**OSEI ROY JAKE SARFO – 9415319**

**AMO GILBERT - 9395819**

**PROJECT PROPOSAL**

**TOPIC**

FACIAL EXPRESSION RECOGNITION INTELLIGENT SECURITY SYSTEM FOR REAL TIME SURVEILLANCE.

**INTRODUCTION**

Facial expression recognition is a part of biometric authentication that focuses on uniquely recognizing human facial expressions based upon one or more intrinsic physical or behavioural traits and inside emotions portrayed on one’s face. This project is based on improving current surveillance systems by adding facial expression recognition to make a system that will detect a person’s expression and

The system captures ones feelings through a camera and send the footages for preprocessing before it can be extracted to see if the emotion/expression captured has the intentions of causing harm.

The system is to be used in examination centres, offices and places of gathering that require surveillance.

**PROBLEM STATEMENT**

The proliferation of AI has resulted in a surge of IoT – enabled systems for healthcare analytics, activity recognition and smart energy consumption. Since public safety is still a significant concern in many developing countries, security and surveillance are essential components of an IoT – enabled smart home. Strangers or intruders who are overly aggressive towards vulnerable groups, such as women and children, commit most of these crimes, including break-ins, physical/sexual assaults, and rioting. The failure to identify the intentions of these intruders, which facilitates their evil designs. To tackle this alarming issue, a smart security and surveillance system that predicts human intentions can play a critical role in ensuring the safety of the people.

**AIM OF PROJECT**

The aim of the project is to develop a facial expression recognition intelligence system that can detect and interpret facial expressions in real-time. The resulting program will be used to provide valuable insights into the behaviour and mood of individuals in monitored areas, which can be used for security and surveillance purposes.

**SPECIFIC OBJECTIVES**

* To develop an intelligent system that can accurately detect facial expressions
* To train the system using a large dataset of facial expressions
* To implement the system in real-time surveillance scenarios
* To provide detailed documentation and user manuals for the program

**JUSTIFICATION FOR PROJECT**

The expected users of the facial expression recognition intelligence system include security and surveillance agencies, law enforcement agencies, and private organizations that require monitoring of their premises. The system can assist in the early detection of potential security threats, as well as provide valuable insights into the behaviour of individuals in monitored areas.

**MOTIVATION FOR PROJECT**

The motivation for the project to develop the facial expression recognition intelligence system is because of its potential to enhance security and surveillance. The system can provide valuable insights into the behaviour and mood of individuals in monitored areas, which can assist in the early detection of potential security threats.

**SCOPE OF PROJECT**

The project is expected to be completed within a timeline of 6 months. The project will require the use of high-performance computing resources and a large dataset of facial expressions. The project team will consist of experienced software developers and data scientists.

**METHODOLOGY**

The project would be divided into the following steps:

**1. Emotion data acquisition.**

We will be using FER 2013 dataset which consists of people with distinct facial features like beard and moustache, different ethnic backgrounds, and varied facial complexions.

**2. Image to arrays**

The array module in NumPy ( nd.array ) is used to convert an image into an array and obtain the image attribute. So, we convert images with their respective attribute in pixels to a 2 dimensions and size 48 x 48 pixels.

**3. Image to landmarks**

The haar cascade library is used to detect facial landmarks. This process consists of two steps, localize the face in an image and detect the facial landmarks.

**4. CNN architecture**

Machine learning models can be built and trained easily using a high-level Application Programming Interface(API) like Keras. In this project, a sequential CNN model is developed using Tensor flow with Keras API since it allows a model to be built layer by layer.

**5. Compiling the model**

Compiling the model requires two parameters, optimizer and metrics. The optimizer used is Adam. The optimizer is used to update the weights in a DL model based on the loss. The metrics used are accuracy,categorical cross-entropy loss, precision, recall and F-score.

**6. Training the model**

To train the model, a function is used to split the dataset into training and testing sets. A training ratio of 0.90 means 90% of the dataset will be used or training and the remaining for testing the model.

**7. Real-time face detection**

The facial visual information of a person is used for emotion analysis; therefore, accurate face detection in real-time with low latency is of prime importance. In this project, the haar facial cascade algorithm is adopted for the face detection module of the system. During real-time validation, suppose, in a video sequence frame there exists a person in the video frame, the haar facial cascade algorithm is applied to extract facial image from the frame using the Haar facial feature-based cascade classifier. The area of interest (AOI), i.e., the face is first localized, and a four -corner bounding box is formed to crop the AOI. The cropped facial image is fed into the system, which sends feedback after appropriate analysis.

**8. Emotion analysis strategy**

The facial emotion analysis system is triggered when the camera detects a person in the vicinity of the building, followed by the human face detection by the face detection module. Since a hostile or dangerous situation is governed by common expressions such as anger or aggression, the facial expression analysis for such negative valence is conducted precisely as long as the face is in the frame. The security module is invoked if aggressive or angry faces are captured.

