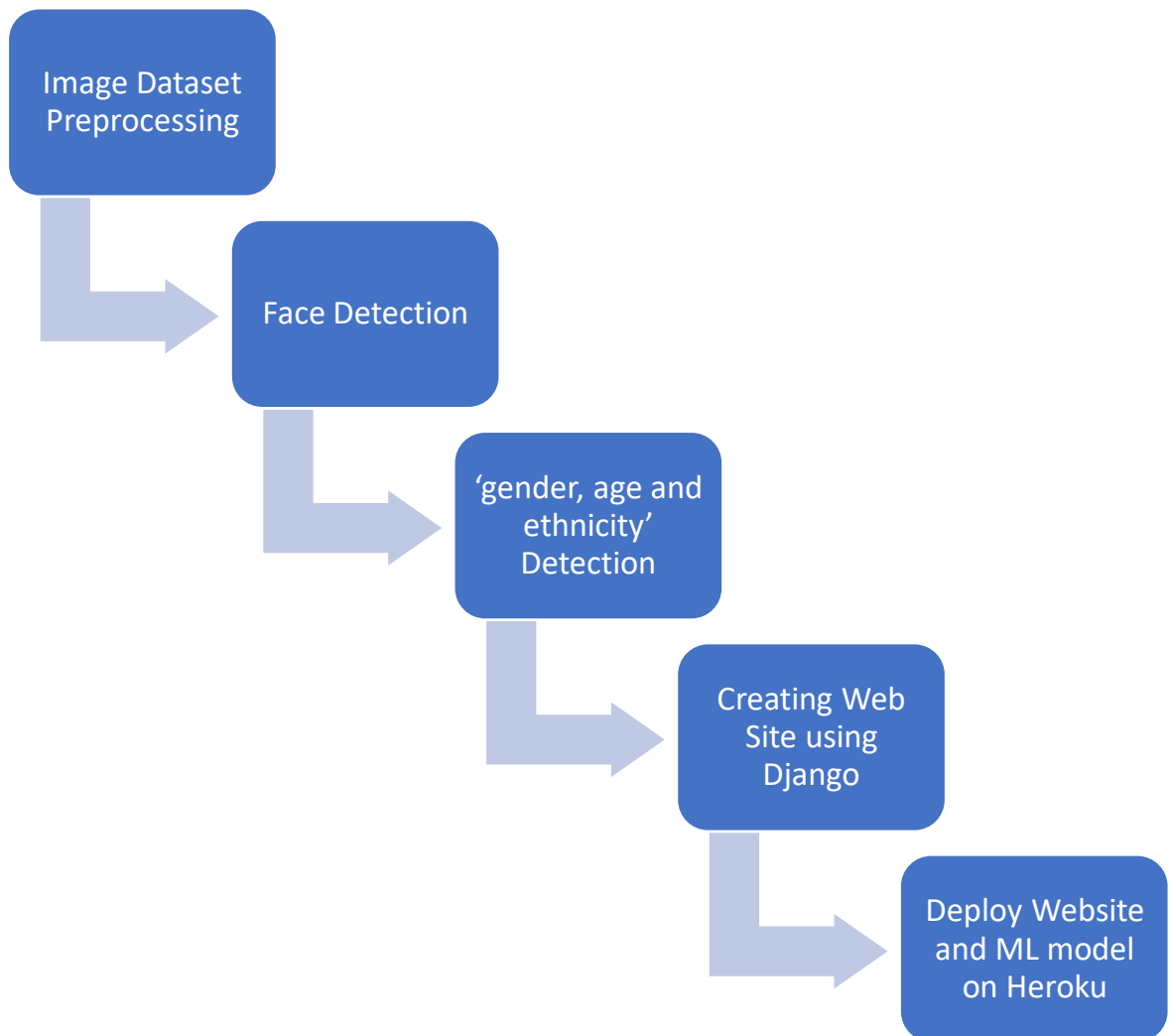


Fig.3 Development and process flow

A. Face detection

CNN model was chosen because it outperformed other approaches (Haar Cascade, Eigenface, and Fisherface) in terms of accuracy, quickness, operating in real-time upon CPU, and detecting faces of varied sizes and alignments. The picture was first extracted from the dataset and transformed to grayscale. The opencv cascade classifier was then utilized, and the pixels were sent through the opencv pretrained model ("haarcascade_frontalface_alt2.xml"), which generated the corner points for the rectangle where the face was detected. Finally, the face was stored into a separate folder for future processing. The photos were then run through a size filter function, which excluded outlier images (less than 5kb in size), and the final image produced was utilized as the dataset for model training.

B. Gender Prediction

'Gender prediction' is believed to be a 'classification issue', and the architecture's output layer uses a 'Sigmoid' function with one node representing Male or Female (0: Male, 1: Female). The model utilized is a convolution architecture with four layers, each of which includes a convolution layer and a maxpooling layer. A "relu" activation function is then applied to the output of these layers, followed by a dense layer. Gender-specific nodes are generated using thick layers with sigmoid functions. In terms of precision, this model was 88.09 percent accurate. The next model architecture's output layer uses a 'Sigmoid' function with two nodes representing two probabilities for Male and Female class respectively (index 0: Male, 1: Female). The model utilized is a convolution architecture with three layers, each of which includes a convolution layer, batch-normalization layer and a maxpooling layer. The output of these layers is sent to a flattening layer, which is then followed by a dense layer using the "relu" activation function. Finally, after dropping 50 percent random nodes, a dense layer with a sigmoid function is used to generate the node with gender probability for both classes. This model gave an accuracy of 89.37 percent. So, we used the second model for gender prediction and imported this model into storage and sending the result of the facial recognition process (identified face picture) through the 'gender prediction network' results in predicted gender variables as an output

from the cnn model. Now we can take the final output and use it to identify the sexual identity.

C. Age Prediction

Owing to the complexity of claiming “age prediction as a regression problem” since humans with ‘same age’ can look ‘entirely different’ relying on each person's genetic variation and it's difficult for the majority of people to accurately guess their age just by looking at them, but it's possible to guess a person's age range. As a result, we treated it as a classification issue by employing the UTKFace dataset. The model utilized is a convolution model with four layers, each of which includes a convolution layer and a maxpooling layer. The output of these layers is sent to a flattening layer, which is then followed by a dense layer using the ‘relu activation function.’ Finally, a thick layer using the 'relu' function is used to produce the node with gender. This model gave us low accuracy due to non-grouping of age ranges. The next model architecture's output layer uses a 'Softmax' function with twenty-four nodes representing probabilities for different age classes respectively. The model utilized is a convolution architecture with three layers, each of which includes a convolution layer, batch-normalization layer and a maxpooling layer. The output of these layers is sent to a flattening layer, which is then followed by a dense layer using the "relu" activation function. Finally, after dropping 50 percent random nodes, a dense layer with a softmax function is used to generate the node with probability for age classes. This model gave us average accuracy and after testing on various images from dataset and google we decided that this accuracy was good for the dataset we used. So, after analysis of both model and testing we decided to move on with the second model for age prediction as well. This model can now be used to forecast the range and apply it as a projected age group once it is imported into storage and the facial recognition software process (detected face photograph) is sent via the age prediction network.

D. Ethnicity Prediction

‘Ethnicity Prediction’ is treated as a classification issue by employing the UTKFace dataset. The model utilized is a convolution model with four layers, each of which includes a convolution layer and a maxpooling layer. The output of these layers is sent to a flattening

layer, which is then followed by a dense layer using the 'relu activation function.' Finally, a thick layer using the 'relu' function is used to produce the node with ethnicity. In terms of precision, this model was 50.11 percent accurate. The next model architecture's output layer uses a 'Softmax' function with five nodes representing five probabilities for five ethnicity classes respectively. The model utilized is a convolution architecture with three layers, each of which includes a convolution layer, batch-normalization layer and a maxpooling layer. The output of these layers is sent to a flattening layer, which is then followed by a dense layer using the "relu" activation function. Finally, after dropping 50 percent random nodes, a dense layer with a softmax function is used to generate the node with ethnicity probability for classes. This model gave an accuracy of 79.50 percent. So, we used the second model for ethnicity prediction and imported this model into storage and sending the result of the facial recognition process (identified face picture) through the 'gender prediction network' results in predicted ethnicity variables as an output from the cnn model. Now we can take the final output and use it to identify the ethnicity identity.

E. Web App

A web app was designed to handle the model's 'age, gender and ethnicity' range validation needs. In addition, Python provides the vast majority of necessary libraries for ML and face detection. So, Django has been selected as the web service implementation framework. It's clear from the diagram below that the web-app has four stages: The first stage collects and validates the user's input (the user's photo, 'age, gender and ethnicity'), and the second and third endpoints are responsible for determining the user's 'age, gender and ethnicity' based on the user picture. 'Age, gender and ethnicity' class may then be predicted using a previously built model that uses facial detection to identify a person's face. Finally, the user's 'age, gender and ethnicity' are input, and the anticipated 'age, gender and ethnicity' are shown in the final result. This live webapp is hosted on 'Heroku' and accessible via link "agmodel.herokuapp.com".

