A picture containing text, metalware, gear

Description automatically generatedLogo

Description automatically generated with low confidence

**RECOMMENDATION SYSTEM BASED ON AGE AND GENDER**

**A PROJECT REPORT**

**Submitted by**

**MIRUTHUVIKASINI.S (18BCS021)**

**PRAGATHI.R (18BCS038)**

**RAGHAVI.S(18BCS057)**

***In partial fulfilment for the award of the degree***

*of*

**BACHELOR OF ENGINEERING**

in

**COMPUTER SCIENCE AND ENGINEERING**

**KUMARAGURU COLLEGE OF TECHNOLOGY**

**COIMBATORE-641 049**

(An Autonomous Institution Affiliated to Anna University, Chennai)

**January 2022**

Logo

Description automatically generated with low confidenceA picture containing logo

Description automatically generated**KUMARAGURU COLLEGE OF TECHNOLOGY**

**COIMBATORE 641 049**

(An Autonomous Institution Affiliated to Anna University, Chennai)

**BONAFIDE CERTIFICATE**

Certified that this project report **“RECOMMENDATION SYSTEM BASED ON AGE AND GENDER”** is the bonafide work of **S.MIRUTHUVIKASINI (18BCS021) ,R.PRAGATHI (18BCS038) AND S.RAGHAVI (18BCS057)”** who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **Dr. Devaki. P, Ph.D.,** | **-----------------** |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| Department of Computer Science and  Engineering, | Department of Computer Science and Engineering, |
| Kumaraguru College of Technology | Kumaraguru College of Technology |
| Coimbatore – 641 049. | Coimbatore – 641 049. |

The candidates with University register number **18BCS021, 18BCS038, 18BCS057** were examined in the Project Viva-Voce examination held on …………………

Internal Examiner External Examiner

**DECLARATION**

We affirm that the project work titled **“RECOMMENDATION SYSTEM BASED ON AGE AND GENDER”** being submitted in partial fulfillment for the award of B.E Computer Science and Engineering is the original work carried out by us. It has not formed the part of any other project work submitted for the award of any degree or diploma, either in this or any other University.

**S. MIRUTHUVIKASINI (18BCS021)**

**R. PRAGATHI (18BCS038)**

**S. RAGHAVI (18BCS057)**

I certify that the declaration made above by the candidates is true.

----------------------

Assistant Professor,

Department of Computer Science and Engineering,

Kumaraguru College of Technology,

Coimbatore – 641 049.

**ACKNOWLEDGEMENT**

We express our profound gratitude to the management of Kumaraguru College of Technology for providing as with the required infrastructure that enabled us to successfully complete the project.

We extend our gratefulness to our Principal, **Dr. D. Saravanan,** for providing us the necessary facilities to pursue the project.

We would like to acknowledge **Dr. P. Devaki,** Professor and Head, Department of Computer Science and Engineering, for her support and encouragement throughout this project.

We thank our project coordinator **Dr. L. Latha,** Professor, Department of Computer Science and Engineering and guide **--------------,** AssistantProfessor, Department of Computer Science and Engineering, for their constant and continuous effort, guidance and valuable time.

Our sincere and hearty thanks to staff members of Department of Computer Science and Engineering of Kumaraguru College of Technology for their well wishes, timely help and support rendered to us during our project. We are greatly indebted to our family, relatives and friends, without whom life would have not been shaped to this level.

**S.MIRUTHUVIKASINI**

**R.PRAGATHI**

**S.RAGHAVI**

### **TABLE OF CONTENTS**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | | | **PAGE NO.** | | | |
|  | **ABSTRACT** | | | 6 | | | |
| 1 | **INTRODUCTION** | | | 7 | | | |
|  | 1.1 | CONCEPTUAL STUDY OF THE PROJECT | | 7 | | | |
|  | 1.2 | OBJECTIVES OF THE PROJECT | | 7 | | | |
|  | 1.3 | SCOPE OF THE PROJECT | | 8 | | | |
| 2 | **LITERATURE REVIEW** | | | 8 | | | |
|  | 2.1 | LITERATURE REVIEW OF JOURNALS | | 8 | | | |
| 3 | **PROBLEM DEFINITION** | | | 11 | | | |
| 4 | **PROPOSED SYSTEM** | | | 11 | | | |
|  | 4.1 | METHODOLOGY | | 11 | | | |
|  | 4.2 | FLOW DIAGRAM | 12 | | |
|  | 4.3 | IMPLEMENTATION | | | 15 | |
| 5 | **SYSTEM REQUIREMENTS** | | | 31 | | | |
|  | 5.1 | REQUIREMENTS | | 31 | | | |
| 6 | **CONCLUSION** | | | 31 | | | |
| 7 | **APPENDIX** | | | 31 | | | |

### **ABSTRACT**

This paper proposes an age and gender prediction method from face images using convolutional neural network (CNN). The CNN architecture involves two levels which are feature extraction and classification. Our CNN is pre-processed and trained on an Adience dataset. Out of the many applications of age and gender prediction, we apply it to improve marketing strategy by recommending products in any online platform based on the target audience.

