Emotion detection with data fusion

By

Maida Khuzhaniyazova

B.A., L.N. Gumilyov Eurasian National University, 2017
M.A., Nazarbayev University, 2022

A Report Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF ENGINEERING

In the Department of Electrical and Computer Engineering

University of Victoria

Supervisory Committee

Dr. Kin Fun Li, Department of Electrical and Computer Engineering

Supervisor

Dr. Daler N. Rakhmatov, Department of Electrical and Computer Engineering

Departmental member

Table of Content

List of Tables

List of Figures

Abstract

This report explores the performance of three machine learning models — SVM with SGDClassifier, Gradient Boosting, and XGBoost — in detecting emotions using data fusion techniques. Early Fusion was chosen for integrating features due to its simplicity and reliable performance. The study employs the MELD dataset, which combines text, audio, and visual data from over 1,300 dialogues and 13,000 utterances in the "Friends" TV show. This dataset provides a unique multimodal approach to understanding emotions in conversational contexts, making it ideal for emotion recognition tasks.

Evaluation metrics for the models included accuracy, F1-score, precision, recall, and AUC-ROC,calculated over multiple training iterations. By comparing the performance of these models on a comprehensive, multimodal dataset, this study meets the growing demand for accurate emotion detection in conversational AI. XGBoost demonstrated high and consistent performance on the MELD dataset; however, its effectiveness may vary under different conditions or datasets. SVM with SGDClassifier achieved the widest accuracy range,though less stable on nuanced emotions. Gradient Boosting delivered consistently strong AUC-ROC values but required full retraining with each data update, affecting its adaptability.

Overall, while XGBoost and SVM delivered good performance, their accuracy was subject to fluctuations across iterations. Gradient Boosting consistently showed strong AUC-ROC values, but its disadvantage is the need to completely retrain the model when new data is added, which reduces efficiency.

Key Words: emotion detection, data fusion, multimodal emotion recognition, machine learning, neural networks, emotion recognition systems, multimodal fusion, XGBoost, Support Vector Machine (SVM), Gradient Boosting, emotion classification, incremental learning, performance metrics

I. Introduction

1.1. Background and motivation

Emotion is a psychological state characterized by subjective experience, physiological response, and behavioral expression [1]. It is divided into seven basic types: anger, disgust, sadness, joy, neutral, surprise, and fear. Emotion detection plays a big role in various fields, including human-computer interaction, mental health, and customer service. Correctly identifying emotions can improve how people and machines communicate, making those interactions feel more understanding and effective. A big challenge in recognizing emotions is figuring out how different emotional states connect to how people express them [2]. Even though computers may not ideally recognize emotions, they can be trained to detect affective signals better than chance, and their continuous monitoring capabilities, especially through wearable devices, offer the potential for enhanced emotional understanding when combined with human interpretation [3].

Emotion detection often focuses on single data sources like text, speech, or facial expressions. While these techniques can be effective, they can sometimes face challenges when capturing the full complexity and nuances of a person's emotions [4]. Emotion detection has come a long way, especially with multimodal approaches, which use different methods like voice, facial expressions, and text together making it possible to create a more complete picture of emotional states. This approach helps to avoid the limitations of using just one type of data because unimodal approaches often have difficulties dealing with the complexity and variety of human emotions [5]. So, the advent of multimodal emotion detection techniques, which integratedata from various sources, has shown promise in improving the accuracy and reliability of emotion recognition systems [6]. In addition, multimodal techniques could be effectively used in fields such as mental health monitoring, where analyzing all emotional signals using different channels can lead to more accurate predictions [7]. Furthermore, there is a good potential for future development due to such technologies as brain-computer interfaces (BCI) and wearable devices that can improve real-time emotion tracking, which could be helpful in healthcare, customer service, and entertainment industries [8].

1.2. Objectives and research questions

The main goal of this study is to research how the use of multimodal data can improve the accuracyof emotion recognition systems. In particular, this work focuses on how the integration of audio, video, and text characteristics can improve the performance of machine learning models. This study also aims to investigate and compare models that have shown effectiveness in otheracademic articles in the field of emotion detection.

The following research questions will guide the investigation:

1. How does the fusion of audio, video, and textual data perform in emotion recognition tasks?

2. Which machine learning model chosen for the study is more effective in handling multimodal data for emotion detection?

Generally, this research aims to identify effective machine learning techniques for emotion

detection, with a focus on replicating or improving upon previously established results. I have selected SVM, Gradient Boosting, and XGBoost models because, in my perspective, these algorithms have demonstrated strong performance in complex classification tasks, especially where nuanced emotion recognition is required.

The dataset used in this study consists of video clips from the TV series "Friends." This dataset was chosen because it captures more lifelike emotional expressions within real conversational contexts, rather than actors explicitly displaying emotions for the camera, which is often the casein other datasets. This realistic portrayal allows the model to generalize better to subtle, naturallyoccurring emotions, making it suitable for building a robust emotion detection system.

II. Methodology

2.1. Literature Review

2.1.1. Search for sources

The first stage brings together important academic articles, books, and materials on the topic of emotion detection published after 2017. This provides up-to-date information on the latest research and effective methods.

Criteria for selecting sources:

- **Relevance:** Sources should be directly related to the topic of emotion detection.
- **Variety:** A variety of methods are expected to be available, the focus will be on research that looks at multiple methods at the same time.
- **Databases:** Platforms such as Google Scholar, IEEE Xplore, and ResearchGate will be used to search for high-quality articles. Attention will also be paid to key journals in the field of artificial intelligence and machine learning.

Exclusion Criteria:

- Non-peer-reviewed articles, irrelevant studies that do not contribute directly to understanding emotion detection, and older publications that do not reflect current trends or technologies are excluded.

2.1.2. Data extraction and synthesis

Once the relevant literature is identified, key information is extracted and organized based on themes, methodologies, applications, and technological approaches. This information is synthesized to highlight:

- The evolution of emotion detection techniques.
- Comparative analyses of unimodal versus multimodal approaches.
- The effectiveness of various machine learning models employed in emotion detection.

A critical analysis is performed to evaluate the strengths and weaknesses of the studies reviewed.This includes assessing the methodologies used, the validity of the findings, and the limitations noted by the authors. The analysis also addresses gaps in the current literature and suggests areasfor future research.

2.2. Application of theory

Based on insights from the literature review and data analysis stages, the coding process was specified for emotion detection with data fusion. The study included an analysis of existing methods proven effectiveness in identifying emotions, particularly testing early fusion and hybrid fusion methods. Ultimately, early fusion was chosen due to its lower complexity and efficiency in integrating multimodal data. To further enhance text processing, models like BERT—developed by Google—were considered, as BERT has significantly advanced natural language understanding through bidirectional context processing, enabling it to capture relationships between words in both directions [9]. However, despite its strengths, BERT was tested but not included in the final model due to its computational complexity and slower performance at this stage. The cross-entropy loss method was also considered due to its limited accuracy of less than 50%.

In analyzing existing Kaggle projects using the Multimodal EmotionLines Dataset (MELD), the project by MD. Hamid Hosen [10] was notable for its consistency and bimodal approach, combining textual and visual data. It employed Convolutional Neural Networks (CNN) for image analysis and Long Short-Term Memory (LSTM) for text processing, achieving an accuracy of 60.34% through decision-level data fusion.

The literature review and project analysis informed the selection of effective approaches to emotion recognition. Various methods were evaluated, with SVM, Gradient Boosting, and XGBoost standing out as particularly effective. The assessment of early fusion methods demonstrated their efficacy and ease of integrating multimodal data, leading to their selection for the model. For feature extraction, TfidfVectorizer was applied to the text data, converting it into fixed-length vectors [21], [22]. Audio files were analyzed using the librosa library, extracting features such as Mel-Frequency Cepstral Coefficients (MFCC), Chroma, and Mel spectrograms [17], [18]. For video analysis, a pre-trained VGG16 model was utilized to obtain high-level features [19], [20].

2.3. Implementation

The implementation should create a complex model for accurate emotion recognition based on multimodal data: audio, video, and text. It is aimed at:

- Data Automation – Simplify data loading, cleaning, and preprocessing processes for standardization and preparation for analysis.

- Extract key features from audio, video, and text — highlight features that help distinguish emotions, such as spectrograms from audio, facial expressions from video, and text vectors.

- Integration of all features into a single system – combining features from different sources for joint analysis and more reliable recognition of emotions.

- Train and optimize models for emotion recognition – Build predictive models to achieve high accuracy based on the highlighted features.

The project design included tools such as Google Colab, Google Drive, and Python, along with their libraries, which are used as the necessary tools for processing audio, video, and text to create an effective development environment.

Google Colab: This platform was chosen because it provides a cloud-based environment that allows Python code to be run without setting up any local infrastructure. Moreover, it supports GPU acceleration, which is useful for running computationally difficult code like processing video and audio data. Furthermore, Google Colab is easily shared and presented to receive feedback.

Google Drive: This tool stores and manages data. It allows easy access to datasets from any device,which is especially important for large datasets, as it helps avoid local storage limitations. It is alsoimportant to note that Google Drive efficiently collaborates with Google Colab.

