GROUP
7010

EXPRESSION DETECTION SYSTEM

ENHANCED WITH AGE, GENDER RECOGNITION AND DATABASE INTEGRATION

GUIDE:

MR. YOGI REDDY MARAMREDDY

ASSISTANT PROFESSOR, COMPUTER SCIENCE AND ENGINEERING, GST , HYDERABAD

OVERVIEW

SCOPE

- The Age-Gender-Emotion Prediction project aims to develop a system or model that can predict the age, gender, and emotional state of individuals based on various input sources such as images, videos, or text.
- Importance of incorporating age and gender recognition for a more nuanced understanding of emotions. Necessity to store data in a database for future analysis and longitudinal tracking.

APPLICATIONS

- MARKETING AND ADVERTISING
- CUSTOMER EXPERIENCE
- HEALTHCARE
- EDUCATION
- HUMAN-COMPUTER INTERACTION (HCI)
- SECURITY AND SURVEILLANCE
- ENTERTAINMENT AND GAMING

DATASETS

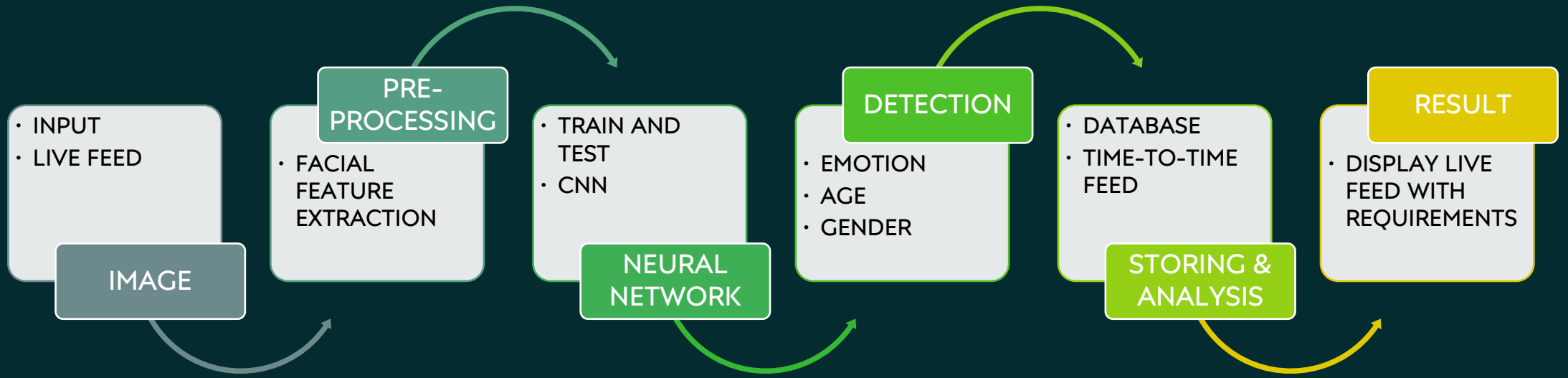
FER2013

- **Number of Images:** FER2013 contains a total of 35,887 images of faces, providing a diverse and comprehensive set of samples.
- **Emotion Categories:** The dataset includes seven emotion categories: Angry, Disgust, Fear, Happy, Neutral, Sad, and Surprise. These categories encompass a wide range of human emotions, making it suitable for training a robust emotion detection model.

UTKFACE

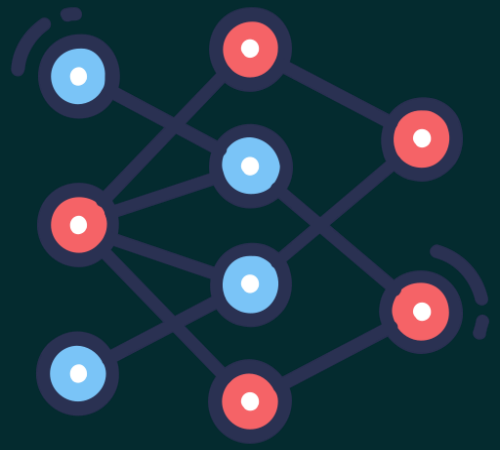
The UTKFace dataset is a large-scale face dataset with long age span (range from 0 to 116 years old). The dataset consists of over 20,000 face images with annotations of age, gender, and ethnicity. The images cover large variation in pose, facial expression, illumination, occlusion, resolution, etc.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

