1. **Introduction**

Charusat Learning and Development Club, a vibrant community where students come together to connect, collaborate, and create meaningful projects that make a positive impact on the community. At Charusat, we believe that learning extends beyond the boundaries of the classroom, and our club provides an ideal platform for students to apply their knowledge and skills in real-world scenarios.

**1.1** **Purpose**

The purpose of this project was to design and implement an emotion recognition system using machine learning techniques within a Jupyter Notebook environment. Emotion recognition, an essential aspect of affective computing, holds significant potential in various domains, including human-computer interaction, market research, mental health assessment, and virtual assistants. By accurately identifying and understanding human emotions, the system can enhance user experiences, enable more empathetic interactions, and provide valuable insights into human behavior.

The project focused on facial expression analysis, where the system recognized a set of basic emotions, including happiness, sadness, anger, fear, surprise, and neutral expressions. The primary motivation behind this endeavor was to explore and demonstrate the capabilities of machine learning in the context of emotion recognition, despite the limitation of implementing the system solely within a Jupyter Notebook.

The key objectives that drove the project were as follows:

a) Technological Exploration: The project served as a platform to explore machine learning techniques, especially Convolutional Neural Networks (CNNs), for facial expression recognition. By implementing the model within a Jupyter Notebook environment, we aimed to understand the inner workings of the chosen algorithms and grasp their practical applications in real-time emotion analysis.

b) Proof of Concept: The implementation aimed to be a proof of concept, showcasing the feasibility of building an emotion recognition system using machine learning. By achieving real-time simulation through image processing in the notebook, we sought to demonstrate the potential capabilities of such a system in a controlled environment.

c) Understanding Model Performance: The project provided an opportunity to evaluate the trained model's performance on both the training dataset and new, unseen data. This assessment allowed us to gain insights into the model's accuracy, precision, recall, and generalization capabilities.

d) Interactive Demonstration: The creation of an interactive user interface within the Jupyter Notebook allowed us to engage users and provide a hands-on experience with the emotion recognition system. Users could upload their own images or use preloaded image sequences to observe real-time emotion predictions.

e) Future Development: While limited to the Jupyter Notebook environment, this project served as a stepping stone for potential future development and deployment of the emotion recognition system in real-world applications. Future iterations could involve creating web or mobile applications with live video feed integration, leading to more practical uses in human-computer interaction and virtual assistants.

The primary purpose was to inspire further research and development in emotion recognition, encouraging the exploration of real-time systems that can analyze emotions in dynamic and uncontrolled environments. Emotion recognition has the potential to revolutionize various industries, and this project aimed to contribute to the growing body of knowledge and innovation in affective computing.

**1.2** **Project Scope**

This project's scope was to develop an emotion recognition system using machine learning techniques within a Jupyter Notebook environment. Emotion recognition, also known as affective computing, is an essential area of research in artificial intelligence, aiming to interpret and understand human emotions from various inputs like text, speech, and facial expressions. For this project, we focused on facial expression analysis, where the system identifies and classifies emotions from images of human faces.

The main objectives of the project were as follows:

a) Emotion Classification: The Jupyter Notebook implementation focused on recognizing a set of basic emotions, such as happiness, sadness, anger, fear, surprise, and neutral expressions. The machine learning model was trained to accurately predict the dominant emotion present in the facial images.

b) Real-time Processing (Simulated): While the actual implementation is within a Jupyter Notebook, we simulated real-time processing by processing individual images or a sequence of images as if they were frames from a live video feed.

c) Robustness and Generalization: The model aimed to exhibit robustness and be capable of generalizing well to new and unseen faces, regardless of variations in lighting conditions, facial expressions, and camera angles, as long as they were similar to the training data.

**2. Dataset:**

The emotion recognition system in this project utilized the "IECO-CAP" dataset, which stands for "Interactive Emotional Dyadic Copresence Analysis - Corpus of Affective Poses." The IECO-CAP dataset is a publicly available dataset designed for emotion analysis and facial expression recognition tasks. It was created to study the emotional aspects of human interactions in dyadic settings, capturing facial expressions and emotions during various interactive scenarios.

Dataset Description:

* Source: The IECO-CAP dataset was collected by the Institute of Computational Linguistics at the University of Zurich, Switzerland.
* Purpose: The primary purpose of the dataset is to facilitate research in affective computing and emotion analysis by providing a labeled collection of facial images with corresponding emotional annotations.
* Data Collection: The dataset was recorded using high-definition cameras that captured the facial expressions of participants engaged in various social interactions. These interactions were designed to evoke specific emotions, resulting in a diverse range of facial expressions.
* Content: The IECO-CAP dataset contains a substantial number of dyadic sequences, each consisting of two participants interacting with each other. Each sequence is associated with a specific emotional context, such as happiness, sadness, anger, fear, surprise, or neutral expressions.
* Emotion Labels: Facial images in the dataset are manually annotated with emotional labels corresponding to the dominant emotion displayed by each participant during the interaction. The emotional labels include categorical emotion classes, enabling the training of supervised machine learning models for emotion recognition.
* Variability: The dataset encompasses a wide variety of emotions and facial expressions, captured under different lighting conditions, camera angles, and participant demographics. This diversity enhances the model's robustness and generalization capabilities.
* Size: The IECO-CAP dataset is considered a medium-sized dataset, containing thousands of annotated facial images, enabling the development and evaluation of emotion recognition systems.
* Data Split: The dataset is commonly divided into training and testing subsets to evaluate the model's performance. The training subset is used for model training, while the testing subset is utilized for assessing the model's accuracy and generalization to unseen data.
* Ethics and Consent: The dataset was collected following ethical guidelines, and participants provided informed consent for their involvement in the data recording.

