# Face Recognition System

## Introduction

Face recognition system refers to a system that identifies a person's identity based on the facial analysis. In today's world, this system is a day-to-day part of most individuals. Be it from unlocking a phone to creating a live security system, face recognition is becoming an inevitable task. It is used in security and surveillance, healthcare, retail and marketing, law enforcement etc. The system involves capturing a face, analyzing its features, and matching those features with the faces already in the database.

### Advantages:

- ***Convenient***: Using face for unlocking devices, taking attendance etc., is convenient compared to doing these tasks manually.
- ***Non-invasive***: Face recognition systems do not need any physical contact with the device and hence are non-invasive in nature.
- ***Security***: Face recognitions systems provide security and authentication capabilities.

### Disadvantages:

- ***Privacy issues***: For face recognition systems to work efficiently, images from individuals need to exist in the database. This makes the data vulnerable against malicious attacks and raises privacy concerns, as breach in data may put individuals in harm's way.
- ***Accuracy and bias***: For efficient recognition optimal environment needs to exist, like proper lighting conditions etc. or database needs to be more precise which involves creating a large database. In addition, bias regarding different ethnicities in facial recognition systems is a frequent problem.
- ***Ownership***: Face recognition systems are generally owned by a third party and when giving consent to facial recognition, users might not realize that their data is being used by a third party.

# Methodology

**Key steps involved in face recognition:**

1. **Face Detection:** This step involves detecting a face or faces from camera or a photograph. It is a better to face directly at a camera and ensure proper lighting.
2. **Face Analysis:** This step involves detecting various features from the captured face like eye dimensions, distance between eyes, etc.
3. **Encoding:** Once the features have been detected, the face is then encoded to create a face print.
4. **Face Match:** The encoded face is compared to the faces in the database. If the face exists in the database, the function returns true.
5. **Output:** If true in the face match, output returns the name of the corresponding match.

# Implementation

## *Requirements (Modules):*

- Pickle
- Open-cv
- Face recognition
- Streamlit

## *Procedure:*

1. **Create Database:** Collect images to train the model for face recognition. Serialize the dataset using pickle module.
2. **Face Encoding:**
    a. Convert images to a numpy array using load image file function of face recognition library.
    b. Encode using face encodings function.
    c. Save to pickle file.
3. **Load Database:** Create a variable to store dataset of known faces.
4. **Face Match:**
    a. Take input from the user in the form of a photo or a captured image.

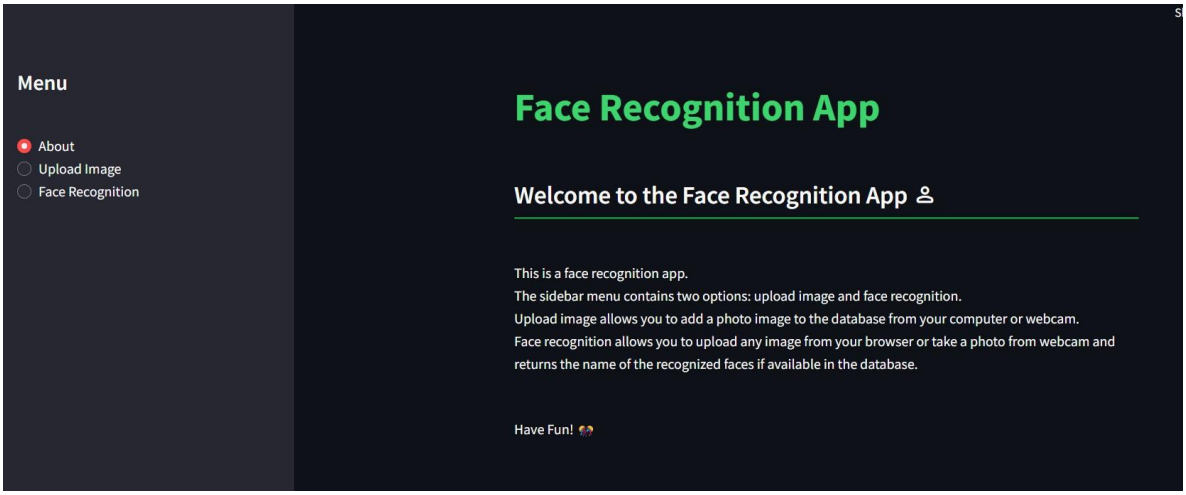
b. Find locations of all the faces in the uploaded file using face locations function of the face recognition module.
c. Encode faces in the detected locations using face encodings function of the face recognition module.
d. Compare faces based on the Euclidean distance between the encoded known faces and the detected faces in the uploaded image and the given threshold value using face compare function of the face recognition module.