FLOWCHART


INPUT
LIVE FEED
IMAGE
PRE-PROCESSING
FACIAL FEATURE EXTRACTION
TRAIN AND TEST
CNN
NEURAL NETWORK
DETECTION
EMOTION
AGE
GENDER
DATABASE
TIME-TO-TIME FEED
STORING & ANALYSIS
RESULT
DISPLAY LIVE FEED WITH REQUIREMENTS

METHODOLOGIES

CNN MODELS



WEBCAM IMPLEMENTATION

DATABASE INTEGRATION

CNN MODELS

LIBRARIES USED

```
from keras.utils import to_categorical
from keras_preprocessing.image import load_img
from keras.models import Sequential
from keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D
import os
import pandas as pd
import numpy as np
```

EMOTION MODEL

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import cv2
from keras.models import Sequential,load_model,Model
from keras.layers import Conv2D,MaxPool2D,Dense,Dropout,BatchNormalization,Flatten,Input
from sklearn.model_selection import train_test_split
```

AGE-GENDER MODEL

```
from pathlib import Path
import cv2
import sys
import numpy as np
import argparse
from contextlib import contextmanager
from wide_resnet import WideResNet
from keras.utils import get_file
from keras.models import load_model
from keras.utils import img_to_array
from datetime import datetime
```

DRIVER

```python
def createdataframe(dir):
    image_paths = []
    labels = []
    for label in os.listdir(dir):
        for imagename in os.listdir(os.path.join(dir,label)):
            image_paths.append(os.path.join(dir,label,imagename))
            labels.append(label)
        print(label, "completed")
    return image_paths,labels
```

DATAFRAME CREATION

EMOTION MODEL

```python
model = Sequential()
# convolutional layers
model.add(Conv2D(128, kernel_size=(3,3), activation='relu', input_shape=(48,48,1)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(256, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(512, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(512, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Flatten())
# fully connected layers
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.3))
# output layer
model.add(Dense(7, activation='softmax'))
```

NEURAL NETWORK

FEATURE EXTRACTION