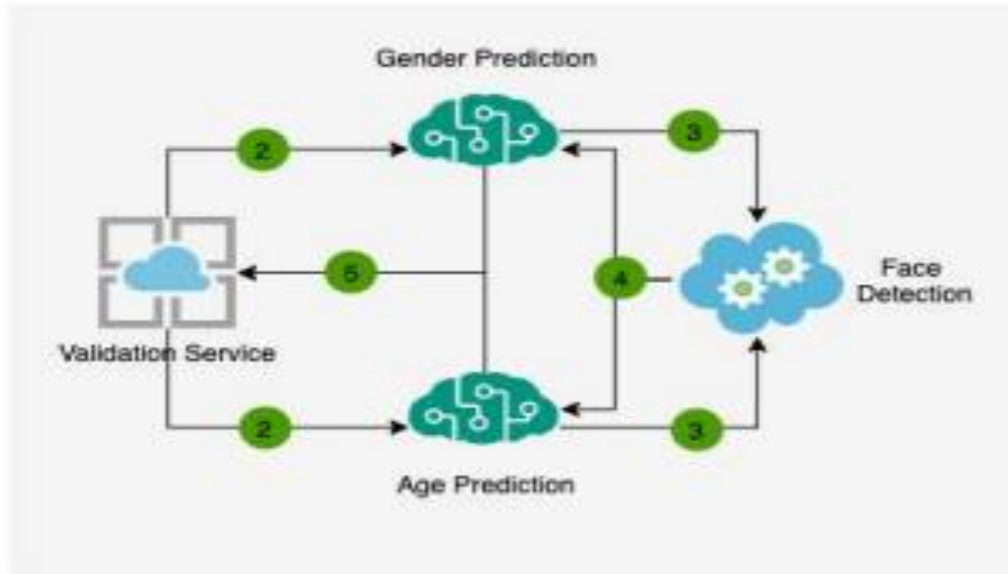


Fig. 1. Shows a web service architecture and the endpoints communication

Fig.4 Architecture showing web service and endpoints

3.2 Model Development

1.) Dataset used-

The model was trained and tested on the UTK Face dataset. With a wide variety of ages, the UTKFace dataset is an extensive face collection (range from 0 to 116 years old). More than 20,000 face photos are included in the collection, which includes age, gender, and ethnicity annotations. Pose, emotion, lighting, obscuration and resolution are all included in the photographs. Face identification, age estimation, age progression and regression, landmark location, and many more applications might benefit from this dataset. The following are a few examples of what you might expect to see:

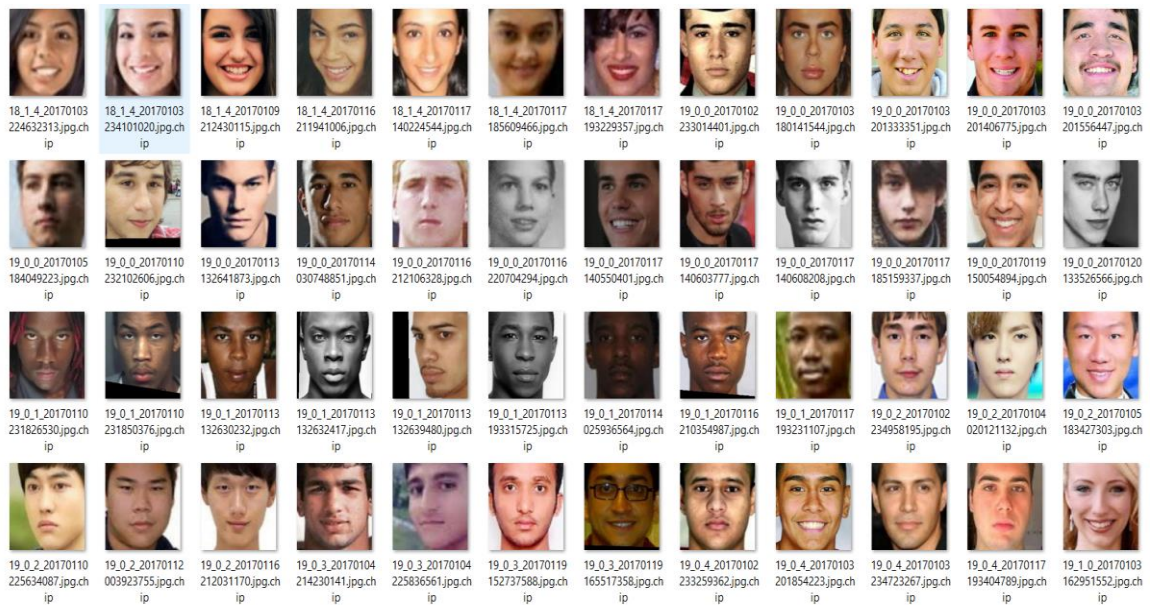


Fig.5 UTKFace Dataset

The collection contains 23708 face pictures in the wild (just one face in one image), which are tagged by age, gender, and ethnicity. It is contained in the file name of each face photograph as [age] [gender] [race] [date&time].jpg.

- Integers ranging from zero to one hundred sixteen indicate an individual's age.
- One of two possibilities exists for [gender] (0: male, 1: female)
- There are four races: White, Black, Asian, Indian, and Others, each of which has a race number between 0 and 4. (like Hispanic, Latino, Middle Eastern).
- According to the [date&time] format, the date and time that a picture was gathered to UTKFace is shown in yyyyymmddHHMMSSFFF [age] is an integer from 0 to 116, indicating the age

2.) Preprocessing of dataset-

All of the photos in the dataset were first run through a face identification model that had been pre-trained. The Opencv cascade classifier ('haarcascade_frontalface_alt2.xml') was utilized for this purpose. Images were saved in a new folder and ran through a filter to exclude outliers that included incorrectly identified faces. Processing was done in the final picture folder. Finally, photos were scanned using opencv and the pixels were retrieved into an array and 'age, gender and ethnicity' were taken from the filename and placed in a

dataframe. Step 2 was a continuation of this process. The age array was mapped to different bins and each age was associated with an age class. A total of 24 age classes were created and the data was sent for further processing.

3.) Visualization of dataset and creating training and testing data-

Various bargraphs, countplots, and line graphs were used to examine the data's consistency. The opencv imshow function was used to verify for accurate pixel alignment. Importing the necessary tensorflow and keras libraries and preparing training and testing data for 'age, gender and ethnicity'.

4.) Convolution neural network-

In the field of artificial intelligence, CNNs are a kind of 'Deep Neural Network' (DNN) that can identify and categorize certain aspects in pictures and are commonly used to analyze visual data. Its uses span from video and picture identification to image classification, image analysis in medicine, and computer vision and natural language processing. When two functions are multiplied together, the result is a third function that reflects how the form of one function is altered by the other. Multiplying two matrices-representable photos produces an output from which features from the image may be extracted.

Feature Extraction is a convolution technique that separates and identifies the image's distinct features for analysis. Convolution output is fed into a fully connected layer that uses the information gathered in earlier rounds to forecast the image's class. The CNN has three layers: "convolutional, pooling, and fully-connected (FC) layers." When these layers are put together, a CNN will be formed.

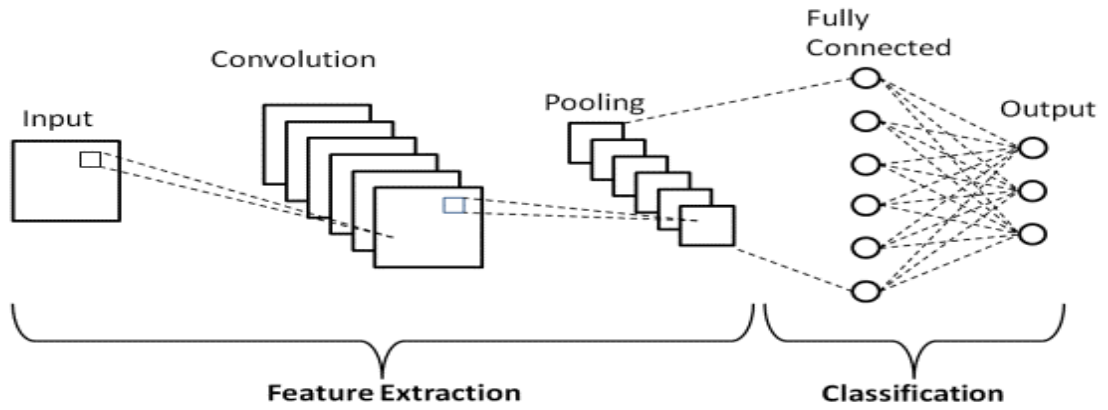


Fig.6 CNN Architecture

A) Convolution layer- The input pictures are initially processed via a convolutional layer, which is utilized to extract different characteristics. The input picture is processed by this layer, which extracts characteristics from it. This layer contains the picture matrix and the kernel/filter that extracts the image's features. It is possible to get an image feature map of dimension $(a-(fh+1), b-(fw+1), 1)$ from an image matrix of volume (a) . In this layer, the convolution of the input picture with a filter of a certain size is carried out. Dot products are calculated by multiplying the input image's pixels by the filter's pixel size. We get information about the image's corners and edges from this layer's output, which is called a Feature map. Afterwards, this 'feature map' is supplied to further layers in order to explore more about the input picture.

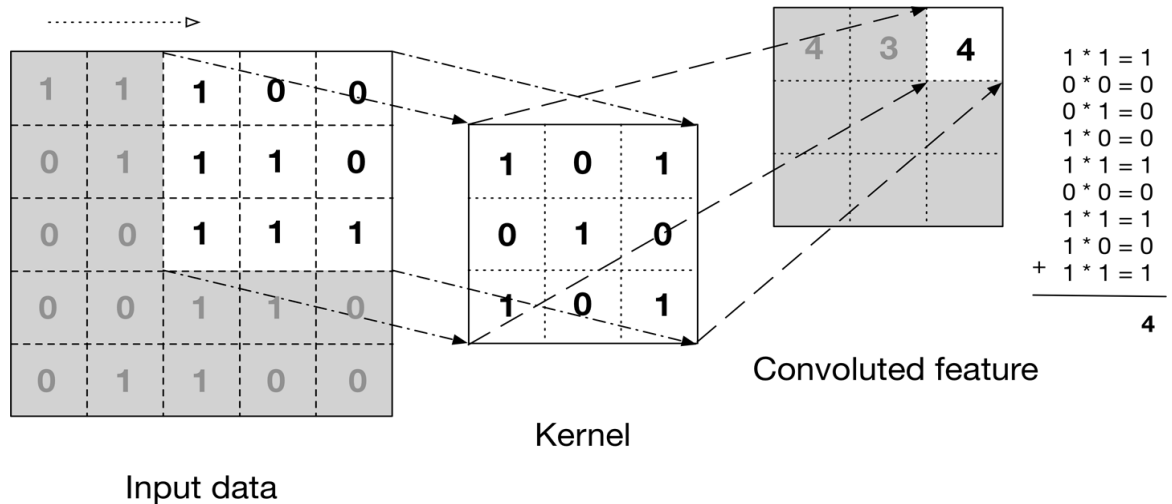


Fig.7 Convolutional Layer

B) Pooling layer- In order to decrease the algorithm's computing costs, this layer aims to reduce the size of the 'convolved feature map.' When the photos, like those we used in the project, are too huge, the pooling layer decreases the number of parameters or blocks in the feature matrix.

Max Pooling: the filter area is pooled to its fullest extent.

Average Pooling: pools the average of the elements.