**1.INTRODUCTION**

**1.1 CONCEPTUAL STUDY OF THE PROJECT:**

Facial analysis has gained much recognition in recent times. Age and gender predictions of unfiltered faces classify the real-world facial images into predefined age and gender. Using deep-CNN method provides an efficient performance compared to the traditional conventional methods[4]. We can use Haar cascade[8] approach which is an object detection algorithm used to identify facial images from real time videos also. The predicted age and gender is used in many applications such as access control, law, human-computer interaction, marketing, visual surveillance and so on.

The face of the customer is detected when they open the camera in online platform and analysed for age and gender prediction based on which the further screens or devices could recommend products which will be useful for them. All the Data necessary for the recommendation system should be previously fed by us. This system of approach will improve the marketing standards and can help both the customers to fulfill their needs and also the seller for greatly improving their sales percentage.

* 1. **OBJECTIVE OF THE PROJECT**

We propose an age and gender prediction method from face images using convolutional neural network (CNN). Our CNN is pre-processed and trained on an Adience dataset. Out of the many applications of age and gender prediction, we apply it to improve marketing strategy by recommending products in any online platform based on the target audience.

**1.3 SCOPE OF THE PROJECT**

Marketing strategy is greatly influenced by recommendation system. We are entering a data collection era, everything from our preferences to our search histories are stored and used to recommend us related items.

Example: YouTube recommends us what videos to watch next.

Amazon recommends related items to add to our shopping cart.

Our Work creates a demographic way of product recommendation in online shopping platforms. The face of the customer is detected using webcam or by uploading the image. By detecting age and gender using our model we will suggest the products .

**2. LITERATURE REVIEW:**

**Strategies used:**

The Physiological parameters such as fingerprint, face, palm etc are the replica of DNA. So, while segmenting the facial feature there are lots of factors to be considered which includes pose, noise, texture, lighting conditions and distance between the object and the camera, Occlusion, Illumination etc. By considering all these features and building a responsive face recognition model has its own great impact on the real-world automation.

**Age Classification:**

After image recognition there are early classification of age was done by calculation ratio between different features of face like nose, eyes, mouth, chin etc. After localizing calculating their sizes and distances, ratio between them is calculated in order to predict age by using conventional methods. Gaussian Mixture Model (GMM) [5] works on facial patches distribution. Gabor image descriptor method is used in used along with Fuzzy-LDA classifier to detect image of face belonging to more than one age group. As a whole Biological Inspired features (BIF) and other manifold-learning features are used for age prediction[7]. Gabor image descriptor and local binary pattern (LBP) [3] were used with hierarchical age classifier method consists of Support Vector Machines (SVM) [5] are used to classify input image to a specific age class by using support vector regression so that the result obtained would be precise.

There are various methods proposed on image recognition all those were acceptable for well aligned and front facing images exactly, thus that proposition also don’t meet the needs. So, these methods give only experimental result on limited data sets. Thus, such methods are inappropriate to use in real time streaming images. Most of the methods described above uses FERET benchmark to develop a most precise system giving accurate results[1]. FERET images were taken to extremely measured complaint and the result obtained from them are highly saturated. It is actually difficult to find out actual advantages of these techniques as FERET benchmark contains filtered images which makes the result more precise[3].

**Gender Classification**

Same as Age classification, Gender classification also has its own significant implementation in image processing world. It may seem easy when we try to classify gender on common but it is not actually true. When it comes to Gender classification, we group into two classes (Male, Female) in general. Previous works were on the neural network trained on a small set of near-frontal face images such as UIUC-IFP-Y, FG-NET and MORPH. Some works used SVM Classifier directly on image intensities[8]. But the model which uses Webers Local texture Descriptor for gender recognition, demonstrated its perfectness comparing with the above models.EEG-Based Age and Gender Prediction Using Deep BLSTM-LSTM Network Model were also made by a couple of researchers [6][9].

**CNN on prediction:**

The first application of Convolutional Neural Network (CNN) is LeNet-5 network by using optical character recognition. In the previous work 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.