```python
def extract_features(images):
    features = []
    for image in tqdm(images):
        img = load_img(image,grayscale =  True )
        img = np.array(img)
        features.append(img)
    features = np.array(features)
    features = features.reshape(len(features),48,48,1)
    return features
```

SAVING FEATURES

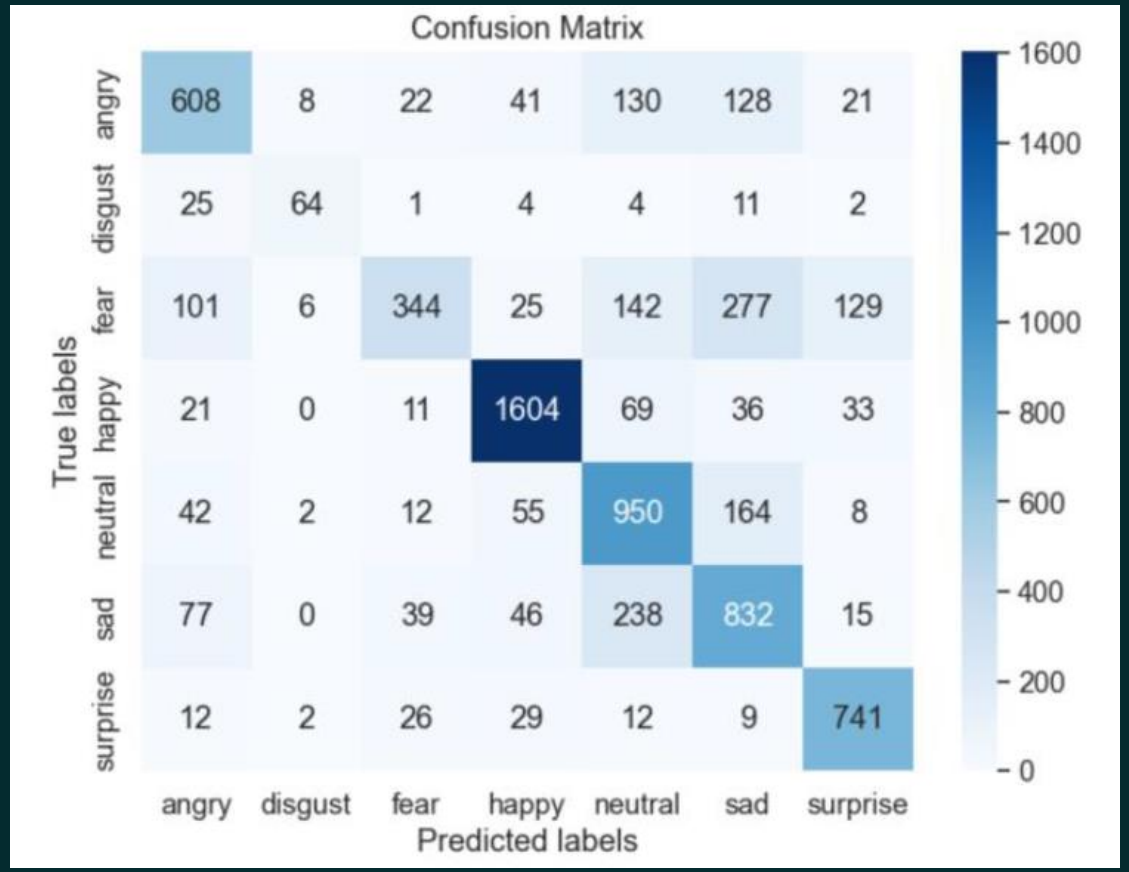
```python
from keras.models import model_from_json

model_json = model.to_json()
with open("emotiondetector.json",'w') as json_file:
    json_file.write(model_json)
model.save("emotiondetector.h5")


json_file = open("facialemotionmodel.json", "r")
model_json = json_file.read()
json_file.close()
model = model_from_json(model_json)
model.load_weights("facialemotionmodel.h5")
```

METRICS

EMOTION MODEL



ACCURACY & CLASSIFICATION REPORT

Accuracy: 0.7164948453608248

Classification Report:

	precision	recall	f1-score	support
angry	0.69	0.63	0.66	958
disgust	0.78	0.58	0.66	111
fear	0.76	0.34	0.47	1024
happy	0.89	0.90	0.90	1774
neutral	0.61	0.77	0.68	1233
sad	0.57	0.67	0.62	1247
surprise	0.78	0.89	0.83	831
accuracy			0.72	7178
macro avg	0.73	0.68	0.69	7178
weighted avg	0.73	0.72	0.71	7178

CONFUSION MATRIX

AGE-GENDER MODEL

DATAFRAME CREATION

```
images = []
age = []
gender = []
for img in os.listdir(path):
      ages = img.split("_")[0]
      genders = img.split("_")[1]
      img = cv2.imread(str(path)+"/"+str(img))
      img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
      images.append(np.array(img))
      age.append(np.array(ages))
      gender.append(np.array(genders))

age = np.array(age,dtype=np.int64)
images = np.array(images)
gender = np.array(gender,np.uint64)
```