Sum Pooling: The total of the components in the filter is gathered in a pool called the sum pooling.

The pooled feature map was created using Max Pooling for this project. The biggest element on the feature map is used in Max Pooling.

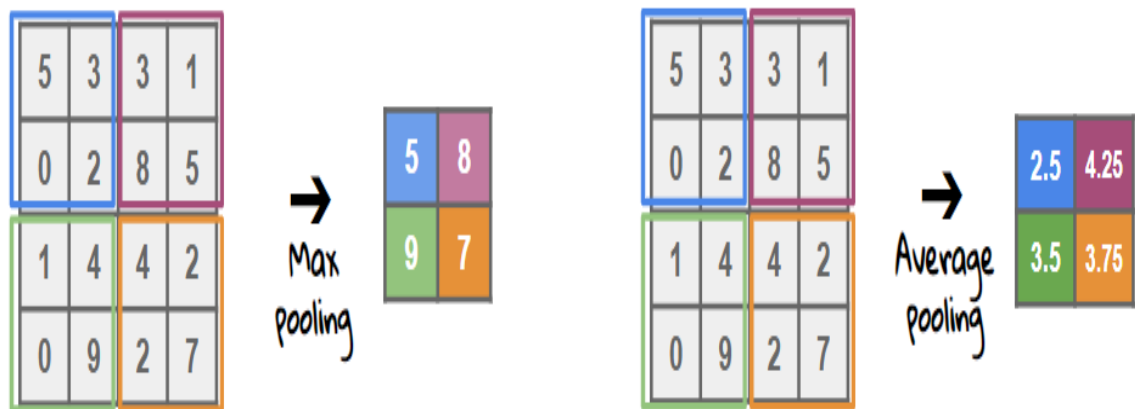


Fig.8 Pooling Layer

C) Flattening layer- Flattening is the process of transforming a dataset into a one-dimensional array that may be used as input to the following layer. Convolutional layers' output is flattened into a single feature vector. Also known as a fully connected layer, it is linked to the final classification model.

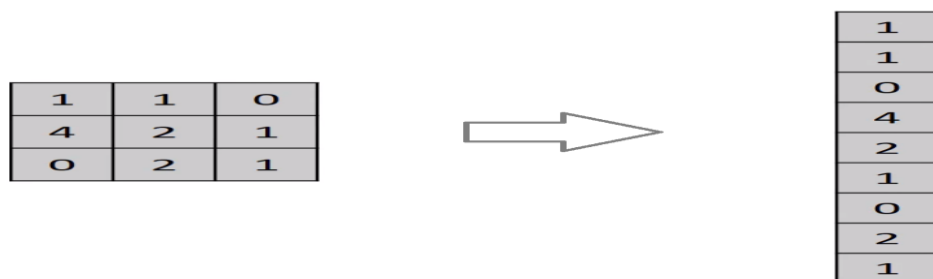


Fig.9 Flattening Layer

- D) Fully Connected Layer- The Fully Connected (FC) layer is utilized to link neurons between multiple layers and comprises of weights and biases along with neurons. It is common practice to position the last few layers of a CNN architecture before the output layer.
- E) The dropout layer and activation function are the two additional factors that must be taken into consideration in addition to these three layers.
- Dropout- Overfitting may be avoided by using a dropout layer, which removes some neurons from the neural network during training, resulting in a smaller model. The neural network loses 30% of its nodes when it reaches a dropout of 0.3.
 - Activation Functions- It is possible to learn and estimate any form of continuous and complicated connection between variables in a network using an Activation Function, which is a function of the network's variables. To put it another way, this function determines which information should be transported to the output and which information should not. The activation functions that we utilized were as follows:
 - ReLU (Rectified Linear Unit): When utilized in our model, it is one of the most often used activation functions since it ensures non-linearity by changing all negative values in the matrix to 0 while leaving all positive numbers as they are.

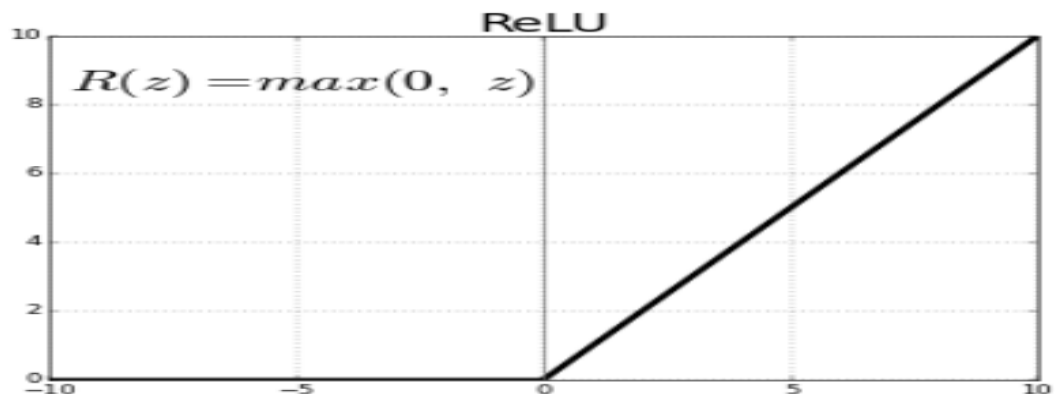


Fig.10 Relu function

- Sigmoid function- The major reason we pick the sigmoid function is because it occurs between and inside the interval (0 to 1). Models in which we need to anticipate probability as an output benefit greatly from it. The sigmoid function is the best choice

since the probability of anything occurs with between '0 and 1.' The function may be distinguished. The slope of the sigmoid curve may be determined at any two points on the curve. The function itself is 'monotonic', but the derivative is not. When training neural networks, the logistic sigmoid function might lead them to get stuck.

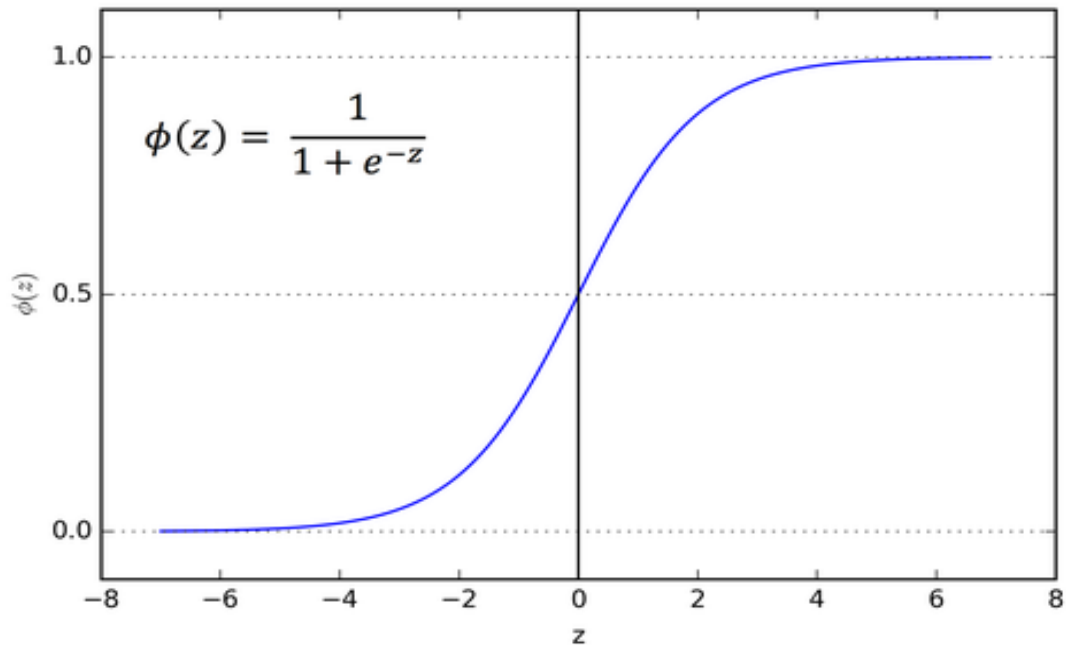


Fig.11 Sigmoid function

- Softmax function- This is an activation function that turns a vector of integers into a vector of likelihood. Our model is designed to produce 24 values corresponding to various age classes. The Softmax function normalizes the outputs and changes them from weighted sum values to probability values adding to one. The Softmax function is depicted in the diagram below.

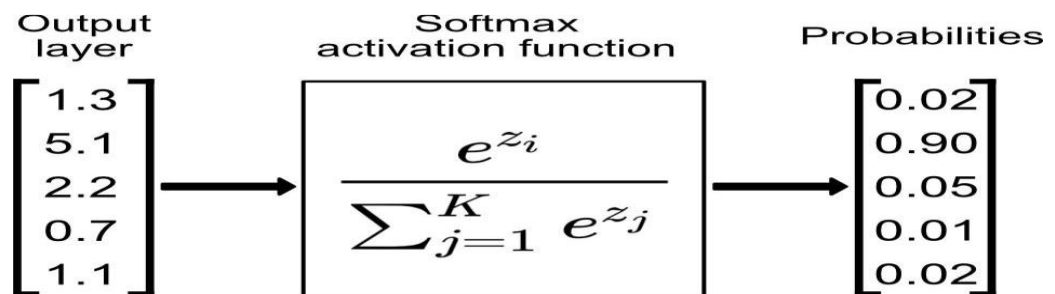


Fig.12 Softmax function

5.) Creating CNN model for 'age, gender and ethnicity' prediction-

CNN Architecture-(Model 1)

Table.1 CNN Model-1

Layer	Details
Convolution Layer	Input Shape- "(48,48,3), 32 filters of size (3x3), Dropout of 0.1, Activation function 'relu'" Output- "32 tensors of size (48,48,32)"
Max Pooling Layer	Input Shape- "(48,48,32), Pool size (2x2)" Output Shape- "(24,24,32)"
Convolution Layer	Input Shape- "(24,24,32), 64 filters of size (3x3), Dropout of 0.1, Activation function 'relu'" Output- "64 tensors of size (24,24,64)"
Max Pooling Layer	Input Shape- "(24,24,64), Pool size (2x2)" Output Shape- "(12,12,64)"
Convolution Layer	Input Shape- "(12,12,64), 128 filters of size (3x3), Dropout of 0.1, Activation function 'relu'" Output- "128 tensors of size (12,12,128)"
Max Pooling Layer	Input Shape- "(12,12,128), Pool size (2x2)" Output Shape- "(6,6,128)"
Convolution Layer	Input Shape- "(6,6,128), 256 filters of size (3x3), Dropout of 0.1, Activation function 'relu'" Output- "256 tensors of size (6,6,256)"
Max Pooling Layer	Input Shape- "(6,6,256), Pool size (2x2)" Output Shape- "(3,3,256)"