Python and its libraries: Python is a popular programming language with a wide range of librariesthat fit data processing tasks. For instance, Libraries like NumPy and Pandas will be used for datamanipulation. NumPy is great for working with multi-dimensional arrays and performing complexmathematical operations efficiently, and Pandas provides powerful data structures and functions that make it easy to analyze and preprocess tabular data. For video processing, OpenCV and MoviePy were chosen. OpenCV is a library for image and video processing, which is effective fortasks like frame extraction, object detection, and image transformations. MoviePy is handy for editing and processing video files, allowing the extraction of audio tracks and creating compositions from multiple clips. Next, Librosa provides a variety of tools for audio analysis, including feature extraction methods like MFCC, etc. Finally, machine learning libraries TensorFlow and PyTorch are used to build and train emotion detection models. TensorFlow is a Python framework for deep learning models, while PyTorch is for building computational graphs.

The audio, video, and text data should be cleaned and preprocessed to make the dataset ready for analysis. This includes normalizing formats, taking into account missing values, and normalizingfeatures to ensure data consistency.

For feature extraction, there are applied specific techniques adapted to each type of data modality:

- Mel-frequency Cepstral Coefficients and, Mel-spectrogram and Chroma Features will be extracted for audio.

- A pre-trained deep learning model will extract facial expression features for video.

- Term Frequency-Inverse Document Frequency (TF-IDF), which weighs terms based on

their frequency in a document and rarity across documents is applied to vectorization for text.

Model development: Machine learning models are developed using the extracted features, experimenting with various algorithms (SVM, Gradient Boosting, and XBoost), and fine-tuning hyperparameters to achieve optimal performance.

Integration and testing: The models will be integrated into a coherent system that combines audio, video, and text data for emotion detection. To ensure that the model works correctly, tests must be run to check how it performs and stays stable over time. For documentation and reporting, the code and methods used throughout the project should be saved to reproduce the results later and clearly share the insights. The trained models play a crucial role in achieving highest possible accuracy and efficiency in emotion recognition tasks.

III. Literature Review

3.1. Introduction to emotion detection

Emotion detection, frequently related to affective computing, recognizes and categorizes human emotions — such as happiness, sadness, anger, fear, and surprise through sophisticated computational techniques. This area has developed extensively in recent years because of its wide-ranging applications, which include human-computer interaction (HCI), healthcare, education, and market research. Since emotions play a vital role in communication, accurately identifying and interpreting emotional states, is essential for enhancing user experiences and achieving better results across various fields [11]. The primary goal of emotion detection systems is to enable machines to recognize and respond to human emotions in real-time, similar to how humans do in social interactions. To predict emotional states, these systems analyze various modalities—such as facial expressions, vocal tone, and textual content — individually or in combination. Advances in machine learning and deep learning have significantly enhanced the accuracy and performance of emotion detection models, allowing for the automatic extraction of relevant features from large-scale datasets [12].

An analysis of existing studies shows that most of them use data sets that do not reflect the real diversity of emotions and contexts. For example, many works are based on datasets where actors show emotions very clearly, which makes them unreliable enough for use in more complex and dynamic scenarios, such as video recordings of real-world interactions. This project, using a more complex MELD dataset, aims to fill this gap by offering a more comprehensive understanding ofemotions in more realistic environments. Moreover, many studies use standard classification methods, such as logistic regression or simple decision trees. However, given the complexity of the emotion recognition task, more advanced algorithms like SVM and XGBoost can provide significant benefits. My project aims to explore these possibilities and demonstrate their effectiveness in the context of emotions.

3.2. Applications and importance of emotion detection

With the rapid advancements in mobile Internet and artificial intelligence technologies, human-machine communication has become increasingly prevalent. There is a growing demand for AI-based systems to recognize user emotions and provide appropriate feedback. Emotional

interaction, an essential aspect of human-computer interaction, enhances machines' ability to understand and respond to human emotions, making the interaction more natural and empathetic. Current service robots often rely on keyword matching and lack true emotional understanding, which limits their capacity for meaningful communication. Incorporating affective computing technology has the potential to enable machines to recognize emotional states and provide responses that align more closely with human emotional expressions. For example, systems like emotional escort such as Japan's Pepper robot analyze facial expressions to determine user emotions [13]. At the same time, MIT has been developing emotional computing frameworks that can be used with various data types, like facialexpressions and physiological signals, to reach similar goals. These innovations show how much potential emotion detection technologies have to change how humans interact with machines, making those interactions more engaging, empathetic, and effective in meeting what users need [13].

Additionally, there has been a growing focus on using emotion detection in clinical settings, particularly when it comes to neurological and psychiatric disorders. Recent research has shown that emotion recognition systems can play a crucial role in diagnosing and monitoring conditions such as Autism Spectrum Disorder (ASD) is characterized by emotional behavior disturbances in this context emotion recognition tools are being developed to create emotionally intelligent training systems, often integrated into video games and augmented reality. These tools aim to enhance emotion recognition skills and help reduce emotional disabilities. They have proven helpful in applying therapy gains to everyday situations, especially given the shortage of therapists available. In the case of Parkinson's Disease (PD), emotion detection is important for keeping an eye on symptoms such as reduced facial expressions (hypomimia) and challenges in recognizing emotions in others (alexithymia). This helps clinicians gain a better understanding of their patients' psychosocial functioning. Moreover, emotion detection systems are also being investigated for diagnosing and tracking psychiatric conditions such as bipolar disorder, schizophrenia, and depression. Accurately identifying emotional states can support early diagnosis and treatment, which might help prevent serious episodes and enhance patient outcomes [14].

In the educational field, there is a growing use of AI methods, such as machine learning and facialrecognition, to assess students' emotional states during their education process. This approach aimsto foster adaptive and personalized educational environments. There are exciting emerging areas like federated learning, convolutional neural networks (CNNs), recurrent neural networks (RNNs),and ethical considerations in AI development that need further exploration to unlock their full potential in enhancing educational outcomes [15].

In addition, there is a growing trend in the automotive industry for emotion recognition systems that use data fusion techniques. These systems are designed to monitor how drivers are feeling—specifically, their levels of stress and anger—by examining multiple types of data, including heartrate, skin response, and vocal patterns. The goal is to improve road safety and driver well-being by allowing vehicles to adapt to the driver’s emotions in real-time, potentially reducing accidentsand enhancing overall mental health [16].

Moreover, various platforms incorporate emotion recognition technologies into virtual assistants to enhance user interactions. For instance, a home virtual assistant can sense emotions in a user's voice and offer activities or media to improve their mood, like relaxing music or a funny podcast.Using emotion recognition technology enables virtual assistants to respond in a more personalizedand understanding way, finally improving the user experience and, as a result, greater engagement [17].

3.3. Modalities in emotion detection

Emotion detection systems generally rely on various modalities to identify and classify emotions.This project includes audio, video, and text to facilitate the analysis of emotions in conversational settings, making it an important resource for multimodal emotion recognition.

3.3.1. Audio-based emotion detection

Audio-based emotion detection analyzes vocal characteristics, including tone, pitch, and rhythm, to recognize and classify emotional states. This approach takes advantage of the natural fluctuations in human speech, which can bring a range of emotions such as happiness, sadness, anger, fear, surprise, etc. Significant techniques in audio emotion detection encompass feature extraction methods, including MFCC and modulation spectral features (MSFs) [18].

- **MFCC:** Widely used in speech and audio processing, MFCCs effectively capture the power spectrum of audio signals. This method creates a representation that is important for differentiating phonetic characteristics and emotional expressions [19].

- **MEL Spectrogram:** This technology transforms audio signals into a time-frequency representation, closely matching human feelings of pitch. It helps identify patterns linked to emotional signals based on the frequency of the content [19].

- **Chroma Features**: These features represent the harmonic content, helping to discern tonal elements within the signal that may indicate emotional states [19].

These characteristics reflect the acoustic nature of speech, which machine learning models like Support Vector Machines (SVM), neural networks, etc. use to forecast emotions. Audio-based systems show great potential for improving human-computer interaction, healthcare, and customer service, as recognizing vocal emotions can lead to more tailored and empathetic responses. Often, this modality is combined with other data sources, including facial expressions and text, in multimodal systems to enhance accuracy and reliability [18].

3.3.2. Video-based emotion detection

In video-based feature extraction using Visual Geometry Group 16 (VGG-16) developed at the University of Oxford, the first step is to pull high-quality frames from the video, followed by preprocessing to standardize their size and improve image quality. After this, the VGG-16 model, which has been pre-trained, examines these frames. Its convolutional layers capture detailed patterns, while the fully connected layers help with classification. Research has shown that using pre-trained weights instead of starting from scratch significantly improves efficiency and accuracy, making this method highly effective for emotion detection. This approach has proven robust even under varying conditions, such as lighting situations and motion distortions [20].

Deep learning methods, particularly Convolutional Neural Networks (CNNs), have demonstratedimpressive performance in recognizing the nuances of human facial expressions. Fine-tuning pre-trained models like VGG-16 for specific tasks has been incredibly beneficial, especially when working with limited data. This method improves accuracy and opens new real-world applications,highlighting emotion recognition as a rapidly growing and impactful area of

research [21].

3.3.3. Text-based emotion detection

Written language plays an important role in people's communication and helps them understand emotions better. With social media popularity increasing, there appeared to be access to a huge amount of text data for analysis. It is important to detect emotions and sentiments in this text to understand what people mean. Emotions can be sorted into groups like joy, anger, fear, etc., orlooked at in terms of sentiment, like positive or negative. Before the text can be analyzed, it is required to be changed into a numerical format. This step, called text vectorization, turns the language into numerical vectors, making it easier to analyze. Early methods include One-Hot Encoding, which represents each word as a unique binary vector and TF-IDF. It captures term relevance but does not account for semantic relationships [22].