[1]Finally, a Deep convolution method was introduced by Levi and Hassner model where the model is trained with more of unclear and unfiltered images which gave a better result for uneven images. The model is so simple and solves the problem of overfitting and works fine for even unconstrained images. When we talk about the major contribution of Deep CNN it was working successfully even on the challenging ImageNet benchmark. Additionally, it was also used in pose estimation, face parsing, facial keypoint detection, speech recognition. In general Overfitting (Low bias and high variance) is the common problem across most of the machine learning model it is mainly due to minimal size of the data as it learns detailed noise in the training data.

Levi and Hassner Deep CNN was fed with rescaled images of 256 × 256 and a crop of 227 × 227. The network comprises of 3 Convolution Layers with 2 Fully connected layer.

Experimental results proved the small misalignments has created a drastic change over the result. So, to overcome the mentioned variation as well as the problem of overfitting the network training was given in 2 methods one is Centre Crop (Cropped to 227 × 227 around the face centre) and another is Over-sampling (Five 227 × 227 pixel crop regions, Four from the corners of the 256 × 256 face image, and an additional crop region from the centre of the face, along with horizontal reflections). Training the network by translating alignments of the same face made this model to stand out of all the existing systems even with simpler network architecture.

**Achievements of existing system:**

A deep convolutional method which was proposed by Levi and Hassner model was trained with more of unclear and unfiltered images. They constitute this model as a multi class classification problem. The existing model was originally pre-trained on age and gender called IMDB-WIKI dataset[2][8]. The images from the mentioned dataset were obtained directly from the website with some degrees of variability and the images were fine-tuned on MORPH II which was the other largest facial aging dataset which include age and gender annotations too[2]. And they also used OIU-Adience (original dataset) which consists of unfiltered faces of age and gender classification to finely tune the Levi and Hassner model. They used robust image pre-processing algorithm, which handles the maximum variability observed in the unfiltered real-world faces (facial images). While investigating the accuracy of classification on OIU-dataset both the age and gender classification accuracy achieved the state-of-the-art performance, which outperformed the existing models.

**Disadvantages of existing system:**

The biggest issue with the age prediction model which was trained by Levi and Hassner is it’s heavily biased toward the age group 25-32. This means that Levi ad Hassner age prediction model may predict the 25-32 age group when the actual age belongs to a different age group[1].

**3.PROBLEM DEFINITION:**

To build a gender and age detector that can approximately guess the gender and age of the person (face) in a picture or through webcam, and recommend the products based on demographic parameters.

**4. PROPOSED SYSTEM:**

The main motto of our system is to recommend products based on demographic parameters such as Age and Gender. This system plays a crucial role in the marketing strategies due to its unique recommendation system. Our application will detect age and gender of users who wants to shop in the online platform, based on online facial analyses and by using recommendation algorithms it will suggest the products for the targeted audience. We will be training a convolutional neural network that will predict the age group and gender from an image containing the face of a person. Output of the above would-be age and gender. By feeding the above as input to the recommendation system we reach our goal finally by getting our recommended product.