Flattening Layer	Input Shape-(3,3,256) Output Shape- “1d array of size 2304”
For Gender	
➤ Dense Layer	Input Shape- “1d array 2304, Units-64, Activation ‘relu’” Output Shape- “64 units array”
➤ Dropout	Dropout of 0.2
➤ Dense Layer	Input Shape- “64 units array, Units-1, Activation ‘sigmoid’” Output Shape- “1 single value (0 or 1)”
For Age	
➤ Dense Layer	Input Shape- “1d array 2304, Units-64, Activation ‘relu’” Output Shape- “64 units array”
➤ Dropout	Dropout of 0.2
➤ Dense Layer	Input Shape- “64 units array, Units-1, Activation ‘relu’” Output Shape- “1 single value (predicted age)”
For Ethnicity	
Dense Layer	Input Shape- “1d array 2304, Units-64, Activation ‘relu’” Output Shape- “64 units array”
Dropout	Dropout of 0.2
Dense Layer	Input Shape- “64 units array, Units-1, Activation ‘relu’” Output Shape- “1 single value (predicted ethnicity)”

CNN Architecture-(Model 2)

Table.2 CNN Model-2

Layer	Details
Convolution Layer	Input Shape- "(48,48,3), 32 filters of size (3x3), Activation function 'relu'" Output- "32 tensors of size (48,48,32)"
Batch Normalization	Input Shape- "(48,48,32)" Output Shape- "(48,48,32)"
Max Pooling Layer	Input Shape- "(48,48,32), Pool size (2x2)" Output Shape- "(24,24,32)"
Convolution Layer	Input Shape- "(24,24,32), 64 filters of size (3x3, Activation function 'relu'" Output- "64 tensors of size (24,24,64)"
Batch Normalization	Input Shape- "(24,24,64)" Output Shape- "(24,24,64)"
Max Pooling Layer	Input Shape- "(24,24,64), Pool size (2x2)" Output Shape- "(12,12,64)"
Convolution Layer	Input Shape- "(12,12,64), 64 filters of size (3x3, Activation function 'relu'" Output- "64 tensors of size (12,12,64)"
Max Pooling Layer	Input Shape- "(12,12,64), Pool size (2x2)" Output Shape- "(6,6,64)"
Flattening Layer	Input Shape- "(6,6,64)" Output Shape- "1d array of size 2304"
For Gender	
➤ Dense Layer	Input Shape- "1d array 2304, Units-256, Activation 'relu'" Output Shape- "256 units array"

➤ Dropout	Dropout of 0.5
➤ Dense Layer	Input Shape- “256 units array, Units-2, Activation ‘sigmoid’” Output Shape- “2 values denoting probability for gender classes”
For Age	
➤ Dense Layer	Input Shape- “1d array 2304, Units-256, Activation ‘relu’” Output Shape- “256 units array”
➤ Dropout	Dropout of 0.5
➤ Dense Layer	Input Shape- “256 units array, Units-24, Activation ‘softmax’” Output Shape- “24 values (probability of predicted age classes)”
For Ethnicity	
➤ Dense Layer	Input Shape- “1d array 2304, Units-256, Activation ‘relu’” Output Shape- “256 units array”
➤ Dropout	Dropout of 0.5
➤ Dense Layer	Input Shape- “256 units array, Units-2, Activation ‘softmax’” Output Shape- “5 values denoting probability for ethnicity classes”

6.) Creating Website using Django-

After installing Django and all of its dependencies, we were able to build the website. A Django model was constructed for the user to upload images, and the ml model that was developed in the previous phases was imported into it. Created were views, urls, and

templates; after that, the migrations were completed, and the website was tested on the local end.

7.) Deploy the model on Heroku platform-

The website, which was built using Django and included the CNN model, was packaged and prepared for deployment on the Heroku cloud computing platform. The Heroku CLI was used to deploy the webapp, which was successfully completed and is now accessible at “agmodel.herokuapp.com”.

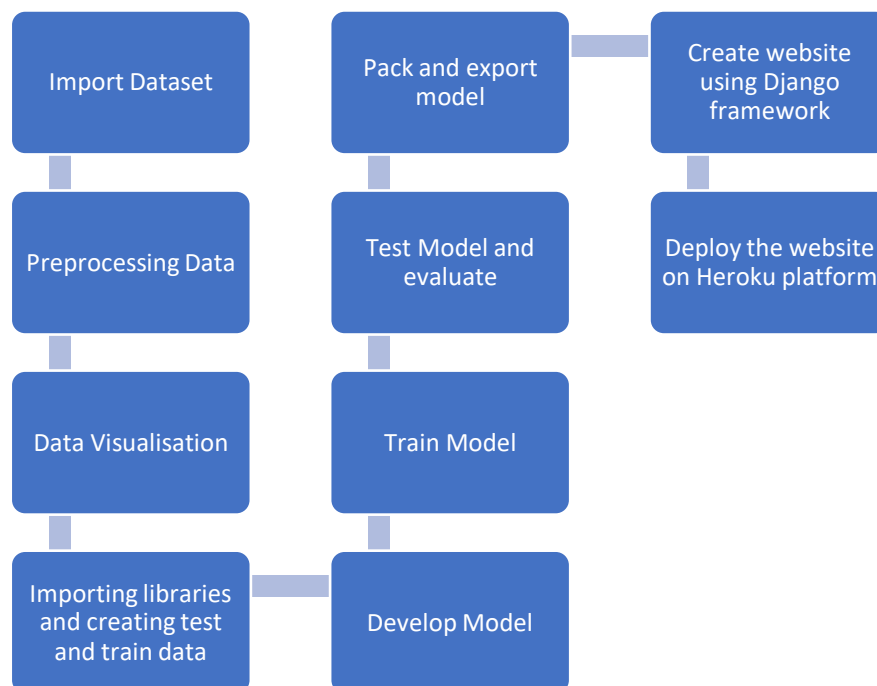


Fig.13 Flowchart of Process

CHAPTER-04: PERFORMANCE ANALYSIS

4.1 Analysis of system developed

Table.3 Analysis of system developed

<i>Model-1</i>	
Age Accuracy	4.59
Gender Accuracy	88.09
Ethnicity Accuracy	51.97
Age Loss	6.47 Loss calculation function- Mean Absolute Error (MAE)
Gender Loss	0.26 Loss calculation function- Binary Cross Entropy
Ethnicity Loss	0.63 Loss calculation function- Categorical Cross Entropy
Remarks	The accuracy was age was less due to the fact that ages were not divided into classes.

<i>Model-2</i>	
Age Accuracy	40.00
Gender Accuracy	89.37
Ethnicity Accuracy	79.50
Age Loss	1.68 Loss calculation function- Categorical Cross Entropy
Gender Loss	0.25 Loss calculation function- Binary Cross Entropy
Ethnicity Loss	0.60 Loss calculation function- Categorical Cross Entropy
Remarks	The accuracy was good in this case as age was divided into 24 bins with an age gap of 5 years and thus after testing the result for various inputs/outputs it can be concluded that with decrease in number of bins or increase in age gap in a bin accuracy increases but the model becomes less realistic.

4.2 Output at various stages

1) Data Preprocessing and Visualization-

```
# Load the cascade
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_alt2.xml')
path='C:\\Users\\thaku\\Desktop\\Major Project\\Major Final\\Age-Gender-Dete
path2='C:\\Users\\thaku\\Desktop\\Major Project\\Major Final\\Age-Gender-Dete
for ig in os.listdir(path):
    total=path+ig
    print(total)
    # Read the input image
    img = cv2.imread(total)
    # Convert into grayscale
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    # Detect faces
    faces = face_cascade.detectMultiScale(gray, 1.1, 4)
    # Draw rectangle around the faces and crop the faces
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x+w, y+h), (0, 0, 255), 2)
        faces = img[y:y + h, x:x + w]
    cv2.imshow("face", faces)
    cv2.imwrite(path2+ig, faces)
##
```

Fig.14 Face Detection

```
[1123 482 289 273 196 131 139 263 166 156 65 130 81 157
 177 247 158 262 98 284 346 395 426 859 734 2197 615 918
 570 724 350 664 143 409 880 483 293 325 266 526 132 266
 157 100 440 153 170 153 148 381 138 232 241 353 268 236
 97 271 82 293 161 125 103 50 259 77 94 100 56 147
 33 98 63 32 148 58 28 69 23 133 22 40 18 24
 155 35 10 34 33 82 2 13 5 9 17 9 11 2
 1 5 5 1 3 4]
```