TF-IDF is a method used to identify significant terms in a document by combining two key components: term frequency and inverse document frequency. This approach emphasizes terms that are frequently used within a specific document while also identifying those that are infrequent across a broader collection of documents. By applying TF-IDF, the text is converted into numerical feature vectors, creating a unique representation for each document based on the importance of its terms. One of the main advantages of this technology is its computational simplicity [23].

The TF-IDF formula is:

$$W(d,t) = TF(d,t) \times \log\left(\frac{N}{df(t)}\right),$$

where

$d -$ denotes documents,

$t -$ term,

$TF(d,t) -$ term frequency in the document,

$N -$ total number of documents

$df(t) -$ document frequency in the corpus of documents [23].

3.3.4. The shift to multimodal emotion detection

While the unimodal approach has helped recognize emotions, researchers have noted its limits. For instance, the person could sound happy based on their voice, but their facial expression couldindicate that they're not really enjoying. To handle such cases, the focus has shifted to multimodalemotion detection. This approach combines information from different sources—like audio, video,and text—to improve how accurately emotions can be classified [24].

Exploitation of multiple data sources is a big advantage because emotions are complex and hard to capture with just one type of input. Combining different types of data is essential for getting it right. Various data fusion strategies are implemented to combine these modalities effectively to

enhance recognition performance [25].

Early Fusion is one of the common strategies where feature vectors from different modalities (like video, audio, and EEG) are concatenated into a single input vector, which is then fed into a deep learning (DL) model. This approach integrates features before classification, allowing the model to process all modalities simultaneously. While early fusion can effectively capture a broad rangeof emotional cues, it does have its challenges. For instance, it can struggle with features specific to certain modalities and might not perform as well when different modalities have irrelevant or redundant information [25].

However, early fusion is still an efficient model, which key benefits include:

- **Combining data:** Combining features from different sources early helps to better reflect emotions, as the model can use different information simultaneously [25].

- **Use of relationships:** When data from multiple sources is combined before machine learning techniques are applied, the relationships between them are better considered, which improves accuracy compared to using a single data source [25].

- **Simplicity:** The early union method is quite simple. Features from different sources are combined into one large data vector, making it easier to implement than more complicated fusion techniques [25].

- **Efficiency in Processing:** Early fusion allows for directly applying deep learning modelsto the fused feature set, potentially optimizing the training and classification processes [25].

Hybrid fusion is another method that combines early fusion and decision fusion (also known as late fusion). In this model, features from each modality are processed separately to extract modality-specific information. These features are then combined later, where decision rules, such as weighted averages, are applied to obtain a final emotion prediction. This method of combining allows us to consider each modality's strengths while combining them for a more accurate recognitionof emotions, making it more flexible compared to early unification [25].

Multitasking Learning (MTL) is a modern strategy for recognizing emotions based on multiplemodalities, such as audio and video. MTL improves learning efficiency by capturing correlationsbetween tasks while reducing the risk of overfitting. This collaborative work with different modalities allows the model to identify common emotional patterns while preserving the unique characteristics of each source. MTL is generally superior to other methods because it can better generalize and manage noise [25]. On this basis, the MTL-BAM (Multitask Learning-Based Attention Mechanism) method solves important problems of multimodal emotion recognition, paying attention to interactions between modalities and their differences, which are often overlooked. MTL-BAM integrates audio, video, and text more effectively, improving emotional analysis by increasing attention to the contribution of each modality. This leads to a more accurate and dynamic integration of emotional information. Unlike traditional early unification, which has difficulty distinguishing between modalities, MTL-BAM successfully combines multitasking learning with attentional mechanisms. This approach helps bring out the shared emotional traits and the unique input from each modality, which plays a key role in accurately recognizing emotions. As pre-training techniques for text, audio, and video continue to improve, MTL-BAM offers a solid groundwork for future developments in combining multiple modalities for emotion

detection [26].

3.4. Models: SVM, Gradient Boosting, and SVM

Selecting the appropriate model is essential for obtaining precise and significant results in data analysis and machine learning. Using models that can handle high-dimensional inputs and capturethe nuanced emotions in multimodal data is crucial as data gets more complex, particularly in areas like emotion detection. In a variety of projects including high-dimensional datasets, Support Vector Machines (SVMs), Gradient Boosting, and XGBoost have demonstrated effectiveness in identifying complex correlations during emotion recognition tasks. According to research, SVMs are excellent at finding suitable hyperparameters, but Gradient Boosting and XGBoost enhance prediction performance by using incremental learning.

Alternatives such as BERT were considered but excluded due to their computational complexity and slower processing speeds. The Cross-Entropy Loss method was also explored but deemed unsuitable due to its low accuracy. Additionally, techniques like Random Forest and k-Nearest Neighbors (k-NN) were evaluated but not chosen for their limitations in modeling complex relationships in multimodal data.

3.4.1. Support Vector Machines

Support Vector Machines (SVMs) have proven to be a powerful tool for classification in a varietyof fields, including the complex task of emotion detection across multiple data sources like video,audio, and text. The strength of SVM lies in its approach of finding a hyperplane that better separates different classes in the feature space, maximizing the margin between distinct class points. This property is especially advantageous in managing high-dimensional datasets, which are often encountered in emotion recognition. When applied to emotion detection, SVMs can use both linearand non-linear kernel functions to reshape the input space, enabling the model to capture the complex patterns within the data. For example, employing an RBF (Radial Basis Function) kernelhelps the SVM handle intricate relationships by recognizing patterns that emerge across audio, visual, and textual features combined. Additionally, SVMs utilize slack variables, which allow themodel to account for data points that do not fall neatly within class boundaries. This feature is essential in real-world contexts, where noise and overlapping class data are common. As a result,SVMs offer both adaptability and improved generalization, making them well-suited for the sophisticated demands of emotion recognition tasks [27].

To address the challenges of real-time emotion recognition, researchers are increasingly utilizingincremental learning methods, as demonstrated by Anowar and Sadaoui in their work on self-labeling of incoming high-dimensional data. They initially built a classifier using the Stochastic Gradient Descent (SGD) algorithm on 225 features with 10-fold cross-validation, achieving a highF1-score of 92% and a False Negative Rate (FNR) of 8% in less than 2 seconds. Incremental learning is then applied to update the model with new data, which is first reduced and self-labeled.This allows the classifier to improve over time while maintaining computational efficiency, as updates take minimal time (0.37 seconds for the largest chunk). Moreover, the model refines its accuracy with each new data chunk, and a comparison with human-labeled data shows that pseudo-labels, while slightly less accurate, remain competitive with minor differences in F1-scores [28].

3.4.2. Gradient Boosting

Gradient Boosting is a learning technique that constructs a model by consistently adding weak learners, typically decision trees, to improve predictive performance. The basic principle of gradient boosting is to improve the accuracy of a model by adjusting new models to match the mistakes of previous ones. Due to this iterative process, the model becomes more accurate, which makes gradient boosting especially effective for complex tasks such as emotion recognition using multimodal data. In emotion recognition, gradient-boosting algorithms can adapt to different data types by using the strengths of both categorical and continuous features present in audio, video, and text data. Techniques such as adaptive boosting (AdaBoost) and gradient decision trees (GBDT) are often used to solve such problems. GBDT, in particular, has gained traction due to its robustness against overfitting and its ability to model intricate patterns in high-dimensional data. One of the significant advantages of gradient boosting is its flexibility in handling different loss functions, allowing it to be tailored for specific applications, such as classification or regression. This ability to adapt plays an important role in the recognition of emotions, as the classification can change significantly depending on the context and the modalities used. In addition, gradient boosting effectively deals with missing values and interactions between features, which expands its application in real datasets [29].

Techniques for gradient boosting, including LightGBM and CatBoost, have become increasinglypopular. These methods enable quicker learning and optimize memory usage without decreasing accuracy. They perform well when handling large datasets and frequently overtake traditional gradient-boosting approaches. In emotion recognition tasks, such improvements can speed up themodel training process and improve the efficiency of real-time data processing, making it suitable for dynamic conditions. In addition, research in emotion recognition is considering integrating gradient-boosting techniques with incremental learning strategies. This allows models to learn adaptively from new data, improving the system's ability to respond to new emotional expressionsand changes in user interactions without the need for complete retraining. This capability is particularly beneficial in applications such as sentiment analysis in social media, where emotionalcontexts continually evolve [29].

3.4.3. eXtreme Gradient Boosting

XGBoost (eXtreme Gradient Boosting) is a machine learning model based on decision trees, designed to handle large datasets and improve accuracy using gradient boosting techniques. In the project written by Chen and Guestrin, they employed incremental learning for XGBoost, allowing the model to be trained on new data while keeping knowledge from previous runs. The model checks for changes in the number of features before deciding whether to retrain from scratch or continue training with the existing model. This approach allows the model to remain flexible and adapt to new conditions without requiring complete retraining if the set of features remains the same [30].

Incremental learning, in which the model is updated with new data instead of completely retraining, is especially useful when data constantly changes. XGBoost, a popular gradient-boosting framework, supports incremental learning, which allows to add new evaluation models while preserving previous training. However, incremental training requires that the feature set remains the same throughout the upgrade phases. If the new data contains other features, XGBoost will throw an error indicating a mismatch in the dimensionality of the features. This feature requires a careful approach to managing the set of characteristics, especially in projects where they

change frequently. Also, with each new boosting round, the model processing time increases, creating problems in applications where quick response is important. Therefore, there is a need to find a trade-off between the complexity of the model and its performance. Despite these nuances, incremental learning in XGBoost is a powerful tool for adaptive modeling in changing data [31].