**MODULES OF OUR PROPOSED SYSTEM:**

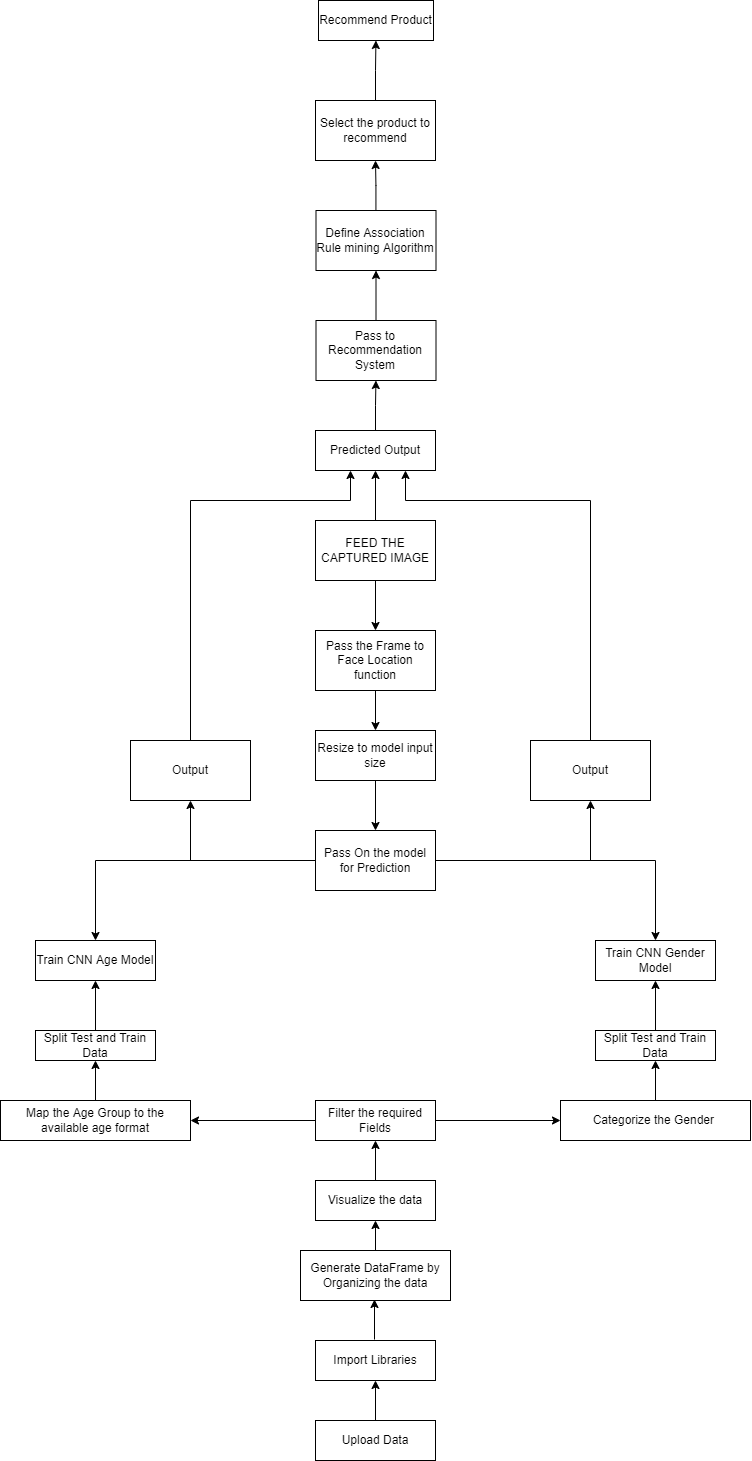
**4.1 METHODOLOGY:**

We have lots of Algorithms trained especially for facial recognition. But CNN has its own way of dealing with image processing. A convolution is essentially sliding a filter over the input. Each convolutional layer contains a series of filters known as convolutional kernels.

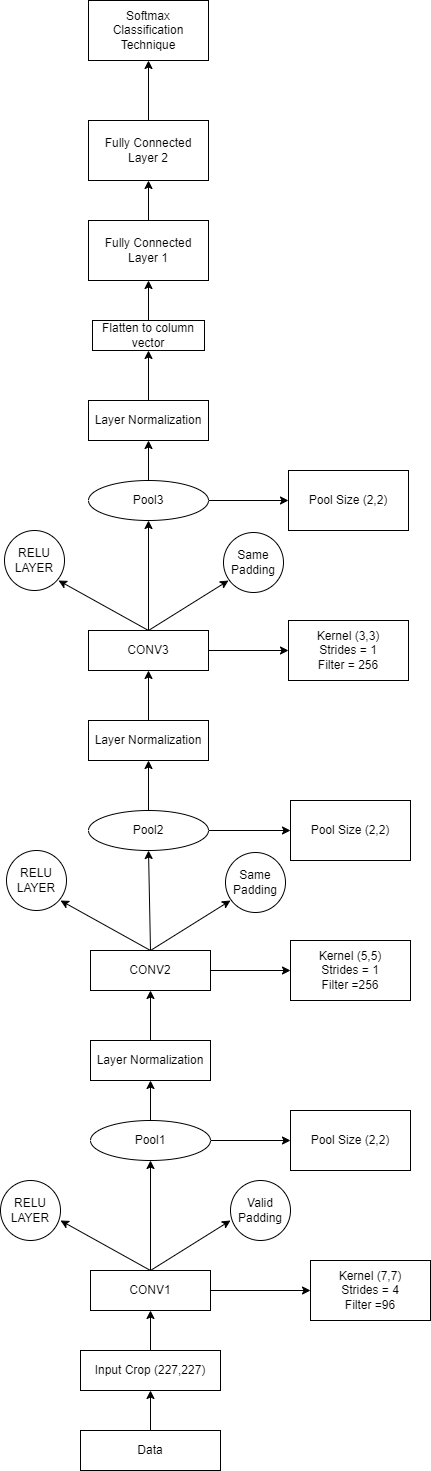
The filter is a matrix of integers that are used on a subset of the input pixel values, the same size as the kernel. Each pixel is multiplied by the corresponding value in the kernel, then the result is summed up for a single value for simplicity representing a grid cell, like a pixel, in the output channel/feature map. Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Moving on, we are going to into fully connected layer which we flatten the final output and feed it to a regular Neural Network for classification purposes.

Flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training.The model is able to distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique. Finally Output is classified into different classes.

**4.2 FLOW DIAGRAM:**

****

**Network Architecture**

****

**4.3. IMPLEMENTATION:**

**DATA COLLECTION:**

For this python project, we had used the Adience dataset; the dataset is available in the public domain. This dataset serves as a benchmark for face photos and is inclusive of various real-world imaging conditions like noise, lighting, pose, and appearance. The images have been collected from Flickr albums and distributed under the Creative Commons (CC) license. It has a total of 26,580 photos of 2,284 subjects in eight age ranges (as mentioned above) and is about 1GB in size. The models we used had been trained on this dataset.

The directory contains the following files:

* faces.tar.gz (936M) – Face images, cropped
* aligned.tar.gz (1.9G) – Face images, cropped and aligned using our 2D, in plain alignment tool
* fold\_0\_data.txt – fold\_4\_data.txt – text files with indices to the five-fold cross validation tests using all faces
* fold\_frontal\_0\_data.txt – fold\_frontal\_4\_data.txt – same as above, but using only faces in approximately frontal pose.

#### **Uploading the data**

Mount the drive and navigate to the folder that has the dataset in it.

#### **Import the necessary libraries for loading and viewing data:**

**Numpy:** for working with the data, cleaning it, formatting it in the required way and deleting irrelevant data.

**Pandas:** for reading the dataset csv files.

**Matplotlib:** for plotting the graphs and showing images inside of the colab console along with seaborn.

**Opencv and PIL:** for working with images resizing it, formating it for the model and stuff.

**Seaborn** is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Graphical user interface, text, application, email

Description automatically generated

#### **Read the Data**

There are 5 files with names

fold\_0\_data.txt,

fold\_1\_data.txt,

fold\_2\_data.txt,

fold\_3\_data.txt,

fold\_4\_data.txt

that contain the data. We will read it using pandas.read csv function.

A picture containing text

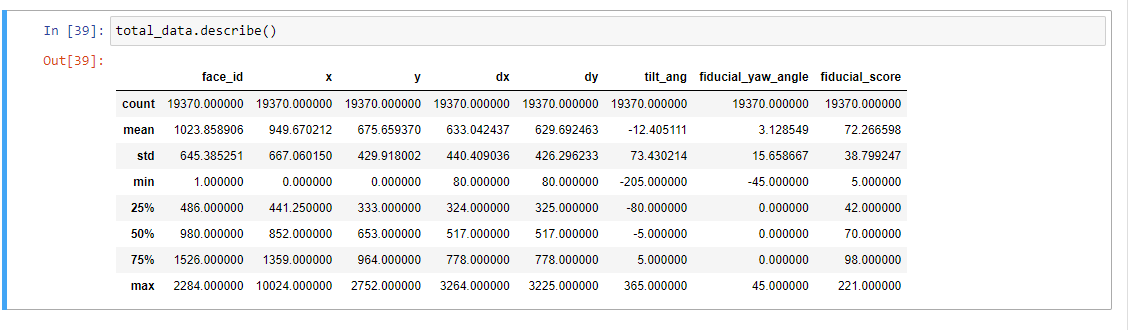
Description automatically generated

**Data Description:**

We have data from 5 different files, we need to add the data from all those files into single pandas dataframe and print the shape and info about it.

**Graphical user interface, text, application, email

Description automatically generated**

****

**Top 10 Records**

Print the top 10 records from the dataframe using.head() function, verifying the data structure.

Table

Description automatically generated

#### **Import the necessary modules for model building**

We will be using Keras with TensorFlow for creating this model.

The model will be a sequential model ie: data will pass sequentially from one layer to another, there will be no jump nodes or breakpoints in the flow of data.

All the required layers are imported from Keras and the function for loading them.

**Sample Check:**

Load the first sample data and check it. This will verify that the data structure and our project structure are correct.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

#### **Use the relevant data and map them**

We need age, gender, position, and bounding box for all images, we copy that in another df. Also from the data, create a relative link to images stored in the drive and add that link as well to the df.

**Graphical user interface, text, application, email

Description automatically generated**

A picture containing graphical user interface

Description automatically generated

**Mapping Age data to a Age Group**

Create an age mapping dictionary and map each age data from dataset to an age group.