Fig.15 Count for unique ages

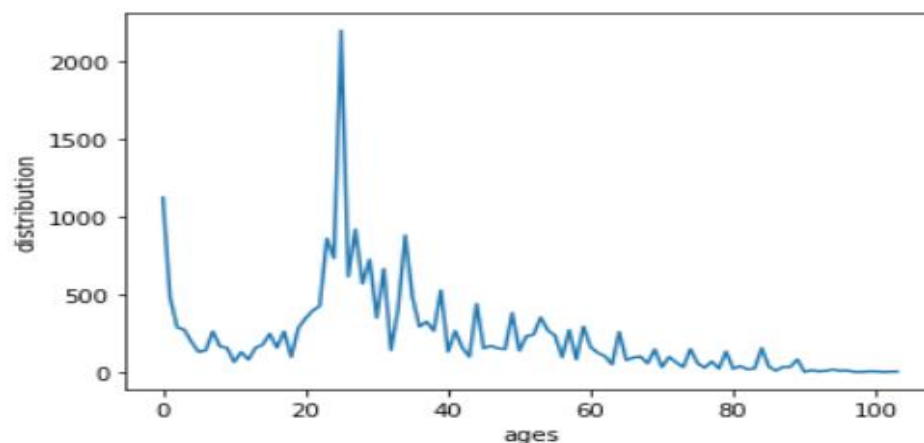


Fig.16 Age Distribution

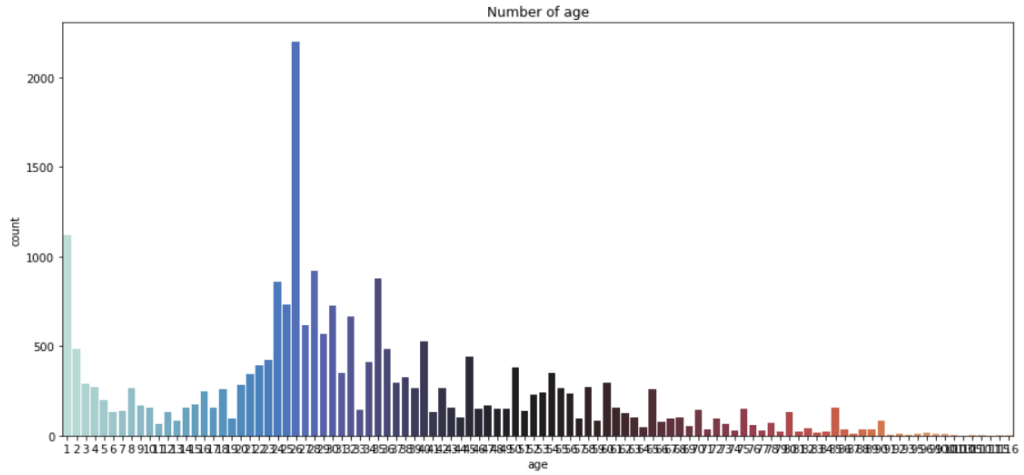


Fig.17 Age Distribution Bar Graph

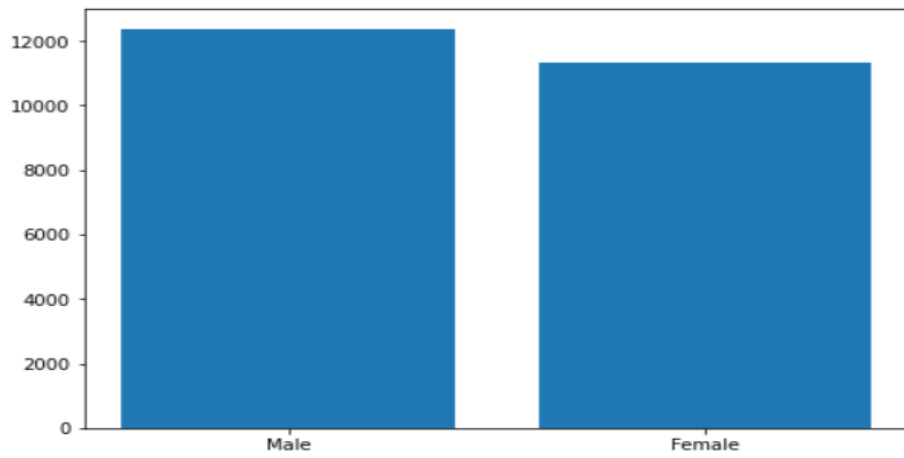


Fig.18 Gender Distribution Bar Graph

```
y["age"] = pd.cut(y["age"],bins=[0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100,105,110,115,120],labels=["0","1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20","21","22","23"])
```

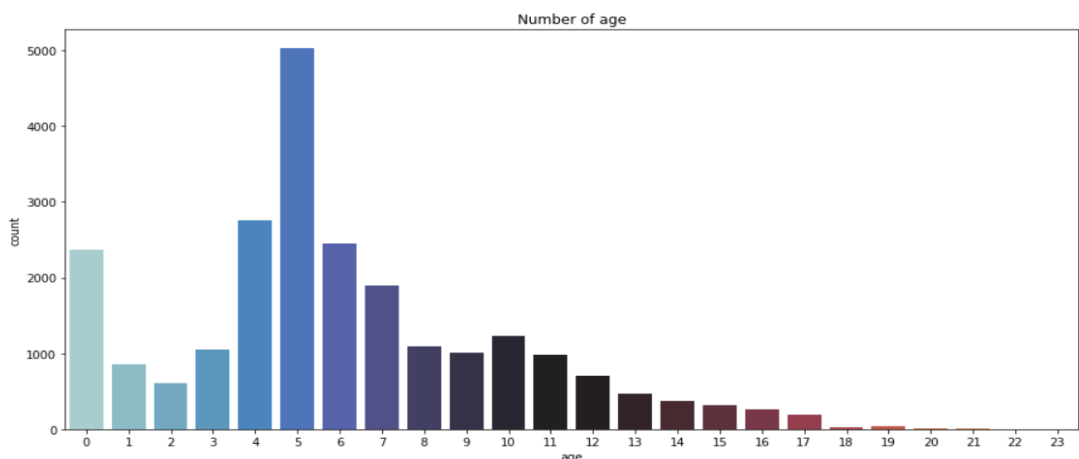


Fig19 Age Bins Bar Graph

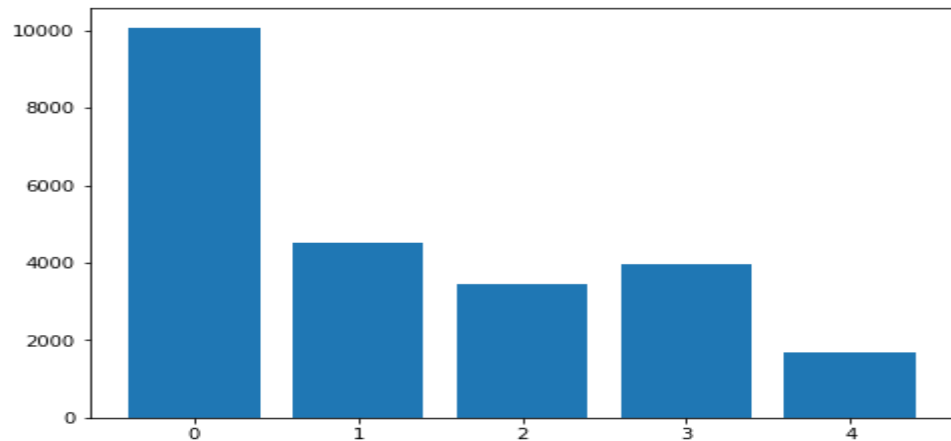


Fig.20 Ethnicity Distribution Bar Graph

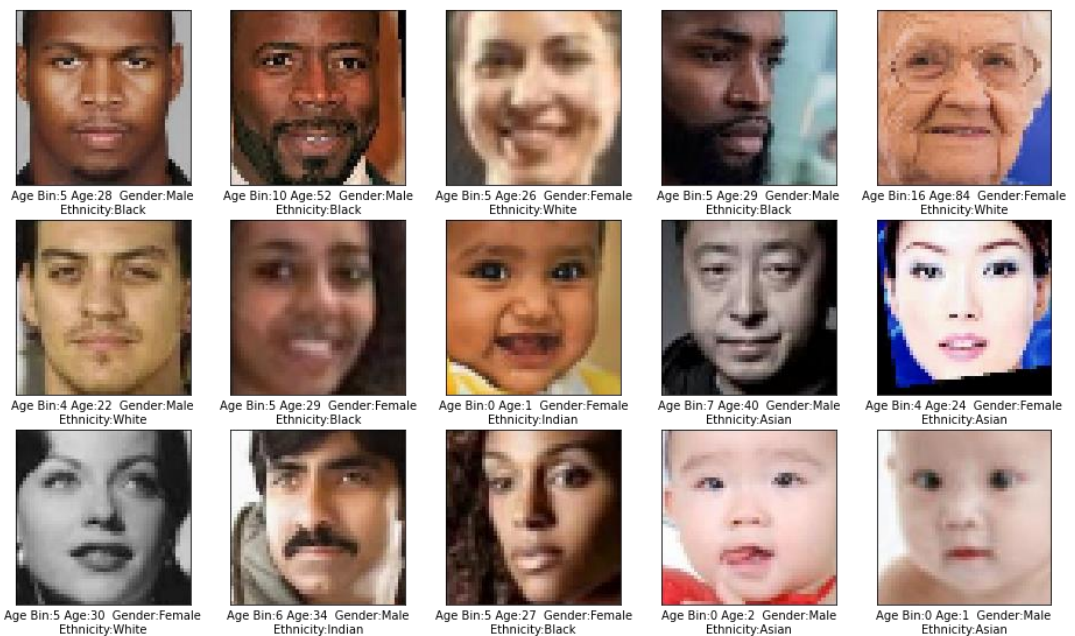


Fig.21 Sample images with age, gender, bin

2) Creating testing and training data-

```
age_m = np.array(y["age"])
gender_m = np.array(genders_f)
age = to_categorical(age_m, num_classes = 24)
gender = to_categorical(genders_f, num_classes = 2)
print("Age",age)
print("Gender", gender)
```

Fig.22 Converting Variables to Categorical


```

from sklearn.model_selection import train_test_split
X_train_gender, X_test_gender, y_train_gender, y_test_gender = train_test_split(images_f2, gender, test_size= 0.25)
X_train_age, X_test_age, y_train_age, y_test_age = train_test_split(images_f2, age, test_size= 0.25)

```

Fig.23 Creating test and train data

3) Developed Models-

Model: "model_7"