IV. Dataset overview: MELD - Multimodal EmotionLines Dataset

The choice of the dataset is another crucial step and besides this MELD dataset several others wereconsidered:

- **EmoReact:** A dataset focused on emotional reactions in video content but lacks the context of dialogues, which MELD provides [32].

- **CMU-Multimodal Opinion Sentiment and Emotion Intensity (CMU-MOSEI):** Contains multimodal data for sentiment analysis but is less detailed in emotion categorization and dialogue context [33].

The choice of Multimodal EmotionLines Dataset (MELD) is based on its combination of textual,audio, and visual data from the popular series “Friends”, allowing a detailed analysis of emotionsin the context of conversations. Additionally, there are only a few completed projects using this dataset on Kaggle, and they did not utilize the methods and approaches applied in this research. Specifically, the MELD is an upgraded version of the EmotionLines dataset designed to enable multimodal emotion recognition in informal contexts. It contains text, audio, and visual data fromover 1,300 dialogues and 13,000 utterances sourced from the widely popular TV series “Friends”.Each utterance is labeled with one of seven emotions: anger, disgust, sadness, joy, neutral, surprise,and fear. Sentiment labels are categorized into three categories: positive, negative, or neutral.

4.1. Key features

- **Multiple Modalities:** MELD includes text, audio, and video for each utterance, assisting multimodal emotion recognition.

- **Emotion Annotations:** Each utterance is labeled with an emotion, enabling detailed emotion analysis across different media.

- **Sentiment Analysis:** In addition to emotions, each utterance is also annotated with sentiment labels (positive, negative, neutral).

- **Contextual Understanding:** As conversations are organized into dialogues with multiple speakers, the dataset supports the development of models for contextual emotion recognition, capturing the flow of emotions between dialogues.

4.2. Dataset details

The MELD dataset was developed by first aligning each line of dialogue from the TV series "Friends" with its corresponding season, episode, and time using subtitles. After that, researchers extracted the audio and video clips for each line. Any errors in the dialogue sequence were

corrected. Finally, a careful review was conducted to ensure the responses were properly ordered and matched within the same scene, addressing mistakes from the original EmotionLines dataset [34], [35].

Dataset Files:

- **train_sent_emo.csv:** Contains training utterances with both sentiment and emotion labels.
- **dev_sent_emo.csv:** Includes development set utterances with sentiment and emotion labels.
- **test_sent_emo.csv:** Provides test set utterances with sentiment and emotion labels.

Each utterance in these CSV files is linked with metadata, including the speaker, dialogue ID, season and episode numbers, and timestamps marking the start and end of the utterance in the episode. This organization allows researchers to use the dataset to develop multimodal models capable of detecting emotions from various input modalities [34], [35].

The sample dialog shown in Figure 1 includes frames from the dataset and text, demonstrating that the MELD dataset reflects natural conversational interactions. Due to the presence of multimodaldata (text, audio, and video), it is as close as possible to real dialogues, which makes it a valuabletool for testing methods of recognizing emotions in more practical and realistic conditions. This context helps improve the relevance of emotion recognition models for everyday situations.

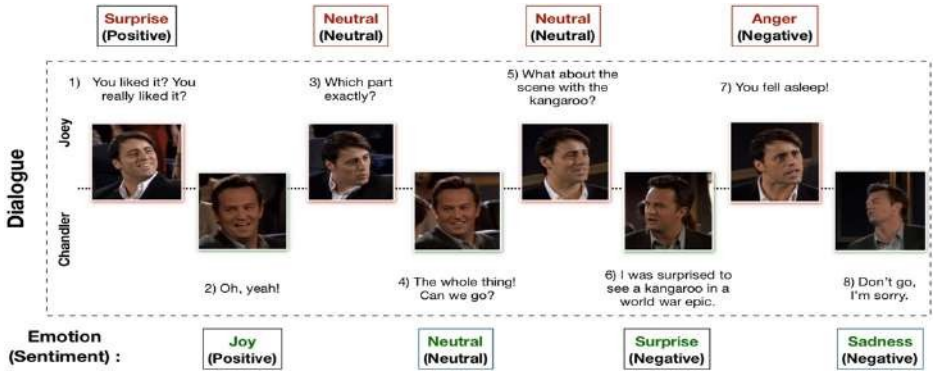


Figure 1.Emotion shift of speakers in a dialogue.
Adapted from Poria, S., Hazarika, D., Majumder, N., Naik, G., Mihalcea, R., & Cambria, E. (2018). MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversation. https://doi.org/10.48550/arXiv.1810.02508

4.3. Dataset statistics

The MELD dataset consists of two main folders: 'MELD-Features-Models,' which includes text, audio, and visual components utilized with various machine learning models (though not used in this project), and 'MELD-RAW,' containing the raw data along with annotations organized into three directories: 'dev,' 'test,' and 'train.' The dataset statistics, which are presented in Table 1, provide a summary of dialogues, lines, emotions, sentiments, and other characteristics in the MELD dataset. The table shows how the data is divided into training, validation, and test sets, including the number of dialogs and replicas.

The training dataset utilized in this study includes a CSV file with a total of 9,989 entries, systematically structured across 11 columns. Each video in the dataset represents a spoken line from a specific dialogue in a TV show, offering useful information on emotional expression and sentiment. The CSV file includes the following columns:

1. **Sr No.**: A unique identifier for each entry
2. **Utterance**: The spoken text in the dialogue, capturing the emotional content
3. **Speaker**: The character delivering the utterance.
4. **Emotion**: The identified emotion associated with the utterance (e.g., neutral, surprise)
5. **Sentiment**: The sentiment classification of the utterance (e.g., neutral, positive)
6. **Dialogue_ID**: An identifier linking the utterance to its corresponding dialogue
7. **Utterance_ID**: A unique identifier for each utterance within the dialogue
8. **Season**: The season number of the series in which the dialogue appears
9. **Episode**: The episode number within the season
10. **StartTime**: The start time of the utterance in the video
11. **EndTime**: The end time of the utterance in the video

Table 1. Dataset statistics

Statistics	Train	Dev	Test
# of modality	{a,v,t}	{a,v,t}	{a,v,t}
# of unique words	10,643	2,384	4,361
Avg. utterance length	8.03	7.99	8.28
Max. utterance length	69	37	45
Avg. # of emotions per dialogue	3.30	3.35	3.24
# of dialogues	1039	114	280
# of utterances	9989	1109	2610
# of speakers	260	47	100
# of emotion shift	4003	427	1003
Avg. duration of an utterance	3.59s	3.59s	3.58s

{a,v,t} = {audio, visual, text}. Adapted from Poria, S., Hazarika, D., Majumder, N., Naik, G., Mihalcea, R., & Cambria, E. (2018). MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversation. https://doi.org/10.48550/arXiv.1810.02508

In my work, only videos from the training directory are used for both training and testing the modelbecause this directory contains a sufficient amount of data for effective model training and

evaluation. This approach helps maintain data consistency and minimizes potential issues relate to variations in video quality or encoding that might arise from using multiple directories. The dataset is complete, with no missing values in its columns. Most utterances are marked as "neutral," while the other emotions are less common. This suggests that more in-depth analysis and stronger models are needed to capture the variety of emotional expressions in the dialogues.

4.4. Visualization of emotion distribution

A frequency plot was created for each emotion category, which can be seen in Figure 2. This chartshows which emotions appear most often and helps in understanding the dataset better for furtheranalysis.

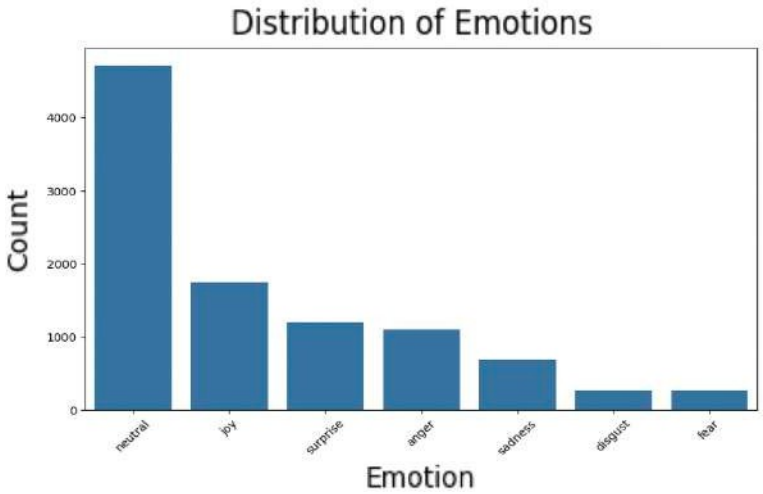


Figure 2.Distribution of Emotions

Neutral emotions refer to expressions that do not express strong positive or negative feelings, oftenrepresenting everyday conversations that lack emotional intensity. The predominance of neutral emotions indicates that:

- **Realistic conversation patterns:** Many interactions in everyday life are neutral in tone, so the dataset likely reflects realistic dialogue.
- **Challenges in classification**: A high number of neutral emotions can make the classification task for machine learning models more difficult, as the model must learn to distinguish nuances of emotions in the context of neutral statements.
- **Need for deeper analysis**: There is a need to explore not only the emotional tone but also thecontext in which neutral statements are made. This may include analyzing intonation, pauses, gestures, and other non-verbal cues that can help in emotion recognition.
- **Class balance**: The predominance of neutral emotions may suggest the necessity of employing class balancing methods to improve model accuracy in recognizing less represented emotions.
- **Character dynamics**: Neutral emotions may also reflect the characteristics of the

characters and their interactions, which can help in creating more complex models that consider the dynamics of relationships between characters.