5. **Output:** If the Euclidean distance between the vectors >= threshold value, aka tolerance, return the name of the known face from the dataset and display the image and the name using open-cv module.

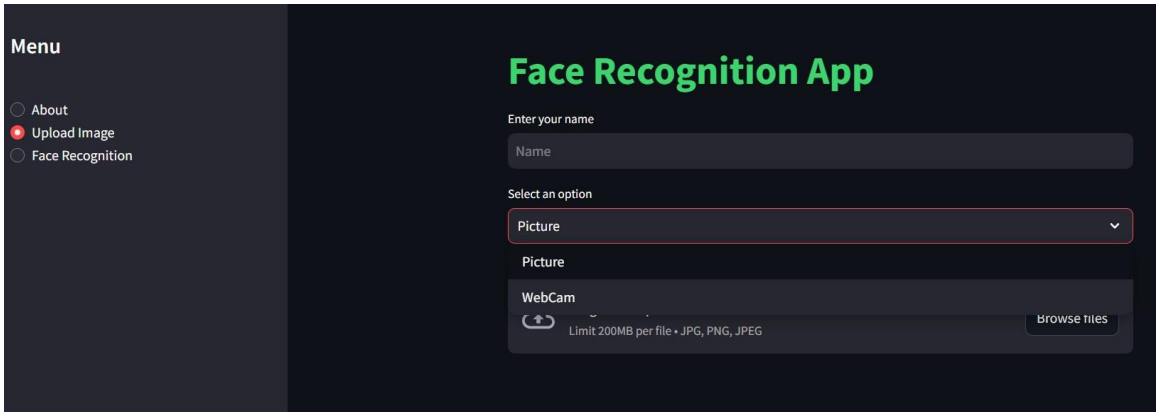
# Streamlit Application

The face recognition model has been deployed using streamlit platform. The application contains three radio options.

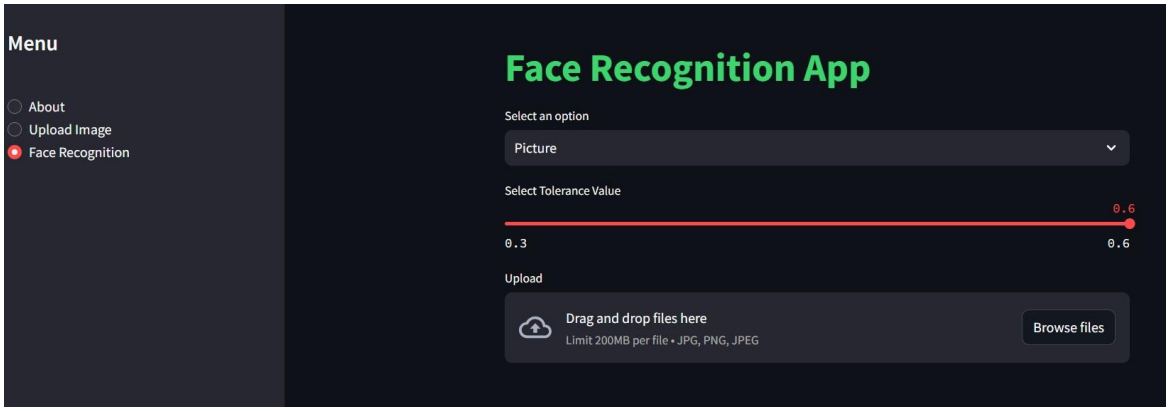
- ***About***: This option provides information about the app.




- ***Upload Image***: This option allows the users to upload images in the database for recognition task.




- ***Face Recognition***: This option allows the user to upload an image or take a photo and return the name, if the image is known or, returns "unknown" if the image is not known.




# Conclusion

Face recognition systems come with advantages as well as disadvantages. They may be tricky to use given the nature of their usage. Like other machine learning tasks, they require advanced computational setups for better performance. In this project, I have created a simple face recognition system that reflects the basics of face recognition task. Although, it is not highly accurate given the computational and data limitations, it reflects how face recognition systems work. This system can be improved using a large database along with cnn model detection of dlib library.