```
gender_model = Sequential()
gender_model.add(Conv2D(36, kernel_size=3, activation='relu', input_shape=(200,200,3)))
gender_model.add(MaxPool2D(pool_size=3, strides=2))
gender_model.add(Conv2D(64, kernel_size=3, activation='relu'))
gender_model.add(MaxPool2D(pool_size=3, strides=2))
gender_model.add(Conv2D(128, kernel_size=3, activation='relu'))
gender_model.add(MaxPool2D(pool_size=3, strides=2))
gender_model.add(Conv2D(256, kernel_size=3, activation='relu'))
gender_model.add(MaxPool2D(pool_size=3, strides=2))
gender_model.add(Conv2D(512, kernel_size=3, activation='relu'))
gender_model.add(MaxPool2D(pool_size=3, strides=2))
gender_model.add(Flatten())
gender_model.add(Dropout(0.2))
gender_model.add(Dense(512, activation='relu'))
gender_model.add(Dense(1, activation='sigmoid', name='gender'))
gender_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
history_gender = gender_model.fit(x_train_gender, y_train_gender,
                                  validation_data=(x_test_gender, y_test_gender), epochs=1)

gender_model.save('gender_model_epochs.h5')
```

```
age_model = Sequential()
age_model.add(Conv2D(128, kernel_size=3, activation='relu', input_shape=(200,200,3)))
age_model.add(MaxPool2D(pool_size=3, strides=2))
age_model.add(Conv2D(128, kernel_size=3, activation='relu'))
age_model.add(MaxPool2D(pool_size=3, strides=2))
age_model.add(Conv2D(256, kernel_size=3, activation='relu'))
age_model.add(MaxPool2D(pool_size=3, strides=2))
age_model.add(Conv2D(512, kernel_size=3, activation='relu'))
age_model.add(MaxPool2D(pool_size=3, strides=2))
age_model.add(Flatten())
age_model.add(Dropout(0.2))
age_model.add(Dense(512, activation='relu'))
age_model.add(Dense(1, activation='linear', name='age'))
age_model.compile(optimizer='adam', loss='mse', metrics=['mae'])
print(age_model.summary())
history_age = age_model.fit(x_train_age, y_train_age,validation_data=(x_test_age, y_test_age), epochs=1)
age_model.save('age_model_epochs.h5')
```

NEURAL NETWORK & SAVING FEATURES

METRICS

AGE-GENDER MODEL

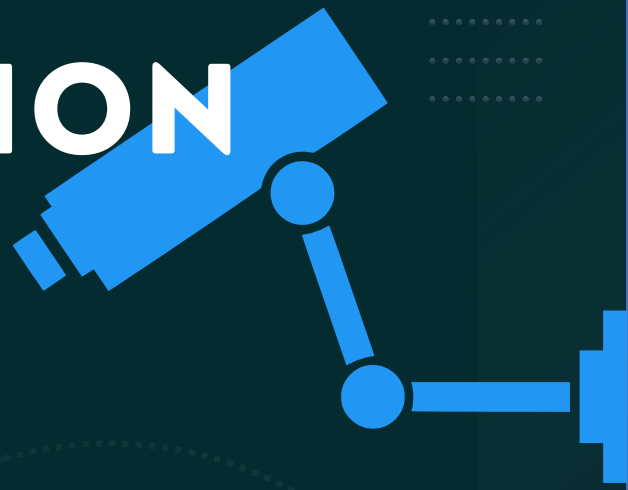
ACCURACY
F1 SCORE
PRECISION
RECALL
CONFUSION MATRIX
CLASSIFICATION REPORT

```
Accuracy: 0.795680782858107
F1 Score: 0.8068888534523999
Precision: 0.7354651162790697
Recall: 0.8936771458848464
Confusion Matrix:
[[2186  910]
 [ 301 2530]]
Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.71      0.78      3096
           1       0.74      0.89      0.81      2831

    accuracy                           0.80      5927
   macro avg       0.81      0.80      0.79      5927
weighted avg       0.81      0.80      0.79      5927
```

WEBCAM IMPLEMENTATION



- VIDEO INPUT ACQUISITION
- FACE DETECTION AND LOCALIZATION
- FEATURE EXTRACTION
- EMOTION RECOGNITION MODEL
- REAL-TIME FEEDBACK AND VISUALIZATION
- CONTINUOUS MONITORING AND ANALYSIS

DRIVER CODE

INTERFACE CODE