In summary, the significant occurrence of neutral emotions highlights the importance of a comprehensive approach to understanding emotional dynamics in dialogues.

4.5. Data distribution analysis

4.5.1. Audio data

The emotion recognition process begins with collecting and preprocessing multimodal data, including audio, video, and textual inputs. Integrating these modalities is crucial, as each contributes unique aspects to understanding the emotional state.

MEL, MFCC, and Chroma are common audio features subgroups that can be used to analyze audiodata. The MEL spectrogram is a representation of audio that approximates human hearing by mapping frequencies to the MEL scale, which is more perceptually relevant than a linear scale. Actually, it is a subgroup of 128 frequency bands, which are typically computed from the Short-Time Fourier Transform (STFT) of an audio signal. Each band captures a range of frequencies, allowing for a better understanding of the spectral characteristics of the audio. MFCCs are coefficients that represent the short-term power spectrum of sound, widely used in speech and audio processing. They are derived from the MEL spectrogram and are designed to mimic the way humans perceive sound. There are 13 MFCC features computed for each audio segment. Chroma features represent the energy distribution across the 12 different pitch classes (notes) in music [19].

PCA (Principal Component Analysis) is a technique used to reduce the dimensionality of data thatallows to transform the original variables into a new coordinate system, where the new variables are called principal components. These components are ordered in descending order of importance,meaning the first component explains most of the variation in the data. The principal componentsare selected based on their eigenvalues, which are computed from the covariance matrix. The eigenvalues indicate how much variance in the data is explained by each major component.

The selection process includes the following steps:

- **Sorting Eigenvalues:** Eigenvalues are sorted in descending order. The highest eigenvalue corresponds to the component that explains the largest part of the variation in the data.

- **Selecting the Number of Components:** Determines the number of principal components that you want to keep. Usually, the first ones are chosen *k* component, where *k* is the number of components that explains the sufficient proportion of the total variance.

- **Preservation of Principal Components:** First *k* major components (PC1 and PC2) are stored for further analysis. These components are linear combinations of the original variables and provide maximum variation in the data, which minimizes information loss.

- **Projection on selected components:** After selecting the main components, the data is projected onto these components, which allows you to create a new feature set containing the most significant information extracted from the source data. When PCA is performed,

these features are combined and projected onto new components such as PC1 and PC2 that capture the most variance in the data set [35].

The PCA scatter plot for audio features in Figure 3 shows the data projected onto two principal components (PC1 and PC2), capturing the most significant variance directions within the audio features. Different colors represent distinct emotion labels, with values ranging from 0 to 6 toindicate emotions including surprise, joy, neutral, sadness, anger, fear, and disgust.

Key observations from this plot include:

- The data points are scattered across the two principal components, reflecting diverse features among samples.

- Despite color coding for different emotions, a significant overlap between clusters of varying emotions is evident, indicating that audio features alone do not form clearly separated groups for each emotion.

- This visualization suggests that audio features, even when reduced to two dimensions, do not provide a strong basis for distinguishing emotional categories. Therefore, relying only on audio data for emotion detection may limit classification accuracy. Using additional modalities, such as visual or textual data, may be essential for improving emotion recognition performance.

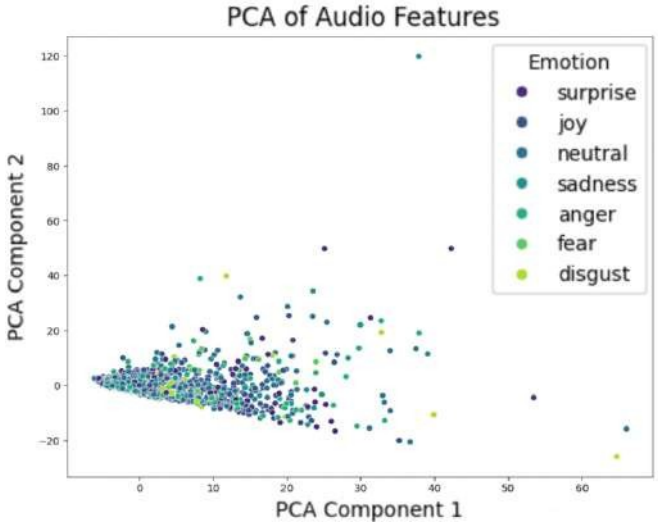


Figure 3.Principal Component Analysis (PCA) scatter plot for the audio features

The second graph (Fig. 4) is a scatter plot showing the distribution of values for each Mel-spectrogram feature across the dataset.

Key insights include:

- The values for most Mel-spectrogram features tend to be concentrated in a lower range, as shown by the clustering of boxplots near the bottom. This indicates that, for most samples, the Mel-spectrogram features do not vary significantly across different frequencies.

- A few features show more extreme outliers, indicating that some samples have higher variance in certain Mel-frequency components. These outliers indicate that some audio

segments have unique characteristics or higher intensity in particular frequency bands.

- This distribution suggests that most audio content has relatively low Mel-spectrogram values, but some outliers skew the distribution upward. These outliers might represent unique emotional expressions or variations in intensity, which could be useful in identifying specific emotions. However, the overall low variance across most features suggests that using Mel-spectrogram features alone may not get enough information to distinguish between nuanced emotions.

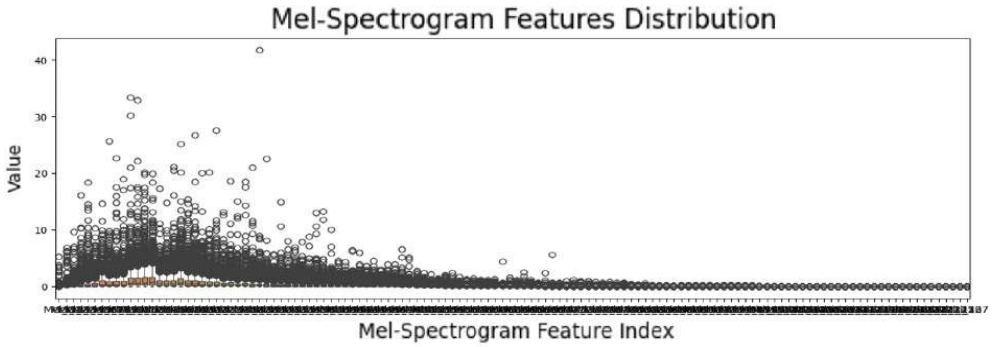


Figure 4. Distribution of values for each Mel-spectrogram feature

The third graph (Fig. 5) is a boxplot representation of the Chroma features, which represent pitchclass profiles.

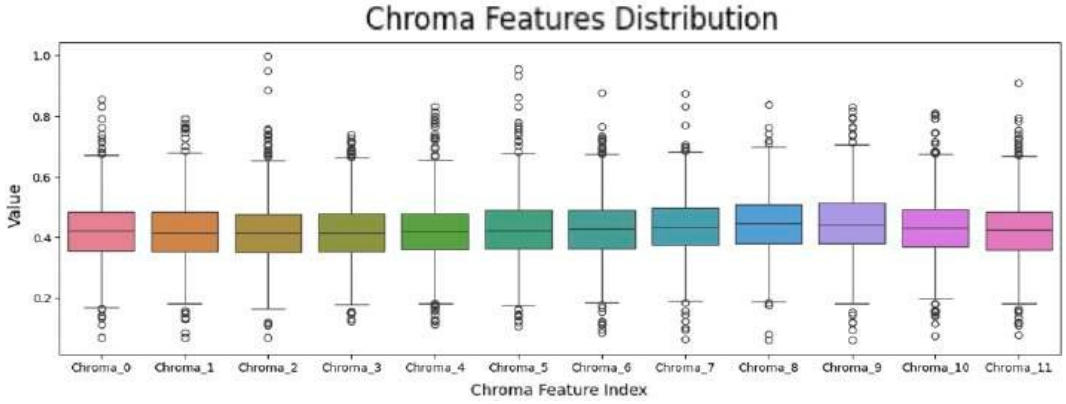


Figure 5. Chroma features distribution.

Notable observations:

- All Chroma features show a reasonably balanced distribution, with median values between 0.3 and 0.6. This balance suggests that information is generally consistent across samples, without extreme deviations in most cases.

- Some outliers are present, indicating a few samples with higher or lower Chroma values than the majority. These outliers might correspond to unique audio characteristics, such as shifts in pitch or more intense harmonic elements, which could be associated with specific emotions.

- The Chroma features display a fairly consistent pattern across different indexes, suggesting uniform pitch-related information in the audio data. It might indicate that pitch alone may not be sufficient to differentiate between emotions, as it does not vary significantly across samples.

The boxplot (Fig. 6) illustrates the distribution of 13 Mel-frequency cepstral coefficients (MFCC)extracted from an audio dataset. Each boxplot represents the statistical distribution for a specific MFCC feature index, ranging from MFCC_0 to MFCC_12. The following key observations can be made:

- MFCC_0 shows a significantly wider range of values compared to other coefficients, witha median around -400 and several outliers below -600. This indicates a large variance in this particular feature.

- The distributions of MFCC_1 through MFCC_12 are comparatively narrower, suggesting less variation within these features.

- The central tendency for most of the other MFCCs remains close to 0, with a few outliers observed in MFCC_1, MFCC_2, and MFCC_3.