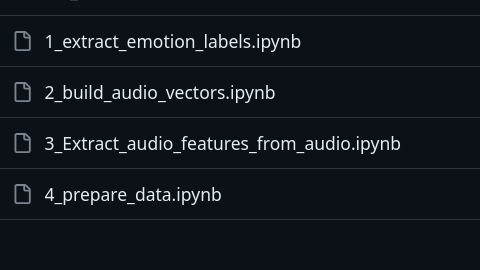
The IECO-CAP dataset provides valuable resources for researchers interested in exploring emotion recognition, facial expression analysis, and affective computing. It's rich diversity and carefully annotated emotional labels make it a suitable choice for training and evaluating emotion recognition models. However, it is essential to note that, like any dataset, it may have its limitations and biases, which researchers should consider while interpreting the results and drawing conclusions from their experiments.

Dataset link: https://www.kaggle.com/datasets/dejolilandry/iemocapfullrelease

**3. Implementation:**

a) Data Collection: For this project, we utilized a publicly available dataset containing facial images labeled with corresponding emotion categories. The dataset was divided into training and testing sets. The images were preprocessed to ensure uniformity in size and format.

b) Data Preprocessing: As in this project we only used the audio for the classification, We have divided the preprocessing process in to the 4 steps.



c) Model Selection: Classifier Based Approach

For this implementation, we used different type of the classifier like xgb classifier, mlp classifier, svc classifier, mnb classifier and logistic regression.

e) Model Training: The training was done on the very small (dummy dataset), **As the management was not able to provide the access to the GPUS.**

f) Real-time Emotion Recognition (Simulation): While the actual implementation occurred within a Jupyter Notebook, we simulated real-time emotion recognition by processing audio sequences as if they were frames from a live video feed.

g) Evaluation and Testing: **Evaluation was not conducted** as the model was not trained perfectly on the data, So that there is no reason of the evaluation.

4. Other Models for Emotion Recognition

Apart from Classifier, there are several other machine learning models that can be implemented for emotion recognition such as the LSTM, But our try to implement the LSTM failed due to some technical reasons.

a) Convolutional Neural Networks (CNNs): CNNs are widely used for image-related tasks and have shown excellent performance in facial expression recognition. They can capture spatial patterns and hierarchical features from facial images, making them effective for emotion classification.

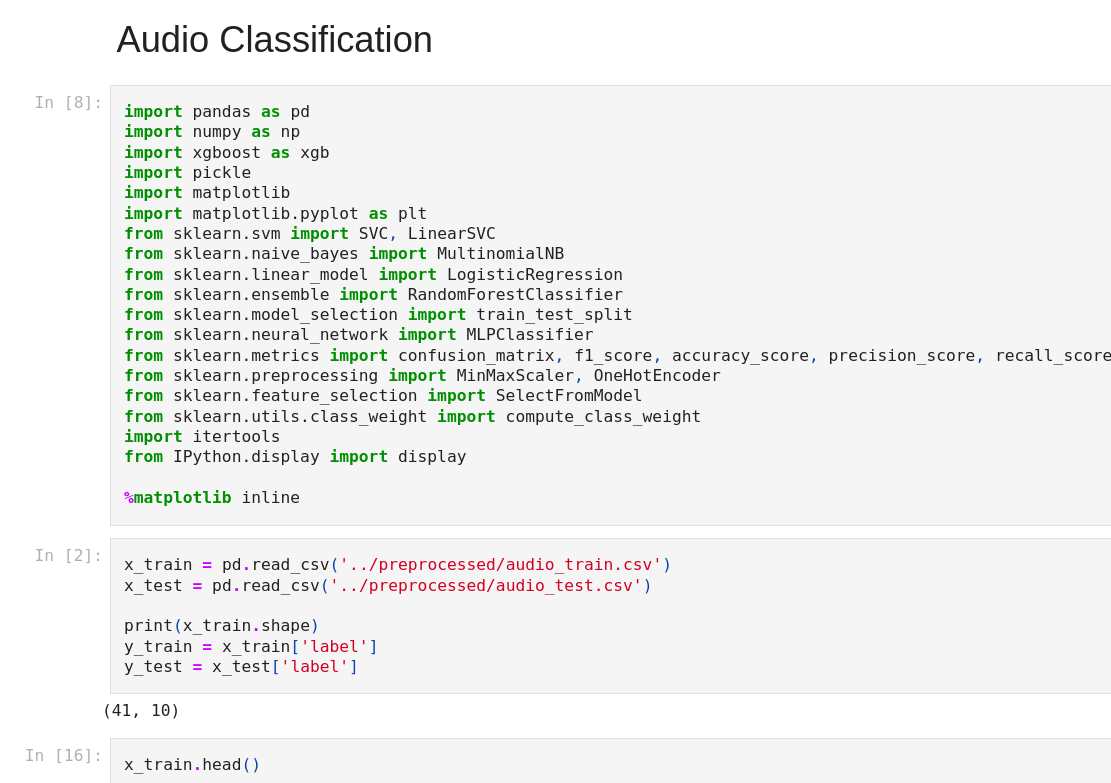
c) Ensemble Methods: Ensemble methods, such as Random Forests and Gradient Boosting, can be employed to combine multiple models' predictions and improve overall performance in emotion recognition.