```
def draw_label(image, point, label, font=cv2.FONT_HERSHEY_SIMPLEX,font_scale=0.8, thickness=1):
    curr_time=str(datetime.now())
    size = cv2.getTextSize(label, font, font_scale, thickness)[0]
    x, y = point
    cv2.rectangle(image, (x, y - size[1]), (x + size[0], y), (255, 0, 0), cv2.FILLED)
    cv2.putText(image, label, point, font, font_scale, (255, 255, 255), thickness, lineType=cv2.LINE_AA)
    cv2.putText(image, curr_time, (x,y+160), font, font_scale, (255, 255, 255), thickness, lineType=cv2.LINE_AA)
```

LOADING MODEL WEIGHTS

```
# Define our model parameters
depth = 16
k = 8
weight_file = None
margin = 0.4
image_dir = None

# Get our weight file
if not weight_file:
    weight_file = get_file("weights.28-3.73.hdf5", pretrained_model, cache_subdir="pretrained_models",
                           file_hash=modhash, cache_dir=Path(sys.argv[0]).resolve().parent)
# load model and weights
img_size = 64
model = WideResNet(img_size, depth=depth, k=k)()
model.load_weights(weight_file)

detector = dlib.get_frontal_face_detector()
```

DRIVER CODE

DETECTING AND LABELLING

```
# Initialize Webcam
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    preprocessed_faces_emo = []
    input_img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    img_h, img_w, _ = np.shape(input_img)
    detected = detector(frame, 1)
    faces = np.empty((len(detected), img_size, img_size, 3))
    preprocessed_faces_emo = []
    if len(detected) > 0:
        for i, d in enumerate(detected):
            x1, y1, x2, y2, w, h = d.left(), d.top(), d.right() + 1, d.bottom() + 1, d.width(), d.height()
            xw1 = max(int(x1 - margin * w), 0)
            yw1 = max(int(y1 - margin * h), 0)
            xw2 = min(int(x2 + margin * w), img_w - 1)
            yw2 = min(int(y2 + margin * h), img_h - 1)
            cv2.rectangle(frame, (x1, y1), (x2, y2), (255, 0, 0), 2)
            # cv2.rectangle(img, (xw1, yw1), (xw2, yw2), (255, 0, 0), 2)
            faces[i, :, :, :] = cv2.resize(frame[yw1:yw2 + 1, xw1:xw2 + 1, :], (img_size, img_size))
            face =  frame[yw1:yw2 + 1, xw1:xw2 + 1, :]
            face_gray_emo = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)
            face_gray_emo = cv2.resize(face_gray_emo, (48, 48), interpolation = cv2.INTER_AREA)
            face_gray_emo = face_gray_emo.astype("float") / 255.0
            face_gray_emo = img_to_array(face_gray_emo)
            face_gray_emo = np.expand_dims(face_gray_emo, axis=0)
            preprocessed_faces_emo.append(face_gray_emo)
```

DRIVER CODE

PREDICTION AND SAVING

```
        # make a prediction for Age and Gender
        results = model.predict(np.array(faces))
        predicted_genders = results[0]
        ages = np.arange(0, 101).reshape(101, 1)
        predicted_ages = results[1].dot(ages).flatten()

        # make a prediction for Emotion
        emo_labels = []
        for i, d in enumerate(detected):
            preds = classifier.predict(preprocessed_faces_emo[i])[0]
            emo_labels.append(emotion_classes[preds.argmax()])

        save_to_excel(emo_labels)
        # draw results
        for i, d in enumerate(detected):
            label = "{}, {}, {}".format(int(predicted_ages[i]-9),
                                        "F" if predicted_genders[i][0] > 0.4 else "M", emo_labels[i])
            draw_label(frame, (d.left(), d.top()), label)