Layer (type)	Output Shape	Param #	Connected to
input_8 (InputLayer)	[(None, 48, 48, 3)]	0	[]
conv2d_28 (Conv2D)	(None, 48, 48, 32)	896	['input_8[0][0]']
dropout_28 (Dropout)	(None, 48, 48, 32)	0	['conv2d_28[0][0]']
activation_28 (Activation)	(None, 48, 48, 32)	0	['dropout_28[0][0]']
max_pooling2d_28 (MaxPooling2D)	(None, 24, 24, 32)	0	['activation_28[0][0]']
conv2d_29 (Conv2D)	(None, 24, 24, 64)	18496	['max_pooling2d_28[0][0]']
dropout_29 (Dropout)	(None, 24, 24, 64)	0	['conv2d_29[0][0]']
activation_29 (Activation)	(None, 24, 24, 64)	0	['dropout_29[0][0]']
max_pooling2d_29 (MaxPooling2D)	(None, 12, 12, 64)	0	['activation_29[0][0]']
conv2d_30 (Conv2D)	(None, 12, 12, 128)	73856	['max_pooling2d_29[0][0]']
dropout_30 (Dropout)	(None, 12, 12, 128)	0	['conv2d_30[0][0]']
activation_30 (Activation)	(None, 12, 12, 128)	0	['dropout_30[0][0]']
max_pooling2d_30 (MaxPooling2D)	(None, 6, 6, 128)	0	['activation_30[0][0]']
conv2d_31 (Conv2D)	(None, 6, 6, 256)	295168	['max_pooling2d_30[0][0]']
dropout_31 (Dropout)	(None, 6, 6, 256)	0	['conv2d_31[0][0]']
activation_31 (Activation)	(None, 6, 6, 256)	0	['dropout_31[0][0]']
max_pooling2d_31 (MaxPooling2D)	(None, 3, 3, 256)	0	['activation_31[0][0]']
flatten_7 (Flatten)	(None, 2304)	0	['max_pooling2d_31[0][0]']
dense_7 (Dense)	(None, 64)	147520	['flatten_7[0][0]']
dropout_32 (Dropout)	(None, 64)	0	['dense_7[0][0]']
sex_out (Dense)	(None, 1)	65	['dropout_32[0][0]']
age_out (Dense)	(None, 1)	65	['dropout_32[0][0]']
eth_out (Dense)	(None, 1)	65	['dropout_32[0][0]']

=====
 Total params: 536,131
 Trainable params: 536,131
 Non-trainable params: 0

Fig.24 Model-1 Summary

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 32)	896
batch_normalization (Batch Normalization)	(None, 48, 48, 32)	128
max_pooling2d (MaxPooling2D)	(None, 24, 24, 32)	0
conv2d_1 (Conv2D)	(None, 24, 24, 64)	18496
batch_normalization_1 (Batch Normalization)	(None, 24, 24, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 64)	0
conv2d_2 (Conv2D)	(None, 12, 12, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 64)	0
flatten (Flatten)	(None, 2304)	0
dense (Dense)	(None, 256)	590080
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 24)	6168

Fig.25 Model-2 Summary

4) Training Model-

```

History=Model.fit(X_train,Y_train_2,batch_size=64,validation_data=(X_test,Y_test_2),epochs=500,callbacks=[callback_list])
Epoch 152/500
277/278 [=====>.] - ETA: 0s - loss: 5.4993 - sex_out_loss: 0.1697 - age_out_loss: 4.3055 - sex_out_accuracy: 0.9231
Epoch 00152: val_loss did not improve from 7.32263
278/278 [=====] - 8s 30ms/step - loss: 5.4982 - sex_out_loss: 0.1701 - age_out_loss: 4.3042 - sex_out_accuracy: 0.
Epoch 153/500
278/278 [=====] - ETA: 0s - loss: 5.4473 - sex_out_loss: 0.1646 - age_out_loss: 4.2573 - sex_out_accuracy: 0.9266
Epoch 00153: val_loss did not improve from 7.32263
278/278 [=====] - 8s 30ms/step - loss: 5.4473 - sex_out_loss: 0.1646 - age_out_loss: 4.2573 - sex_out_accuracy: 0.
Epoch 154/500
278/278 [=====] - ETA: 0s - loss: 5.4890 - sex_out_loss: 0.1631 - age_out_loss: 4.3005 - sex_out_accuracy: 0.9272
Epoch 00154: val_loss did not improve from 7.32263
278/278 [=====] - 9s 31ms/step - loss: 5.4890 - sex_out_loss: 0.1631 - age_out_loss: 4.3005 - sex_out_accuracy: 0.
Epoch 155/500
277/278 [=====>.] - ETA: 0s - loss: 5.5521 - sex_out_loss: 0.1626 - age_out_loss: 4.3629 - sex_out_accuracy: 0.9259
Epoch 00155: val_loss did not improve from 7.32263
model_age = my_model(24,"softmax","categorical_crossentropy')
history_age = model_age.fit(X_train_age, y_train_age, batch_size=batch_size,
                           epochs = epochs, validation_data = (X_test_age,y_test_age), steps_per_epoch= X_train_age.shape[0] // batch_size, callbacks= [early_stoppi
Epoch 1/200
277/277 [=====] - 17s 19ms/step - loss: 2.2951 - accuracy: 0.2766 - val_loss: 3.0899 - val_accuracy: 0.0472 - lr: 0.0010
Epoch 2/200
277/277 [=====] - 5s 18ms/step - loss: 1.9604 - accuracy: 0.3305 - val_loss: 2.2778 - val_accuracy: 0.2797 - lr: 0.0010
Epoch 3/200
277/277 [=====] - 5s 18ms/step - loss: 1.8409 - accuracy: 0.3490 - val_loss: 1.8088 - val_accuracy: 0.3673 - lr: 0.0010
Epoch 4/200
277/277 [=====] - 5s 18ms/step - loss: 1.7475 - accuracy: 0.3654 - val_loss: 1.7206 - val_accuracy: 0.3774 - lr: 0.0010
Epoch 5/200
277/277 [=====] - 5s 18ms/step - loss: 1.6883 - accuracy: 0.3765 - val_loss: 1.7253 - val_accuracy: 0.3678 - lr: 0.0010
Epoch 6/200
277/277 [=====] - 5s 18ms/step - loss: 1.6234 - accuracy: 0.3940 - val_loss: 1.7939 - val_accuracy: 0.3597 - lr: 0.0010
Epoch 7/200
275/277 [=====>.] - ETA: 0s - loss: 1.5838 - accuracy: 0.4086

```

Fig.26 Training Model

5) Evaluate Model-

```

▶ Model.evaluate(X_test,Y_test2)
↳ 186/186 [=====] - 1s 5ms/step
  [8.24200439453125,
   0.29888781905174255,
   6.166737079620361,
   0.6320450305938721,
   0.8869580030441284,
   0.047916315495967865,
   0.5196558237075806]

```

Fig.27 Model-1 Evaluation

```

loss, acc = model_age.evaluate(X_test_age, y_test_age, verbose=0)
print('Test loss: {}'.format(loss))
print('Test Accuracy: {}'.format(acc))

```

```

Test loss: 1.6870245933532715
Test Accuracy: 0.39024803042411804

```

```

loss, acc = model_gender.evaluate(X_test_gender, y_test_gender, verbose=0)
print('Test loss: {}'.format(loss))
print('Test Accuracy: {}'.format(acc))

```

```

Test loss: 0.25247007608413696
Test Accuracy: 0.893706738948822

```

```

▶ loss, acc = model_ethnicity.evaluate(X_test_ethnicity, y_test_ethnicity, verbose=0)
  print('Test loss: {}'.format(loss))
  print('Test Accuracy: {}'.format(acc))
↳ Test loss: 0.6053451299667358
  Test Accuracy: 0.7950059175491333