Graphical user interface, text, application

Description automatically generated

**Visualization**

Ploting a bar graph for gender values. This will visualize the variance in data as well as overview of what the gender data holds.

Graphical user interface, text, application

Description automatically generated

Chart, pie chart

Description automatically generated

Chart

Description automatically generated

**Removing the irrelevant data:**

Drop the records that do not have a correct age value

Drop the records that do not have a known gender associated with them and print the stats for the remaining data.

The records that have unknown gender will be dropped because we cannot use them to train our model.

Graphical user interface, text, application

Description automatically generated

**Mapping Gender as 0 and 1:**

Map the gender to class labels 0 and 1 and print the first five records to check the integrity of the data

Text

Description automatically generated with medium confidence

**Describing class labels:**

Map the ages to class labels 0 to 7 for each age group and print the first 5 records to check

Text

Description automatically generated

**Create training and test datasets for Gender:**

With sklearn train test split method

For gender model resize the images to 227,227 convert them to numpy array and add them to the test and train dataset. This process may take up a little time because the dataset is not very small.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

#### **Gender Model**

We define the CNN model here, we are using a sequential model i.e. data will travel sequentially from one layer to another without any jumps and splitting.

Graphical user interface, text, application

Description automatically generated

A picture containing text

Description automatically generated

Table

Description automatically generated with medium confidence

#### **Train the model**

A picture containing graphical user interface

Description automatically generated

Table

Description automatically generated with medium confidence

A picture containing table

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

#### **Create training and testing split for age data**

Text

Description automatically generated

#### **Age Model**

**Text, application

Description automatically generated**

**Table

Description automatically generated with medium confidence**

**Graphical user interface

Description automatically generated with low confidence**

**Graphical user interface, text, application

Description automatically generated**

**Table

Description automatically generated with medium confidence**

**5. SYSTEM REQUIREMENTS:**

Operating system : Windows 10

Framework : Tensorflow,Keras,NumPy,Pandas, Matplotlib,seaborn,

sklearn, opencv and pillow v8

Language : Python

Source : OIU-Adience Dataset

CPU : intel i5

RAM : 8 GB

**6.CONCLUSION**

A Convolutional Neural Network (ConvNet/CNN) is a **Deep Learning algorithm** which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. Hence our project is designed to detect the demographic details such as age and gender and recommend the products to user based on result with help of deep learning techniques. Now the dataset is preprocessed and trained .

**7.APPENDIX**

import os

from google.colab import drive

drive.mount('/content/drive')

os.chdir('/content/drive/My Drive/DataFlair')

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import cv2

from PIL import Image

#load data

fold0 = pd.read\_csv("AdienceBenchmarkGenderAndAgeClassification/fold\_0\_data.txt",sep = "\t" )

fold1 = pd.read\_csv("AdienceBenchmarkGenderAndAgeClassification/fold\_1\_data.txt",sep = "\t")

fold2 = pd.read\_csv("AdienceBenchmarkGenderAndAgeClassification/fold\_2\_data.txt",sep = "\t")

fold3 = pd.read\_csv("AdienceBenchmarkGenderAndAgeClassification/fold\_3\_data.txt",sep = "\t")

fold4 = pd.read\_csv("AdienceBenchmarkGenderAndAgeClassification/fold\_4\_data.txt",sep = "\t")

total\_data = pd.concat([fold0, fold1, fold2, fold3, fold4], ignore\_index=True)

print(total\_data.shape)

total\_data.info()

total\_data.head()

#bar chart

gender = ['f','m','u']

plt.bar(gender, total\_data.gender.value\_counts(), align='center', alpha=0.5)

plt.show()

import tensorflow as tf

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D, Activation, Dropout, Flatten, Dense, Dropout, LayerNormalization

from keras.preprocessing.image import ImageDataGenerator, img\_to\_array, load\_img

path = "AdienceBenchmarkGenderAndAgeClassification/faces/"+total\_data.user\_id.loc[0]+"/coarse\_tilt\_aligned\_face."+str(total\_data.face\_id.loc[0])+"."+total\_data.original\_image.loc[0]

img = load\_img(path)

plt.imshow(img)

plt.show()

imp\_data = total\_data[['age', 'gender', 'x', 'y', 'dx', 'dy']].copy()

imp\_data.info()

img\_path = []

for row in total\_data.iterrows():

path = "AdienceBenchmarkGenderAndAgeClassification/faces/"+row[1].user\_id+"/coarse\_tilt\_aligned\_face."+str(row[1].face\_id)+"."+row[1].original\_image

img\_path.append(path)