- The interquartile ranges (IQR) for MFCC_4 to MFCC_12 are small, indicating that most of the values for these features are concentrated near the median.

The variations between different MFCC features highlight the importance of understanding the individual contribution of each coefficient during the analysis of audio signals, such as in speech or music classification tasks.

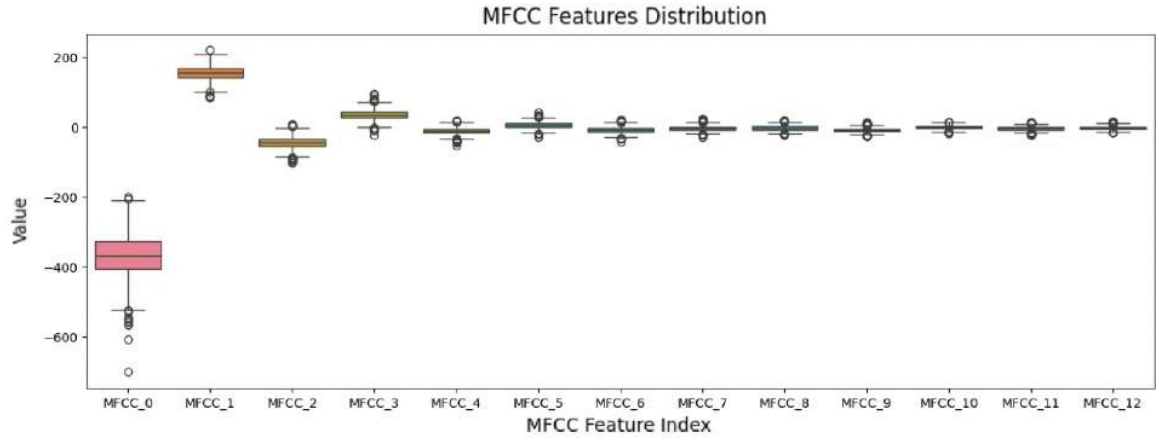


Figure 6.MFCC Features Distribution

4.5.2. Video data

UMAP (Uniform Manifold Approximation and Projection) is one of the modern methods of dimensionality reduction and data visualization, widely applied in various fields such as image processing. The main purpose of UMAP is to project high-dimensional data into low-dimensional space while maintaining the topological structure of the data and in my case, it is visualization of complex data in 2D [36]. The selection process of features for utilizing UMAP includes the

following steps:

- **Data preparation:** The algorithm selects high-dimensional data that represents the visual characteristics of the video.
- **Parameter Setting:** Determines the algorithm parameters, such as the number of neighbors, the minimum distance, and the dimension to which the data needs to be converted.
- **Graph Construction:** UMAP creates a graph that displays the relationships between data based on their distances from each other.
- **Optimization:** The algorithm finds the optimal distribution of points in the new, lower dimensionality, preserving as much of the original data structure as possible.
- **Output:** The result is a set of points in a two-dimensional space, where similar videos are close to each other and different videos are farther away [36].

The UMAP scatter plot for video features in Fig. 7 shows the data projected to two dimensions, capturing the main patterns and relationships of the video features. Different colors represent emotion labels, with values indicating emotions such as surprise, joy, neutral, sadness, anger, fear,and disgust. This visualization allows us to observe how video features cluster based on emotion,providing insight into the ability to separate and distinctive characteristics of each emotion within the low-dimensional space created by UMAP.

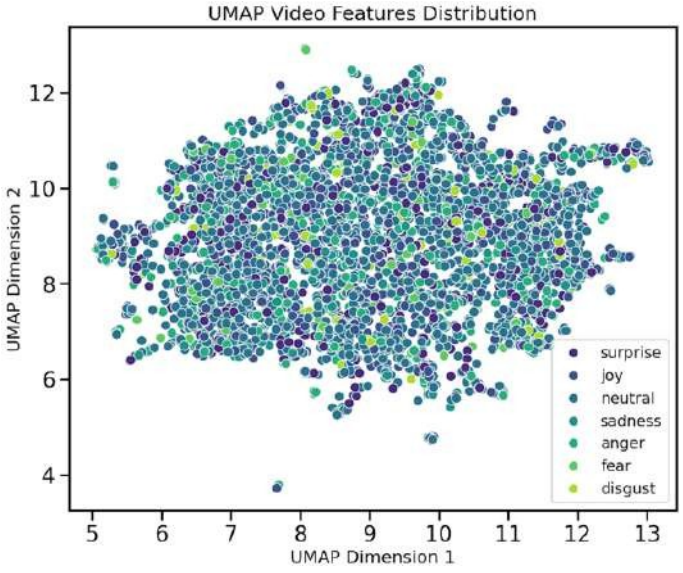


Figure 7.2D UMAP Visualization of Video Features

Key observations from this plot include:

- It is noticeable that many of the points representing different emotions are distributed without strictly separate clusters. This indicates that UMAP has not been able to clearly separate emotions, and it is likely that some emotions have similar features, making it difficult to separate them from each other.

- Despite the lack of strict clustering, it is possible to observe some areas with a higher density of points of certain colors. For example, the points representing the emotions "joy"and "surprise" are closer to each other, which may indicate the similarity of their visual features.

- Since the emotion points overlap to a large extent, this may indicate that the vector representations of the video features are not distinct enough to classify all emotions properly. It may also indicate that the current feature representation (taking frame averages) may be missing important details that are critical to emotion separation.

- Some points are on the periphery of the overall data distribution. These points can be outliers or anomalous videos in which emotions are expressed in an unusual way.

- To improve classification, it is recommended to train the model on mixed data (video, audio, and text), as different modalities can provide additional information for distinguishing emotions.

4.5.3. Textual data

The heatmap demonstrated in Fig. 8 visualizes the correlation matrix of text-based features. Thecolor scale ranges from 0 (blue) to 1 (red), indicating the level of correlation between the features.

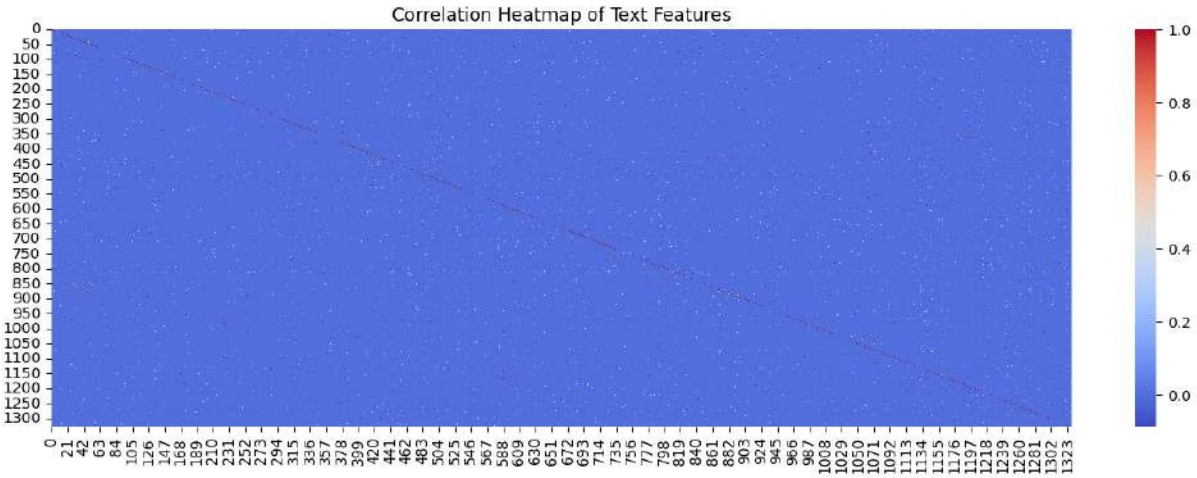


Figure 8.Correlation Heatmap of Text Features.

Key insights related to emotion detection:

- The dominance of blue color suggests low correlations, indicating feature independence. This is beneficial as independent features reduce redundancy, improving the model's abilityto generalize. Therefore, we can confidently use these features without the need for dimensionality reduction.

- Red dots indicate pairs of features with moderate to high correlation, which may signify overlapping information. Figure 9 shows a closer view of these feature relationships. These correlations could point to specific linguistic patterns that are commonly associated with certain emotions (e.g., words or phrases indicative of happiness or sadness). We can examine these pairs further to understand shared emotional cues across features.

- The red diagonal confirms self-correlation, verifying that the feature extraction process preserved the integrity of each feature.

So, most of the features are independent, which is good for models, because it allows the model tobe more accurate and easier to tune since it does not require complex selection procedures or feature reduction.

Figure 9.Examples of Features with Moderate to High Correlation

4.5.4. Insights from audio, video, and text features analysis

In emotion detection using data fusion, the analysis of audio, video, and text features shows key insights for enhancing model performance. The audio features provide diverse and valuable information, but their scattered distribution and the presence of outliers, as seen in the PCA plot, Mel-spectrograms, Chroma features, and MFCCs, suggest the need for careful preprocessing and possibly further dimensionality reduction or clustering to improve emotion classification accuracy.

Analysis using UMAP shows that emotions do not have clear clusters, which makes them difficultto classify. For example, the small distance between the points of "joy" and "surprise" indicates their similarity. Overlapping emotions suggest a lack of clarity in vector representations, which can lead to errors. To improve classification, it is useful to use data from video, audio, and text toaccount for additional aspects of emotions. Moreover, overlapping emotions and a lack of clear clusters may indicate that some emotions are underrepresented or overrepresented in the training dataset. Balancing classes can help improve distinctiveness and classification accuracy, ensuring a more uniform representation of each emotion in the model.

