**G.O.D.S.E.Y.E.S**

**(Guided Object Detection and Surveillance for Enhanced Yielding Estimation System)**

G.O.D.S.E.Y.E.S is an intelligent system designed to use advanced computer vision algorithms for face and pose detection. Its primary function is to recognize faces, and map body landmarks, providing detailed analytics for monitoring and estimation purposes. It leverages OpenCV, face\_recognition, and Mediapipe libraries to deliver high-accuracy performance in real-time video feed analysis.

**Getting Started**

These instructions will guide you through the installation process and how to run the program.

**Prerequisites**

* Python 3.8 or newer.
* The following Python libraries: os, cv2, numpy, face\_recognition, datetime, mediapipe, pyttsx3, playsound.
* To install these, use pip:

pip install opencv-python numpy face\_recognition datetime mediapipe pyttsx3 playsound

**Installation**

* Clone this repository or download as a zip file and extract it.
* Install the required Python libraries.
* Run the resulting run\_godseyes.exe file to execute the godseyes.py script.

**Usage**

* Click on run\_godseyes.exe to run the program. It will use your system's default camera for face detection and pose estimation. When it detects a known face, it will log the time and the name of the person in a CSV file named database.csv.
* Known faces and their encoding should be stored in the ./faces directory. Make sure to add the images of people you want to recognize in this directory. The name of the image file (excluding the extension) will be used as the person's name.

**Notes**

* Ensure that the images in ./faces directory are clear and the face is visible for best results.
* The program will recognize a person even if they have been recognized before. Please modify the code if you want to recognize a person only once.
* If the program does not recognize a face, it will mark it as "Unknown" and log it in the database.csv file.
* Press ESC to exit the program.

**Built With**

* Python - Programming language used.
* face\_recognition - Library used for face detection and recognition.
* mediapipe - Library used for real-time pose estimation.
* pyttsx3 - Library used for text-to-speech conversion.

**Introduction**

Discrete Mathematics is the backbone of Computer Science and plays a vital role in the development of algorithms, data structures, and software applications. The Python script provided for the "G.O.D.S.E.Y.E.S" system, which stands for Guided Object Detection and Surveillance for Enhanced Yield Evaluation System, offers a concrete example of how various principles of Discrete Mathematics are applied in real-world programming.

**Body**

1. **Sets**: In the script, sets, which are an integral part of Discrete Mathematics, are used for storing unique names in the database. This illustrates the practical use of set theory, which is a fundamental concept in Discrete Mathematics.

NameList = set()

1. **Functions and Relations**: The script encapsulates different functionalities within various functions. Functions represent a special kind of relation, and their use here mirrors their mathematical counterpart.

def get\_encoded\_faces():

1. **Logic and Proposition**: The script uses propositional logic, another key area of Discrete Mathematics, within its conditional statements (if-else). The logic operators contribute to decision-making based on whether certain conditions are true or false.

if name in Male\_Names:

        Honorific\_Address = "Sir"

    elif name in Female\_Names:

        Honorific\_Address = "Ma'am"

1. **Counting Techniques**: The script employs counting techniques when it calculates the number of faces detected in each frame. This practical implementation mirrors the counting principles used in Discrete Mathematics to enumerate elements within a set.

count = len(face\_locations)

font = cv2.FONT\_HERSHEY\_TRIPLEX

        cv2.putText(frame, f"Faces: {count}", (10, 30), font, 0.8, (0, 255, 0), 2, cv2.LINE\_AA)

1. **Boolean Algebra**: Boolean algebra is utilized when comparing the faces and deciding if a condition is true or false. This kind of binary decision-making process is an integral part of Discrete Mathematics.

matches = fr.compare\_faces(faces\_encoded, face\_encoding)

1. **Matrices**: Although not directly noticeable, the use of matrices is implicit in the handling of images in the script. Each image is effectively a 2D matrix (or 3D for color images), and operations like resizing or flipping the image are actually matrix operations.

frame = resize(frame, 0.70)

        frame = cv2.flip(frame, 1)

**Conclusion**

In conclusion, the Python script for the G.O.D.S.E.Y.E.S system exemplifies the implementation of various concepts from Discrete Mathematics in computer programming. Whether it's the use of sets for storing unique names, the use of functions for encapsulating functionalities, applying logic for decision making, using counting techniques to enumerate elements, or handling images as matrices, the script is a testament to the pervasive influence of Discrete Mathematics in the field of Computer Science and Information Technology.