The tools that will be used in the development of the system include Python programming language, TensorFlow machine learning library, OpenCV computer vision library.

**PROJECT RESULT**

The resulting software will be a facial expression recognition intelligence system that can accurately detect and interpret facial expressions in real-time. The system will be able to provide valuable insights into the behaviour and mood of individuals in monitored areas, which can be used for security and surveillance purposes. The software will be accompanied by detailed documentation and user manuals, as well as other forms of documents that would be usable to the user.

**REQUIREMENT DOCUMENT**

**User Functional Requirements:**

* The software should be able to detect and recognize facial expressions in real-time.
* The software should be able to identify facial expressions such as happiness, sadness, anger, fear, surprise, disgust, and neutral expressions.
* The software should be able to work with live video feeds from surveillance cameras.
* The user should be able to set up the software easily and configure the system to work with different camera models.
* The software should provide an easy-to-use interface for viewing and analyzing real-time and stored data.

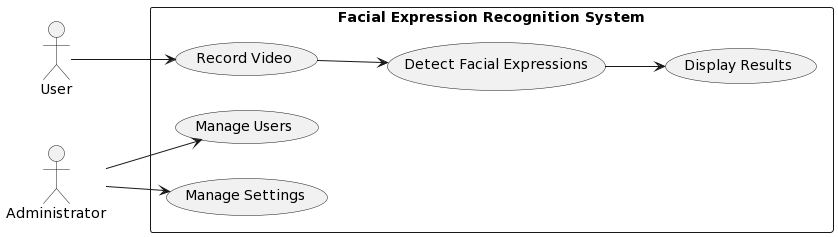
**System Functional Requirements:**

* The software should be able to capture live video feeds from surveillance cameras.
* The software should use advanced algorithms to detect and recognize facial expressions in real-time.
* The software should be able to process large amounts of data quickly and efficiently.
* The software should be able to handle multiple video feeds simultaneously.
* The software should be able to provide real-time alerts to the user when a facial expression of interest is detected.

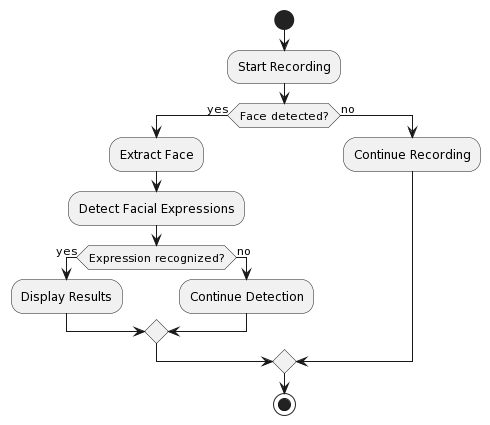
**Non-functional Requirements:**

* **Performance:** The software should be able to process video feeds and detect facial expressions in real-time with minimal delay.
* **Security:** The software should be secure and protect user data from unauthorized access.
* **Scalability:** The software should be able to handle an increasing number of cameras and users without compromising its performance.
* **Usability:** The software should have a user-friendly interface that is easy to navigate and use.
* **Maintainability:** The software should be easy to maintain and update as needed.

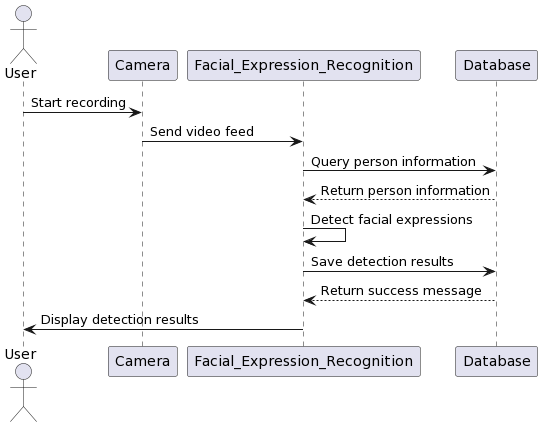
Use Case Diagram:



Activity Diagram:

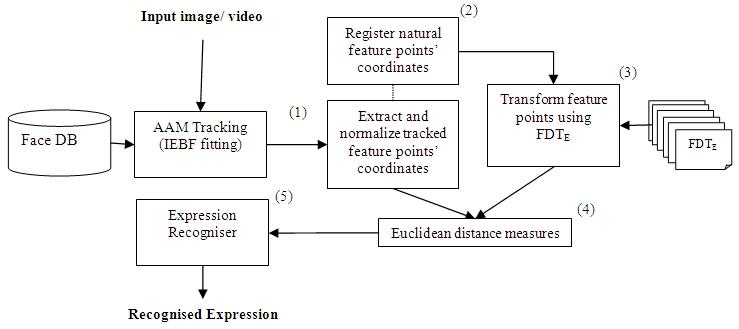


Sequence diagram:

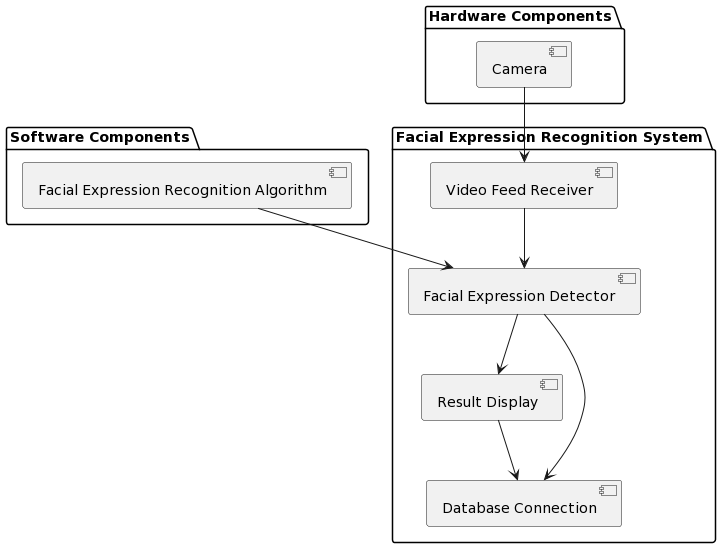


**PROJECT ARCHITECTURE**

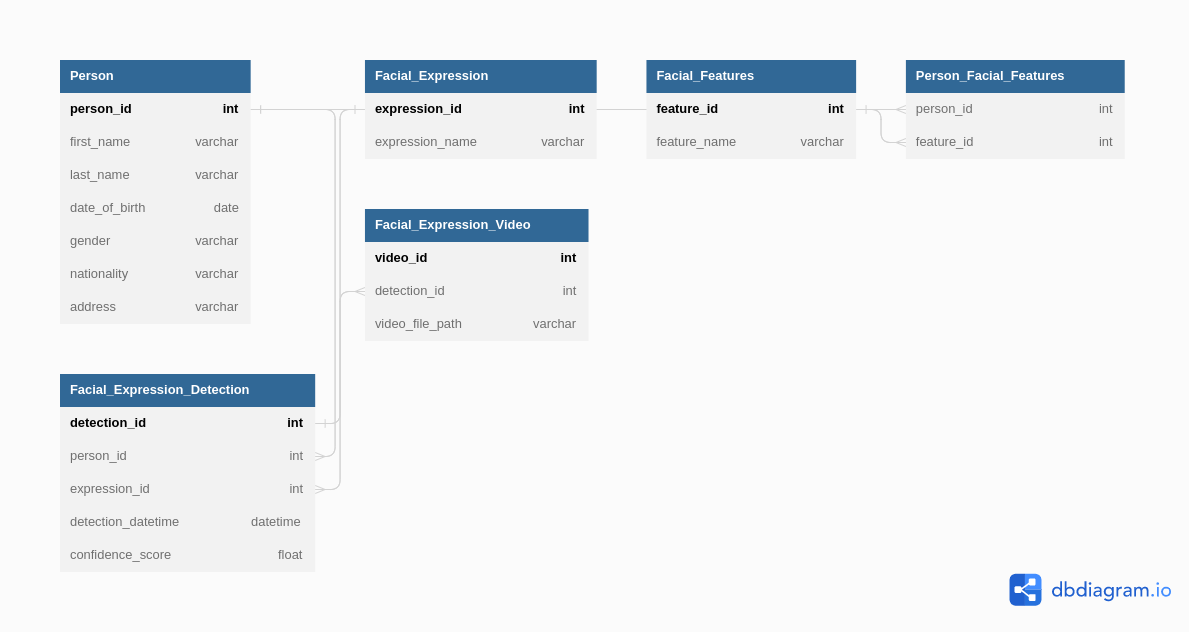
**1. ARCHITECTURE DESIGN:**

****

**2. COMPONENT DESIGN:**

****

**3. DATABASE SCHEMA:**

****

**ALGORITHM**

* Install the FER library and its dependencies.
* Import the necessary libraries, including cv2, numpy, and fer.
* Load the pre-trained model using the fer.Model() method.
* Capture an image or video stream using the cv2.VideoCapture() method.
* Convert the image to grayscale using the cv2.cvtColor() method.
* Detect the faces in the image using the fer.detect\_faces() method.
* For each face detected, extract the bounding box coordinates using the fer.extract\_face() method.
* Preprocess the face image by resizing it to the required dimensions using the fer.resize() method.
* Use the pre-trained model to predict the emotion from the preprocessed face image using the fer.predict() method.
* Draw a rectangle around each detected face using the cv2.rectangle() method.
* Write the predicted emotion label on top of the rectangle using the cv2.putText() method.
* Display the image with the detected faces and their corresponding emotion labels using the cv2.imshow() method.

**FLOW CHART**