d) Deep Convolutional Generative Adversarial Networks (DCGANs): DCGANs can be used for generating synthetic facial images that represent different emotions. These synthetic images can be combined with real data to create a more diverse and augmented dataset for emotion recognition.

e) Transformer-based Models: Transformer-based models, like the Vision Transformer (ViT), have shown promise in various computer vision tasks. They can be adapted for emotion recognition by using self-attention mechanisms to process facial image sequences.

f) 3D Convolutional Networks: For video-based emotion recognition, 3D CNNs can be employed to capture spatiotemporal features from video frames, providing a better understanding of emotional dynamics over time.

Each model has its strengths and weaknesses, and the choice of model depends on the specific requirements of the emotion recognition application, the available dataset, and computational resources. Implementing and comparing multiple models can provide valuable insights into their performance and guide future improvements in the emotion recognition system.



**4. Brief Of Method Used:**

1. XGBoost (XGB) Classifier:

XGBoost stands for "Extreme Gradient Boosting," and it is an ensemble learning method that utilizes decision trees as base learners. XGBoost is known for its efficiency, scalability, and high predictive performance. It is widely used in machine learning competitions and real-world applications. The key idea behind XGBoost is to build a strong predictive model by combining the predictions of multiple weak models (decision trees) in an iterative manner, while optimizing a specific objective function.

1. MLP Classifier (Multilayer Perceptron Classifier):

The MLP Classifier is a type of artificial neural network (ANN) used for classification tasks. It is a feedforward neural network, meaning the information flows in one direction, from input to output layers. The network consists of multiple layers, including an input layer, one or more hidden layers with neurons, and an output layer. Each neuron in a layer is connected to all neurons in the subsequent layer with weighted connections. MLP classifiers can learn complex patterns from data and are particularly effective for tasks involving high-dimensional data, such as image and text classification.

1. SVC Classifier (Support Vector Classifier):

The SVC Classifier, also known as Support Vector Machine (SVM) classifier, is a popular supervised learning algorithm used for both classification and regression tasks. SVMs are primarily used for binary classification, but can be extended to handle multi-class classification as well. The main idea behind SVM is to find the hyperplane that best separates the data points belonging to different classes in the feature space. It aims to maximize the margin between the support vectors (data points closest to the decision boundary) of different classes. SVMs are effective in high-dimensional spaces and can handle complex datasets by using kernel functions to transform the data into higher dimensions.

1. MNB Classifier (Multinomial Naive Bayes Classifier):

The MNB Classifier is based on the Naive Bayes algorithm and is commonly used for text classification tasks, such as spam detection and sentiment analysis. It is particularly useful when dealing with large feature spaces, like in text data, and works well even with limited training data. The "naive" assumption in Naive Bayes is that the features are conditionally independent, which simplifies the computations. Despite this simplification, Naive Bayes classifiers often perform surprisingly well in practice.

1. Logistic Regression:

Logistic Regression is a popular linear classification algorithm used for binary classification problems. Despite its name, it is primarily used for classification, not regression. Logistic Regression models the probability that an instance belongs to a particular class using the logistic function (sigmoid function), which maps the output to a probability value between 0 and 1. If the probability is greater than a threshold (usually 0.5), the instance is classified into one class; otherwise, it is classified into the other class. Logistic Regression can handle linearly separable data and is relatively interpretable.

**5. Expected Outcome:**

The expected outcomes of the project include:

Development of an emotion recognition model capable of accurately classifying human emotions based on preprocessed data.

Documentation of the project, including details of data preprocessing, algorithm selection, and model development process.

\*21CE002 WAS NOT INVOLVED IN THE PROJECT, AS HE WAS NOT RESPONDING.

→ Whole idea of implementation and the code was prepared by Neel Shah(22AIML048) only.

→ Harshit Kajakiya(21CE001) and Neel Shah(22AIML048) equivalently contributed in the creation of the report.

Prepared by: 21CE001(Harshit Kajakiya),22AIML048(Neel Shah)

Github Link: https://github.com/NeelDevenShah/emotion-detection/tree/main