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

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

DATABASE INTEGRATION

EMOTION ANALYSIS INTEGRATION

TIMESTAMP INCLUSION

ENHANCED DATA VISUALIZATION

ACTIONABLE INSIGHTS

CONTINUOUS IMPROVEMENT

DATABASE INTEGRATION

SAVING TO EXCEL (DATABASE)

```python
def save_to_excel(emo_labels):
    current_time = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    if not os.path.isfile('emotion_data.xlsx'):
        wb = Workbook()
        ws = wb.active
        ws.append(["Emotion", "Time"])
    else:
        wb = load_workbook('emotion_data.xlsx')
        ws = wb.active

    for emo_label in emo_labels:
        ws.append([emo_label, current_time])
    wb.save('emotion_data.xlsx')
```

E16

	A	B	C	D	E	F	G
1	**Emotion**	**Time**					
2	happy	2024-03-25 13:47:40					
3	happy	2024-03-25 13:47:41					
4	neutral	2024-03-25 13:47:41					
5	happy	2024-03-25 13:47:42					
6	happy	2024-03-25 13:47:43					
7	happy	2024-03-25 13:47:43					
8	happy	2024-03-25 13:47:44					
9	neutral	2024-03-25 13:47:44					
10	happy	2024-03-25 13:47:45					
11	neutral	2024-03-25 13:47:45					
12	happy	2024-03-25 13:47:46					
13	neutral	2024-03-25 13:47:46					
14	happy	2024-03-25 13:47:47					
15	neutral	2024-03-25 13:47:47					
16	happy	2024-03-25 13:47:47					
17	neutral	2024-03-25 13:47:47					
18	happy	2024-03-25 13:47:48					
19	neutral	2024-03-25 13:47:48					
20	fear	2024-03-25 13:47:49					
21	neutral	2024-03-25 13:47:49					
22	fear	2024-03-25 13:47:50					

DISPLAYING THE DATA FOR FURTHUR ANALYSIS

RESULTS

ANGER

FEAR

SURPRISE

NEUTRAL

SAD

HAPPY

ADVANCEMENTS

- IMPROVE ACCURACY AND FRAME RATE
- CLOUD BASED DATABASE
- EXTENDING SYSTEM TO BEHAVIOURAL ANALYSIS
- CAN ADD AUDIO MODELS FOR BETTER ANALYSIS
- ETHICAL CONSIDERATION

CONCLUSION

- OUR PROJECT MARKS A SIGNIFICANT MILESTONE IN THE REALM OF FACIAL EXPRESSION ANALYSIS AND RECOGNITION, AMPLIFIED BY THE INTEGRATION OF AGE AND GENDER RECOGNITION CAPABILITIES AND SEAMLESS DATABASE INTEGRATION USING EXCEL.

- THIS MULTIFACETED SYSTEM EMBODIES A SYNERGY OF STATE-OF-THE-ART TECHNOLOGIES AND METHODOLOGIES, PROVIDING A ROBUST FOUNDATION FOR UNRAVELING THE COMPLEXITIES OF HUMAN EMOTIONS AND DEMOGRAPHICS.

REFERENCES

[1] Deep Learning Approach For Human Emotion-Gender-Age Recognition by Suraj S, Department of Computer science & Engineering , PES University Bangalore, India

[2] John, Ansamma, M. C. Abhishek, Ananthu S. Ajayan, S. Sanoop, and Vishnu R. Kumar. "Real-Time Facial Emotion Recognition System With Improved Preprocessing and Feature Extraction." In 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 1328 1333. IEEE, 2020.

[3] Noroozi, Fatemeh, Marina Marjanovic, Angelina Njegus, Sergio Escalera, and Gholamreza Anbarjafari. "Audio-visual emotion recognition in video clips." IEEE Transactions on Affective Computing 10, no. 1 (2017): 60-75.

THANK YOU

J RUSHIL MANIDHAN
222010303042

PVN JAYANTH RAJ
222010303048

K RUTHVICK
222010303054

G SUPRITHA
222010303058