Figure 8 shows that the textual data provides a variety of emotional indicators that are mostly independent, meaning the features do not overlap much. This is great for emotion detection because it cuts down on redundancy, allowing the text to bring in different insights along with the audio and video data.

Combining different types of data allows the emotion detection model to take advantage of all features. For example, audio can capture the subtle details of how people speak, while video shows both subtle and strong emotional expressions. Text adds important context withits meaning and language features. When these elements are brought together, they can enhance the accuracy of emotion detection. By using this approach, it is likely that the model will achievebetter and more accurate emotion detection, as long as effective preprocessing and featureselection techniques are applied to manage variability and potential overlaps between the different modalities.

V. System design

The system design is shown in Figure 10. This section details the processes of data feature extraction and models' utilization, highlighting how audio, visual, and textual data are used to classify emotions.

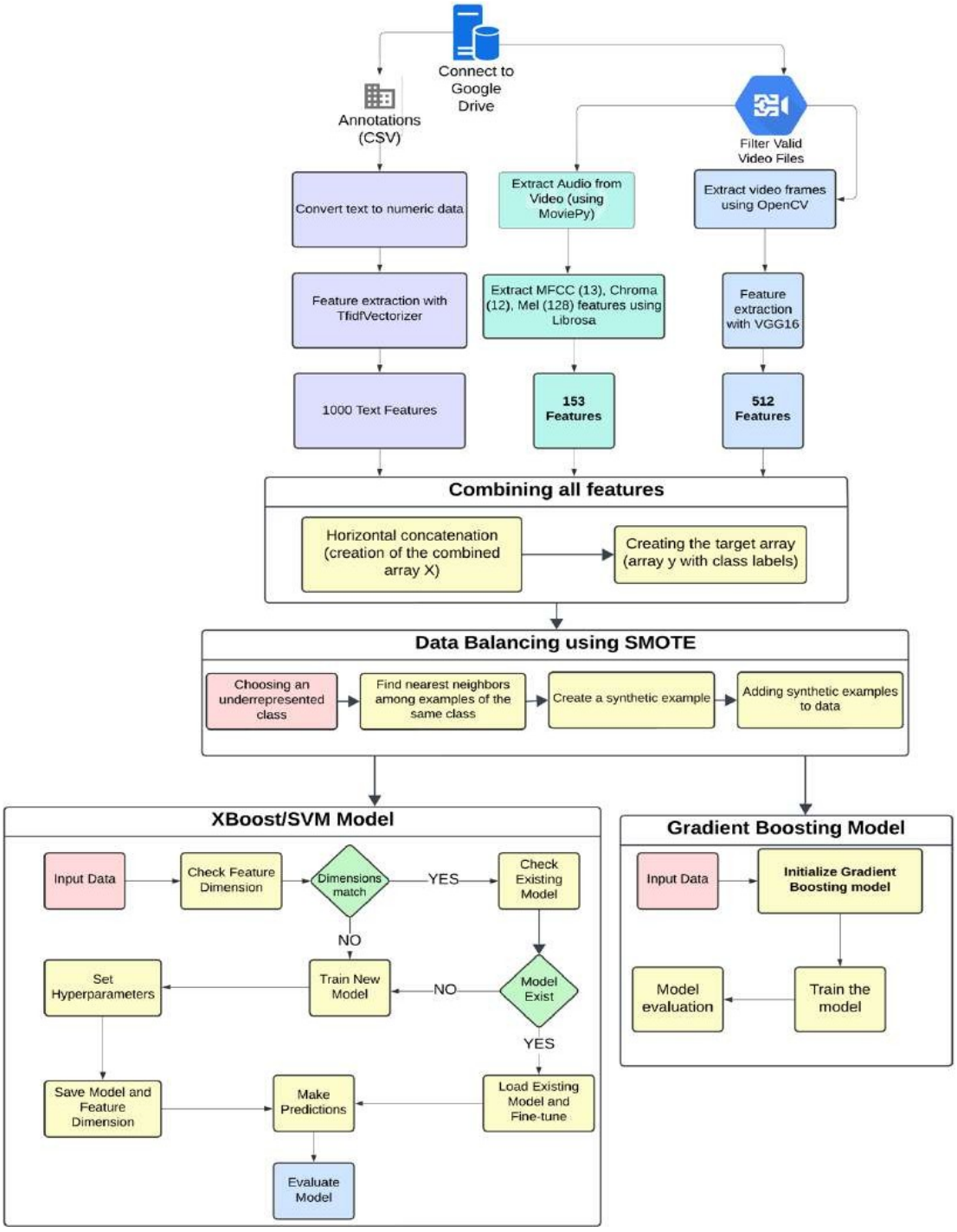


Figure 10.System Design Structure

5.1. Data acquisition

Source: Video files and their corresponding emotional annotations are collected from a specifieddirectory on Google Drive.

Filtering: The system filters the data to ensure only valid files are processed, discarding corrupted or irrelevant videos.

5.2. Feature extraction

The system extracts audio features from the video using the librosa library, which includes:

- **Mel Frequency Cepstral Coefficients**: These coefficients capture the power spectrum of audio signals, representing phonetic properties.
- **Chroma Features:** These features represent the energy distribution across different pitch classes, which helps identify musical characteristics and emotional undertones.
- **Mel Spectrograms:** Visual representations of the spectrum of frequencies in audio signals, providing insights into the temporal aspects of sound.

Visual features are extracted from video frames using the following techniques:

- **Pre-trained VGG16 Convolutional Neural Network (CNN):** A widely used model for feature extraction from images. The VGG16 model is applied to the resized frames to extract features.
- **Frame Extraction:** The video is processed to extract frames at regular intervals (5 frames per video, controlled by the num_frames parameter). Each frame is resized to 224x224 pixels before being passed through the VGG16 model. The extracted features from each frame are stored.
- **Feature Computation:** The system computes the mean of the extracted features across all selected frames to form a 512-dimensional feature vector, summarizing the visual content. If no features are extracted, a zero vector of length 512 is returned.

Textual data, transcriptions, or dialogues from the videos, are processed using TF-IDF (Term Frequency-Inverse Document Frequency) to create numerical representations. A TfidfVectorizer with a maximum of 1000 features is employed to ensure computational efficiency. This captures important words while minimizing the influence of common terms. The vectorizer is saved to maintain consistency in future data transformations.

5.3. Data preprocessing

Normalization: Features gathered from text, audio, and video are standardized using StandardScaler to maintain a consistent scale.

Handling Class Imbalance: To address the uneven distribution of emotional labels,

SMOTE (Synthetic Minority Over-Sampling Technique) is implemented to create synthetic examples for the less represented classes. SMOTE (Synthetic Minority Over-Sampling Technique) addresses class imbalance by generatingsynthetic examples rather than duplicating existing minority class samples [38].

Here's a detailed explanation of how SMOTE functions:

- **Identify Neighbors**: For each sample in the minority class, find its *k* nearest neighbors using a distance metric (commonly Euclidean distance).

- **Select Neighbors**: Based on a specified over-sampling ratio, randomly select *m* neighbors from the *k* nearest neighbors.

- **Generate Synthetic Samples**: Create synthetic samples by interpolating between the minority sample and each selected neighbor using the formula:

$$p_i = x_i + rand(0,1) \times (x_j - x_i),$$

where

$p_i -$ synthetic sample,

$x_i -$ feature vector of the current sample *i* from the minority class,

$x_j -$ selected randomly feature vector of one of the neighbors of x_i,

$rand(0,1) -$ helps in controlling the position of the synthetic sample between x_i and x_j, ensuring that the new synthetic samplelies between the original sample and its neighbors.

- **Continue Until Desired Balance**: Repeat the above steps until the minority class reachesa desired balance with the majority class [39].

There also exists other balancing techniques including Random Undersampling (RUS), which reduces the majority class by discarding samples, potentially losing valuable information, and Random Oversampling, which duplicates minority class samples, risking overfitting [38]. Hybridmethods, such as RUSBoost, combine both undersampling and oversampling to enhance classification performance. Algorithm-level approaches, like Near-Bayesian Support Vector Machine (NBSVM), adjust decision boundaries by varying regularization costs for different classes [39]. However, SMOTE was chosen because it offers several advantages over other balancing techniques. RUS reduces the majority class by discarding samples, which can result in valuable data loss. In contrast, SMOTE maintains all original samples and generates unique synthetic examples for the minority class, preserving dataset size and diversity. Random Oversampling duplicates minority class samples, risking overfitting, while SMOTE creates new samples through interpolation, enhancing learning without redundancy. Hybrid methods, like RUSBoost, combine different techniques but still carry individual risks. Overall, SMOTE effectively improves class representation without compromising data integrity or increasing overfitting risk, making it a better choice for addressing class imbalance in our study.

5.4. Data fusion

The method chosen for data fusion in this project is early fusion, which combines text, audio, andvideo features into a single feature vector before input into the classifier. This technique is favoredover late fusion, where predictions are combined later because early fusion takes advantage of therelationships between different data types at the feature level. By merging information from various sources, early fusion effectively captures the difficult interactions among text, audio, and video signals, crucial for detecting subtle emotional cues. This combined feature representation boosts the model's performance and reliability in emotion recognition tasks.

