Face Identification and Unlock System

Date: 21 July 2025

Prepared by: Ammash Binti Ishrat

import face\_recognition

import os

def encode\_known\_faces():

known\_face\_encodings = []

known\_face\_names = []

# Load images from 'known\_faces' folder

for filename in os.listdir("known\_faces"):

image = face\_recognition.load\_image\_file(f"known\_faces/{filename}")

encoding = face\_recognition.face\_encodings(image)[0]

known\_face\_encodings.append(encoding)

known\_face\_names.append(filename.split(".")[0]) # Remove file extension

return known\_face\_encodings, known\_face\_names

if \_\_name\_\_ == "\_\_main\_\_":

encodings, names = encode\_known\_faces()

print(f"Encoded {len(encodings)} faces: {names}")

import face\_recognition

import cv2

import numpy as np

from encode\_known\_faces import encode\_known\_faces

def main():

# Load known faces

known\_face\_encodings, known\_face\_names = encode\_known\_faces()

video\_capture = cv2.VideoCapture(0)

while True:

ret, frame = video\_capture.read()

if not ret:

break

# Convert BGR (OpenCV) to RGB (face\_recognition)

rgb\_frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

face\_locations = face\_recognition.face\_locations(rgb\_frame)

face\_encodings = face\_recognition.face\_encodings(rgb\_frame, face\_locations)

for (top, right, bottom, left), face\_encoding in zip(face\_locations, face\_encodings):

matches = face\_recognition.compare\_faces(known\_face\_encodings, face\_encoding)

name = "Unknown"

face\_distances = face\_recognition.face\_distance(known\_face\_encodings, face\_encoding)

best\_match\_index = np.argmin(face\_distances)

if matches[best\_match\_index]:

name = known\_face\_names[best\_match\_index]

cv2.putText(frame, f"Access Granted: Welcome, {name}", (left, top - 20),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.8, (0, 255, 0), 2)

else:

cv2.putText(frame, "Access Denied", (left, top - 20),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.8, (0, 0, 255), 2)

# Draw face bounding box

cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)

# Display result

cv2.imshow('Face Unlock System', frame)

# Quit on 'q' key press

if cv2.waitKey(1) & 0xFF == ord('q'):

break

video\_capture.release()

cv2.destroyAllWindows()

if \_\_name\_\_ == "\_\_main\_\_":

main()

python encode\_known\_faces.py

python face\_unlock\_system.py