```

Fig.28 Model-2 Evaluation

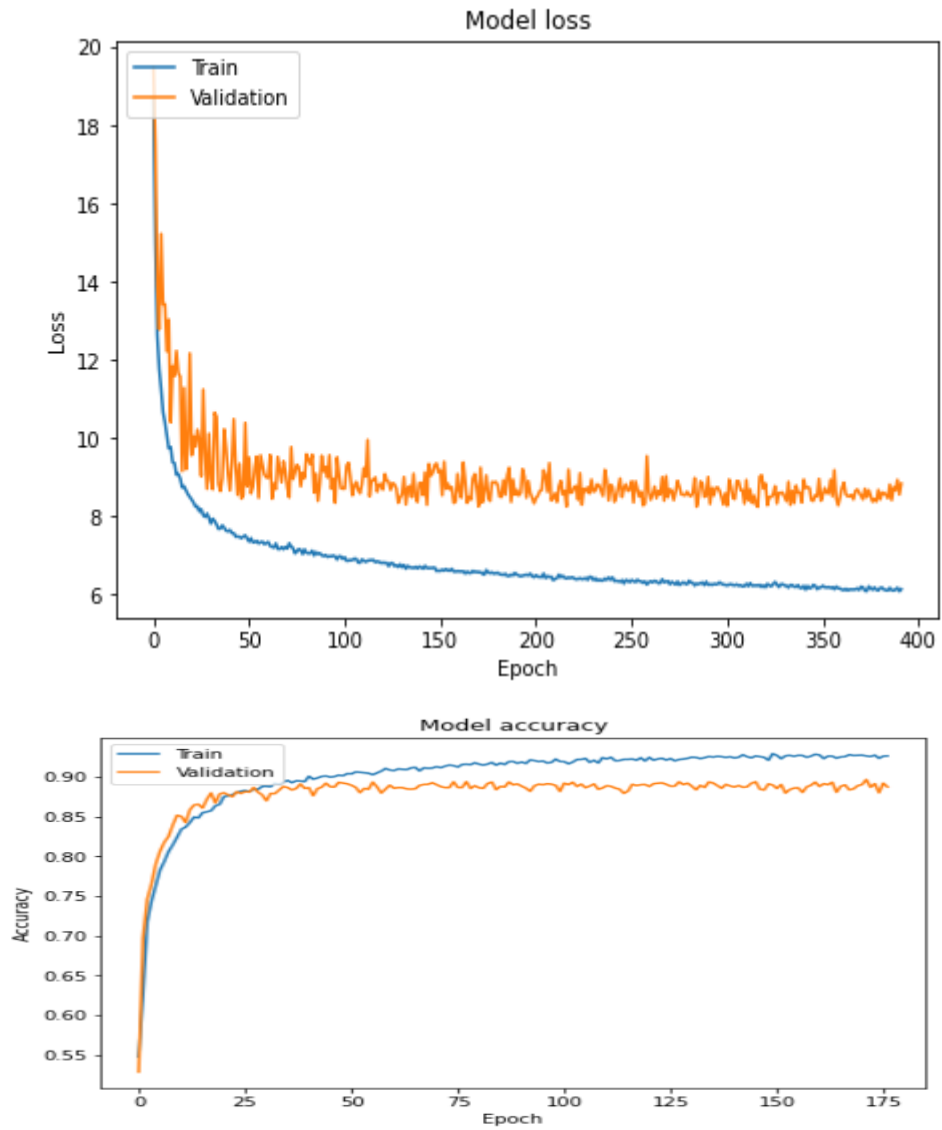


Fig.29 Model-1 Accuracy and Loss vs Epoch

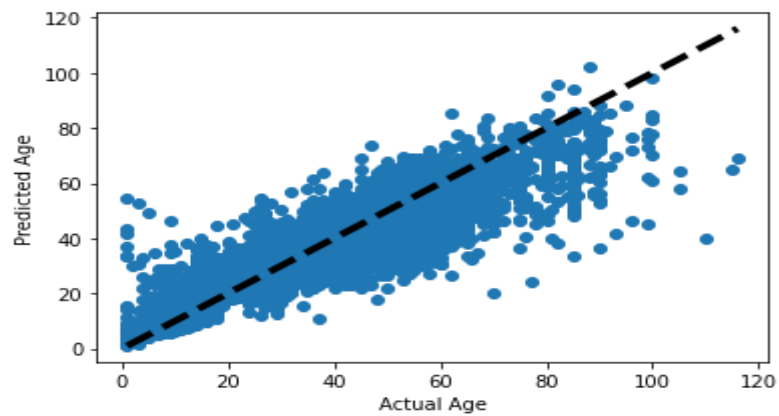


Fig.30 Model-1 Predicted vs Actual Age

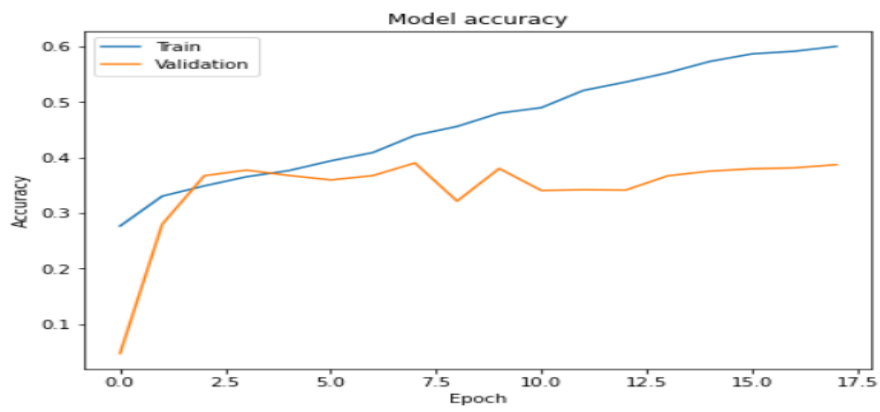
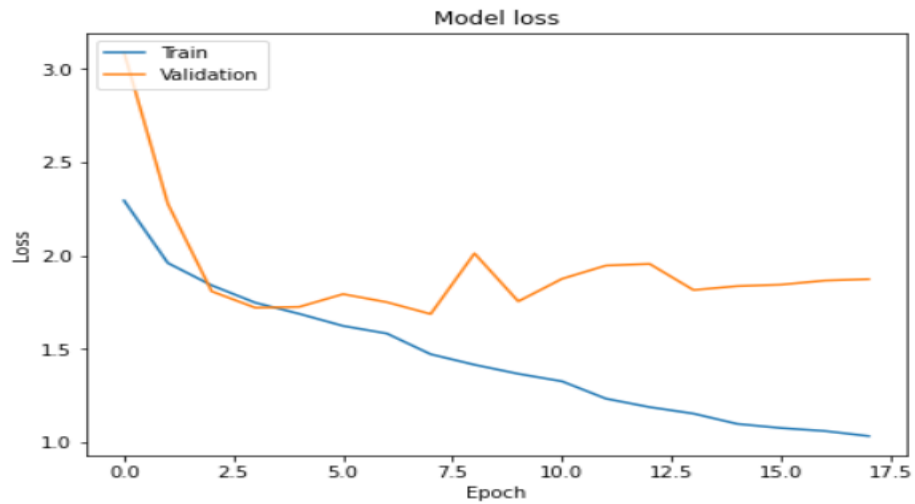
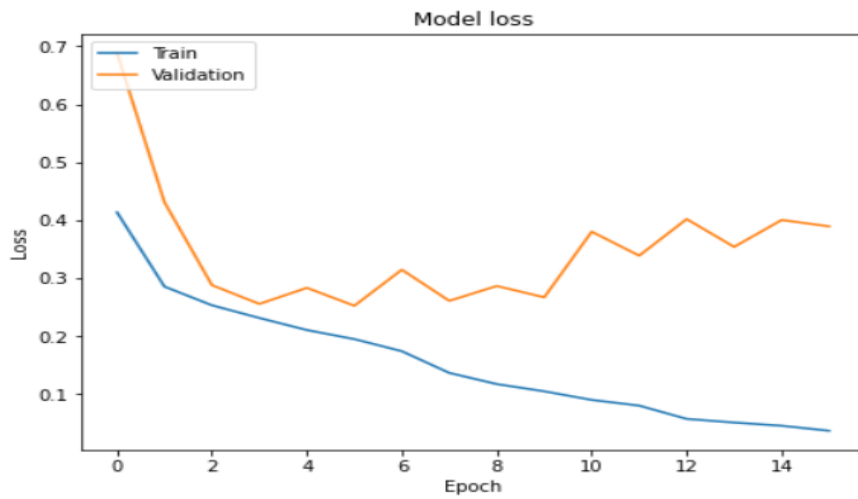


Fig.31 Model-2 Age Accuracy and Loss vs Epoch



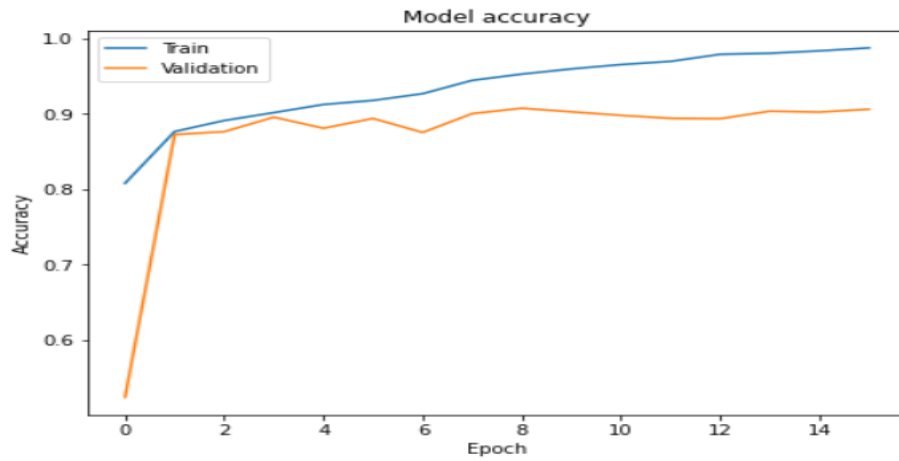


Fig.32 Model-2 Gender Accuracy and Loss vs Epoch

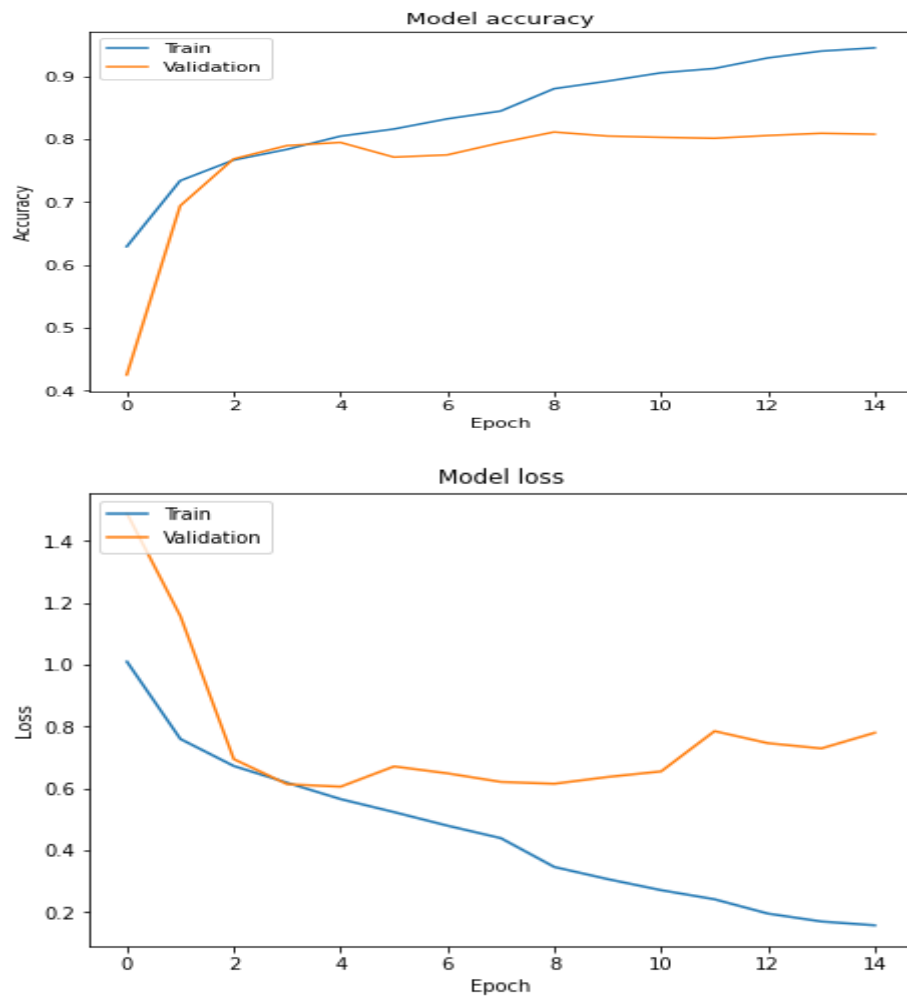


Fig.33 Model-2 Ethnicity Accuracy and Loss vs Epoch

6) Testing on some random inputs from dataset-

```

▶ test_image(57,images_i,images2,Model)
└─┬─┘
   │ Predicted Age: 71
   │ Predicted Sex: Male
   │ Predicted Ethnicity: White
   └─┬─┘
215] test_image(137,images_i,images2,Model)
     │ Predicted Age: 11
     │ Predicted Sex: Female
     │ Predicted Ethnicity: Asian
     └─┬─┘
216] test_image(502,images_i,images2,Model)
     │ Predicted Age: 72
     │ Predicted Sex: Female
     │ Predicted Ethnicity: White
     └─┬─┘
217] test_image(24,images_i,images2,Model)
     │ Predicted Age: 63
     │ Predicted Sex: Female
     │ Predicted Ethnicity: Black
     └─┬─┘