imp\_data['img\_path'] = img\_path

imp\_data.head()

age\_mapping = [('(0, 2)', '0-2'), ('2', '0-2'), ('3', '0-2'), ('(4, 6)', '4-6'), ('(8, 12)', '8-13'), ('13', '8-13'), ('22', '15-20'), ('(8, 23)','15-20'), ('23', '25-32'), ('(15, 20)', '15-20'), ('(25, 32)', '25-32'), ('(27, 32)', '25-32'), ('32', '25-32'), ('34', '25-32'), ('29', '25-32'), ('(38, 42)', '38-43'), ('35', '38-43'), ('36', '38-43'), ('42', '48-53'), ('45', '38-43'), ('(38, 43)', '38-43'), ('(38, 42)', '38-43'), ('(38, 48)', '48-53'), ('46', '48-53'), ('(48, 53)', '48-53'), ('55', '48-53'), ('56', '48-53'), ('(60, 100)', '60+'), ('57', '60+'), ('58', '60+')]

age\_mapping\_dict = {each[0]: each[1] for each in age\_mapping}

drop\_labels = []

for idx, each in enumerate(imp\_data.age):

if each == 'None':

drop\_labels.append(idx)

else:

imp\_data.age.loc[idx] = age\_mapping\_dict[each]

imp\_data = imp\_data.drop(labels=drop\_labels, axis=0) #droped None values

imp\_data.age.value\_counts(dropna=False)

imp\_data = imp\_data.dropna()

clean\_data = imp\_data[imp\_data.gender != 'u'].copy()

clean\_data.info()

gender\_to\_label\_map = {

'f' : 0,

'm' : 1

}

clean\_data['gender'] = clean\_data['gender'].apply(lambda g: gender\_to\_label\_map[g])

clean\_data.head()

age\_to\_label\_map = {

'0-2' :0,

'4-6' :1,

'8-13' :2,

'15-20':3,

'25-32':4,

'38-43':5,

'48-53':6,

'60+' :7

}

clean\_data['age'] = clean\_data['age'].apply(lambda age: age\_to\_label\_map[age])

clean\_data.head()

X = clean\_data[['img\_path']]

y = clean\_data[['gender']]

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

print('Train data shape {}'.format(X\_train.shape))

print('Test data shape {}'.format(X\_test.shape))

train\_images = []

test\_images = []

for row in X\_train.iterrows():

image = Image.open(row[1].img\_path)

image = image.resize((227, 227)) # Resize the image

data = np.asarray(image)

train\_images.append(data)

for row in X\_test.iterrows():

image = Image.open(row[1].img\_path)

image = image.resize((227, 227)) # Resize the image

data = np.asarray(image)

test\_images.append(data)

train\_images = np.asarray(train\_images)

test\_images = np.asarray(test\_images)

print('Train images shape {}'.format(train\_images.shape))

print('Test images shape {}'.format(test\_images.shape))

model = Sequential()

model.add(Conv2D(input\_shape=(227, 227, 3), filters=96, kernel\_size=(7, 7), strides=4, padding='valid', activation='relu'))

model.add(MaxPooling2D(pool\_size=(2,2),strides=(2,2)))

model.add(LayerNormalization())

model.add(Conv2D(filters=256, kernel\_size=(5, 5), strides=1, padding='same', activation='relu'))

model.add(MaxPooling2D(pool\_size=(2,2),strides=(2,2)))

model.add(LayerNormalization())

model.add(Conv2D(filters=256, kernel\_size=(3, 3), strides=1, padding='same', activation='relu'))

model.add(MaxPooling2D(pool\_size=(2,2),strides=(2,2)))

model.add(LayerNormalization())

model.add(Flatten())

model.add(Dense(units=512, activation='relu'))

model.add(Dropout(rate=0.25))

model.add(Dense(units=512, activation='relu'))

model.add(Dropout(rate=0.25))

model.add(Dense(units=2, activation='softmax'))

model.summary()

“””TRAIN”””

callback = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=3) # Callback for earlystopping

model.compile(optimizer='adam', loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True), metrics=['accuracy'])

history = model.fit(train\_images, y\_train, batch\_size=32, epochs=25, validation\_data=(test\_images, y\_test), callbacks=[callback])

print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")

model.save('gender\_model25.h5')

test\_loss, test\_acc = model.evaluate(test\_images, y\_test, verbose=2)

print(test\_acc)

X = clean\_data[['img\_path']]

y = clean\_data[['age']]