```

Fig.34 Model-1 Sample Prediction

```

[ ] test_image(57,images_i,images_i2,model_age,model_gender, model_ethnicity)
   │ Predicted Age: 50-55
   │ Predicted Gender: Male
   │ Predicted Ethnicity: White
   │ Actual Age: 80
   │ Actual Gender: Male
   │ Actual Ethnicity: White
   └─┬─┘
▶ test_image(137,images_i,images_i2,model_age,model_gender,model_ethnicity)
└─┬─┘
   │ Predicted Age: 5-10
   │ Predicted Gender: Female
   │ Predicted Ethnicity: Indian
   │ Actual Age: 8
   │ Actual Gender: Female
   │ Actual Ethnicity: Indian
   └─┬─┘
[ ] test_image(502,images_i,images_i2,model_age,model_gender,model_ethnicity)
   │ Predicted Age: 55-60
   │ Predicted Gender: Female
   │ Predicted Ethnicity: White
   │ Actual Age: 80
   │ Actual Gender: Female
   │ Actual Ethnicity: White
   └─┬─┘
▶ test_image(1150,images_i,images_i2,model_age,model_gender,model_ethnicity)
└─┬─┘
   │ Predicted Age: 50-55
   │ Predicted Gender: Male
   │ Predicted Ethnicity: Indian
   │ Actual Age: 70
   │ Actual Gender: Male
   │ Actual Ethnicity: Indian
   └─┬─┘

```

Fig.35 Model-2 Sample Prediction

7) Creating Website using Django and deploy using Heroku-

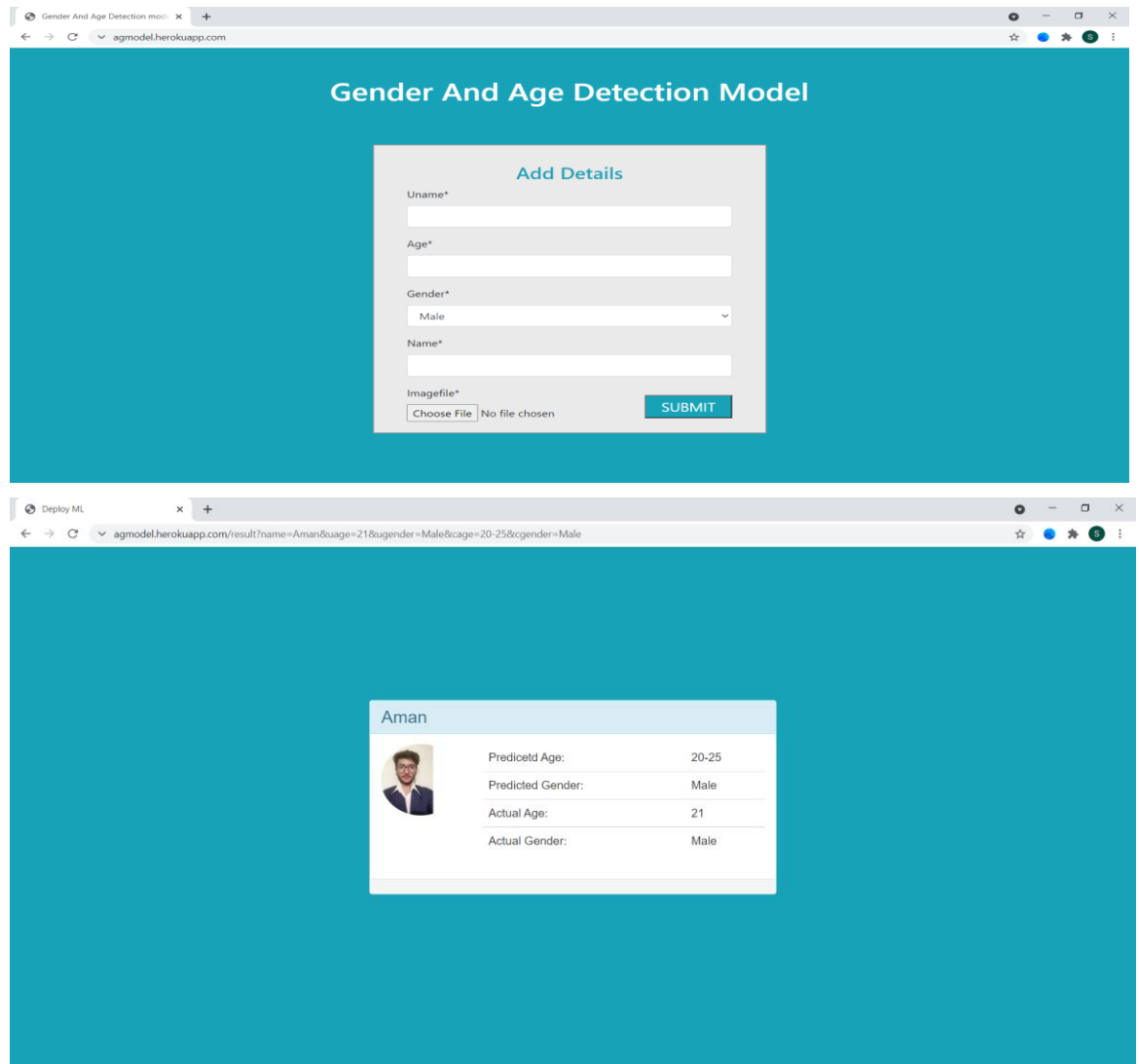


Fig.36 Heroku deployed website and results

4.3 Comparison of above results by at least two methods and justification for the differences or error in with theory or earlier published results

Our model gave a good result in terms of gender and ethnicity prediction but lacked accuracy in terms of age as compared to earlier models due to the fact that they used a relatively larger dataset and created bins with greater age gaps. In our model we used age gaps of 5 years while the model published earlier had age gaps of about 8 years for ages below 60 and ages above 60 were considered as one bin while in our model the age gap was continuous throughout. Moreover, the bins were discontinuous in their case and certain ages were left out at time of training thus increasing their age accuracy than ours.

CHAPTER-05: CONCLUSIONS

5.1 Conclusions

For a number of 'enterprises' and 'governments', "age, gender, ethnicity and the date of a person's most recent personal photograph" have lately become significant pieces of information that might serve as "commercial, identification, or safety objectives." To ensure that the correct data was entered into the enterprise system, forms validators were advised since this information was acquired from people. As part of this project, we want to present an alternative method of verifying the information provided by a single person picture, which will be compared to the gender and date of birth submitted by the person themselves. Then, after a thorough study, we discovered that it performs well in terms of gender, ethnicity prediction, but averagely in terms of age forecasting.

The first model has been trained on 500 epochs and gives a final accuracy of 88.09 percent for gender, 51.97 percent for ethnicity and accuracy of 4.59 percent for age with loss of 0.26 for gender, 0.63 for ethnicity and 6.47 for age. The next model which has been hosted on Heroku has been trained on 200 epochs and gives a final accuracy of 89.37 percent for gender, 79.50 for ethnicity and 40 percent accuracy for age with a loss of 0.25 for gender, 0.60 for ethnicity and 1.68 for age. The UTKface dataset is an average dataset hence makes the model more prone to error. It can be concluded that with increase in number of photos in dataset the accuracy can be improved. Decreasing the number of bin classes for age though increases the accuracy but in real time we want an integer number for age or a class with less gap, example: one can easily predict the age in bins of size 20 but same would not be accurate in real world and so to increase accuracy with number of bins work is being carried out to detect ages.

5.2 Future Scope

There are several ways in which this technique might be enhanced in order to mitigate the weaknesses in 'age, gender and ethnicity' prediction. First, by using a huge and complicated model that was trained on more people photographs, then analyzing the

estimate accuracy & performance speed while not consuming a significant number of resources, and the optimizing it to be useful in application forms. Due to the lack of accuracy in 'age, gender and ethnicity' prediction' from women images, it is also possible to train and test a CNN model using a 'big Face dataset' that comprises portraits and a fair distribution of 'males and females' numbers, and then assess it using the testing dataset. In addition, we want to build a 'Restful API' for the same, which will allow it to be utilized by anybody without difficulty.

5.3 Applications

A number of applications of 'age, gender and ethnicity' classification are:

- I. Visual Surveillance
- II. Content based searching
- III. Targeted Advertisement
- IV. Forensic Science
- V. Human Computer Interaction System

REFERENCES

- 1) Sooyeon Lim, “Estimation of ‘age, gender and ethnicity’ using CNN-based face recognition algorithm”, International Journal of Advanced Smart Convergence Vol.9 No.2 203-211 (2020)
- 2) R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, “Face detection in color images,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 5, pp. 696–706, May 2002, doi: 10.1109/34.1000242.
- 3) R. Rothe, R. Timofte and L. V. Gool, “DEX: Deep EXpectation of apparent age from a single image,” in Proc. IEEE International Conference on Computer Vision Workshops, Dec, 2015.
- 4) Y. H. Kwon and D. Vitoria Lobo, “Age classification from facial images,” Computer vision and image understanding, Vol. 74, pp. 1-21, 1999.
- 5) T. R. Kalansuriya, and A. T. Dharmaratne, “Neural Network based ‘age, gender and ethnicity’ Classification for Facial Images,” International Journal on Advances in ICT for Emerging Regions, vol. 7, no. 2, 2014.
- 6) Abdullah M. Abu Nada1, “‘age, gender and ethnicity’ Prediction and Validation Through Single User Images Using CNN”, International Journal of Academic Engineering Research (IJAER) ISSN: 2643-9085 Vol. 4, Issue 8, August – 2020, Pages: 21-24
- 7) Müge Çarıkçı, “A Face Recognition System Based on Eigenfaces Method”, 2212-0173 © 2012 Published by Elsevier Ltd.
- 8) Mustamin Anggo and La Arapu, “Face Recognition Using Fisherface Method”, IOP Conf. Series: Journal of Physics: Conf. Series 1028 (2018) 012119
- 9) Gonzalo Benito, “Comparing classical and deep approaches for face recognition in a smartgym application”, Sergio Escalera, Mathematics and Informatics Dept., Universitat de Barcelona and Computer Vision Center, September 2017
- 10) N. Chauhan, "Predict ‘age, gender and ethnicity’ using Convolutional Neural Network and openCV", Medium, 2020.