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

print('Train data shape {}'.format(X\_train.shape))

print('Test data shape {}'.format(X\_test.shape))

train\_images = []

test\_images = []

for row in X\_train.iterrows():

image = Image.open(row[1].img\_path)

image = image.resize((227, 227)) # Resize the image

data = np.asarray(image)

train\_images.append(data)

for row in X\_test.iterrows():

image = Image.open(row[1].img\_path)

image = image.resize((227, 227)) # Resize the image

data = np.asarray(image)

test\_images.append(data)

train\_images = np.asarray(train\_images)

test\_images = np.asarray(test\_images)

print('Train images shape {}'.format(train\_images.shape))

print('Test images shape {}'.format(test\_images.shape))

"""Age model"""

model = Sequential()

model.add(Conv2D(input\_shape=(227, 227, 3), filters=96, kernel\_size=(7, 7), strides=4, padding='valid', activation='relu'))

model.add(MaxPooling2D(pool\_size=(2,2),strides=(2,2)))

model.add(LayerNormalization())

model.add(Conv2D(filters=256, kernel\_size=(5, 5), strides=1, padding='same', activation='relu'))

model.add(MaxPooling2D(pool\_size=(2,2),strides=(2,2)))

model.add(LayerNormalization())

model.add(Conv2D(filters=256, kernel\_size=(3, 3), strides=1, padding='same', activation='relu'))

model.add(MaxPooling2D(pool\_size=(2,2),strides=(2,2)))

model.add(LayerNormalization())

model.add(Flatten())

model.add(Dense(units=512, activation='relu'))

model.add(Dropout(rate=0.25))

model.add(Dense(units=512, activation='relu'))

model.add(Dropout(rate=0.25))

model.add(Dense(units=8, activation='softmax'))

model.summary()

callback = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=3) # Callback for earlystopping

model.compile(optimizer='adam', loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True), metrics=['accuracy'])

history = model.fit(train\_images, y\_train, batch\_size=32, epochs=50, validation\_data=(test\_images, y\_test), callbacks=[callback])

model.save('age\_model50.h5')

test\_loss, test\_acc = model.evaluate(test\_images, y\_test, verbose=2)

print(test\_acc)

**GITHUB Link:**

https://github.com/PRAGATHIRAM/Recommendation-System-Based-on-Age-and-Gender

**REFERENCES:**

1. G. Levi and T. Hassner, “Age and gender classification using convolutional neural networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015, pp. 34–42.
2. K. Ito, H. Kawai, T. Okano and T. Aoki, "Age and Gender Prediction from Face Images Using Convolutional Neural Network," 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2018, pp. 7-11, doi:
   1. 10.23919/APSIPA.2018.8659655.
3. T. Ahonen, A. Hadid, and M. Pietikainen, “Face description with local binary patterns: Application to face recognition,” IEEE Trans. Pattern Anal. Mach. Intell., no. 12, pp. 2037– 2041, 2006.
4. Rafique, A. Hamid, S. Naseer, M. Asad, M. Awais and T. Yasir, "Age and Gender Prediction using Deep Convolutional Neural Networks," 2019 International Conference on Innovative Computing (ICIC), 2019, pp. 1-6, doi: 10.1109/ICIC48496.2019.8966704.
5. H. A. Sánchez-Hevia, R. Gil-Pita, M. Utrilla-Manso and M. Rosa-Zurera, "Convolutionalrecurrent Neural Network for Age and Gender Prediction from Speech," 2019 Signal Processing Symposium (SPSympo), 2019, pp. 242-245, doi: 10.1109/SPS.2019.8881961.
6. P. Kaushik, A. Gupta, P. P. Roy and D. P. Dogra, "EEG-Based Age and Gender Prediction Using Deep BLSTM-LSTM Network Model," in IEEE Sensors Journal, vol. 19, no. 7, pp. 2634-2641, 1 April1, 2019, doi: 10.1109/JSEN.2018.2885582.
7. Khaung Tin, Dr.Hlaing Htake. (2011). Gender and Age Estimation Based on Facial Images. International Journal : ACTA TECJNICA NAPOCENSIS Electronics and Telecommunications.
8. *Michal Uricar, Radu Timofte, Rasmus Rothe, Jiri Matas, Luc Van Gool*; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2016, pp. 25-33
9. P. Nguyen, D. Tran, X. Huang and W. Ma, "Age and gender classification using EEG paralinguistic features," *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)*, 2013, pp. 1295-1298, doi: 10.1109/NER.2013.6696178.
10. Marquardt, James, et al. “Age and Gender Identification in Social Media.” *Proceedings of CLEF 2014 Evaluation Labs*, vol. 1180, 2014, pp. 1129–1136.