5.5. Model training

5.5.1. Support Vector Machine (SVM)

The system begins by checking for a pre-trained SVM model (svm_model.pkl) and a feature transformer (transformer.pkl). If these files are found, the model is loaded; if not, a new model is trained and saved. For incremental training, the partial_fit method is used to integrate new data into the model without needing to retrain it from the start. The model is trained using the SGDClassifier, which implements an SVM and allows for the customization of key parameters, such as the loss function, regularization strength, and iteration limits.

The system first checks if an existing model is available for loading or if a new one needs to be trained. After loading the model, the training data is standardized using the transformer. During initial training, SMOTE is applied to balance class distribution. New data is incorporated into the model through incremental learning using partial_fit. The model’s performance is evaluated, and visualizations, such as ROC curves, are generated. Finally, the updated model and transformer are saved for future use.

5.5.2. Gradient Boosting

For the system using Gradient Boosting, it first accepts a dataset with features relevant to emotion classification. The data is then cleaned and formatted to ensure it's ready for model training. The model is initialized with the GradientBoostingClassifier, using parameters like the number of estimators (n_estimators), learning rate (learning_rate), and maximum depth (max_depth). The model's performance is evaluated using various metrics such as accuracy, F1-score, confusion matrix, ROC curve, and AUC.

The system starts by loading and preparing the dataset, splitting it into training and testing sets. The model is then trained on the training set with the specified parameters. After training, the model makes predictions, calculates metrics, and generates a confusion matrix and ROC curves. To better represent the results, visualizations such as the confusion matrix and ROC curves are plotted. Finally, the system returns the trained model and displays all the evaluation metrics.

5.5.3. XGBoost

The system first checks if a saved XGBoost model (XGBoost.model) exists and ensures that the feature dimensions of the new dataset match those used during the initial model training. If there is a mismatch in feature dimensions, the existing model is discarded, and a new model is trained. Trained models are saved using the save_model() function and can be reloaded with

load_model() for fine-tuning. Both models and feature dimensions are stored in a designated directory to ensure consistency across sessions.

Fine-tuning is performed by loading the pre-trained model and adjusting it with new data. While the system facilitates fine-tuning, it does not implement true incremental learning. Incremental learning continuously updates a model as new data arrives, allowing it to adapt over time without forgetting previous knowledge. In contrast, fine-tuning adjusts a pre-trained model to specialize in a new task using new data. Therefore, while incremental learning focuses on evolving a model over time, fine-tuning adapts a model for a specific task after pre-training.

To optimize hyperparameters, such as learning rate, maximum depth, and number of estimators, the system uses GridSearchCV. The steps in the system workflow for XGBoost are as follows: the system first checks the existence of the model and validates feature dimensions. If the feature dimensions do not match, it retrains the model. Then, it either trains a new model or optimizes the existing one, depending on whether fine-tuning is required. The model is updated with new data, evaluated using metrics like accuracy, F1-score, and ROC curves, and visualized with plots such as the ROC curve and confusion matrix. Finally, the model, along with the updated dimensions, is saved for future sessions.

5.6. Model evaluation

Model evaluation is a crucial step in assessing the performance and effectiveness of the trained model. The system employs the following techniques to consistently evaluate a model's performance across different algorithms. After training, the model is evaluated on a separate test dataset (X_test, y_test) to assess its generalization performance. This ensures that the model's predictions are accurate on the training data and can also generalize to unseen data. The system evaluates model performance using key metrics, including accuracy, precision, recall, F1-score, and area under the ROC curve (AUC), providing a comprehensive understanding of performance, especially in multi-class classification. The confusion matrix visualizes classification results, displaying true positives, true negatives, false positives, and false negatives for each class. It helps identify areas for model improvement and is plotted with axes labeled by emotion labels, using color coding for counts. The system generates Receiver Operating Characteristic (ROC) curves for each class, plotting the true positive rate (Recall) against the false positive rate. This visualization tool provides insights into the model's ability to differentiate between classes at various thresholds. The AUC (Area Under the Curve) is calculated to assess the classifier's overall effectiveness, with a higher AUC signifying superior performance. Precision, Recall, and F1-Score are computed for each class, especially useful in imbalanced datasets.

In multiclass classification, the evaluation metrics—accuracy, precision, recall, F1-score, and AUC—are computed differently than in binary classification. Here's how each of these metrics can be calculated for multiclass scenarios:

- **Accuracy:** Calculated as the ratio of correctly predicted instances to the total instances. In a multiclass setting, it is calculated as:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

- **Weighted Precision** for multiclass classification works with precision_score and average='weighted' parameter:

$$Weighted\ Precision = \frac{\sum_{i=1}^{n} w_i \times P_i}{\sum_{i=1}^{n} w_i},$$

where

w_i – the number of instances in class iii (the weight for class *i*),

P_i – Precision for class *i*, which is calculated as:

$$P_i = \frac{TP_i}{TP_i + FP_i},$$

where

FP_i – false positives for the class *i,*

TP_i – true positives for the class *i.*

- **Weighted Recall** can be computed as:

$$Weighted\ Recall = \frac{\sum_{i=1}^{n} w_i \times R_i}{\sum_{i=1}^{n} w_i},$$

where

w_i – the number of instances in class *i* (the weight for class *i*),

R_i – Recall for class *i*, calculated as:

$$R_i = \frac{TP_i}{TP_i + FN_i},$$

where

TP_i – the number of true positives for class *i*,

FN_i – the number of false negatives for class *i*.

- **Weighted F1-Score** combines precision and recall into a single metric:

$$Wheighted\ F1 - Score = 2 \times \frac{Weighted\ Precision \times Weighted\ Recall}{Weighted\ Precision + Weighted\ Recall}.$$

- **Area Under the ROC Curve (AUC):** For each class, treat it as the positive class and all other classes as negative. Calculate the AUC for each binary classification problem, then

average these scores (macro-average).

- **Confusion Matrix:** The confusion matrix is a useful visualization tool for understanding how well your model is performing across all classes. For multiclass classification, the confusion matrix shows counts of true positives, false positives, false negatives, and true negatives for each class: true positives (TP) - correctly predicted instances for a class; false positives (FP) - instances incorrectly predicted as the class; false negatives (FN) - instances that belong to the class but were notpredicted as such; true negatives (TN) - instances that do not belong to the class and werecorrectly predicted as not being the class.

These metrics are beneficial when the class distribution is uneven, as they give more insight thanaccuracy alone while accuracy measures the percentage of correct predictions over the total number of predictions.

5.7. Model persistence

Trained models and transformers are typically saved using joblib, ensuring that the system can resume previous training sessions without loss of progress. However, this approach is not directly applicable to the Gradient Boosting model used in this system. The GradientBoostingClassifier from scikit-learn does not have built-in support for partial fitting or resuming training from a checkpoint. Once the model is trained, it cannot be incrementally updated with new data. Thus, if additional training data is available or if there is a need to retrain, the model must be retrained from the ground up. Models should be saved after the training process to facilitate future predictions, but any updates necessitate retraining with the entire dataset.

5.8. Libraries and Frameworks

This project utilizes a variety of libraries and frameworks essential for building, evaluating, and improving machine learning models for emotion detection. Below is an overview of these tools, detailing their functions and importance.

XGBoost (xgboost library): XGBoost is a popular library for gradient boosting, designed for speed and performance. It is especially efficient for structured data and supports multi-class classification. The framework is based on the principles of decision trees and utilizesa boosting technique to improve prediction accuracy. It is widely used in data science competitions due to its robustness and performance.

Scikit-learn (sklearn library): Scikit-learn is a foundational machine learning library in Python that provides tools for data mining and data analysis. It includes various algorithmsfor classification, regression, clustering, and dimensionality reduction. The framework is built on NumPy, SciPy, and Matplotlib, making it a comprehensive tool for machine learning. Key features include model evaluation metrics and hyperparameter tuning capabilities through GridSearchCV, which helps optimize model settings using cross-validation.

SGDClassifier: This classifier is part of Scikit-learn and implements Stochastic Gradient Descent (SGD), a popular optimization algorithm. It is suitable for large-scale machine learning problems,especially in classification tasks like support vector machines (SVM). The framework is designed to be memory efficient, allowing models to be updated incrementally with new data.

GradientBoostingClassifier: This classifier is also part of Scikit-learn and implements a boosting method to create a strong predictive model by combining multiple weak learners, specifically decision trees. It is effective for handling complex datasets and is built on principles from ensemble learning, where the predictions of several models are combinedto improve overall performance.

NumPy: NumPy is a core library for numerical computations in Python. It provides support for multi-dimensional arrays and matrices, along with a variety of mathematical functions.The framework is essential for performing fast operations on large datasets, making it a cornerstone of scientific computing in Python.

Pandas: Pandas is a powerful data manipulation library that offers data structures such as DataFrames, allowing for easy handling and analysis of structured data. The framework simplifies tasks such as data cleaning, transformation, and aggregation, enabling efficient preparation of data for analysis and modeling.

Matplotlib: Matplotlib is a widely used plotting library for creating static, animated, and interactive visualizations in Python. It allows for the generation of various types of plots and graphs, which are crucial for understanding model performance and trends in data.

OS: The OS library in Python provides a way to interact with the operating system. It is used for file operations, such as checking for existing files, navigating directories, and managing file paths. This functionality is essential for organizing project files and ensuring proper data management.

Time: The Time library is used to track and measure time intervals in Python programs. This is important for understanding the duration of various processes within the project, such as model training and evaluation, helping to identify performance bottlenecks.

Joblib: Joblib is a library for lightweight pipelining in Python. It is used for efficient serialization of Python objects, especially NumPy arrays, allowing for quick saving and loading of models. This feature helps to avoid retraining models every time they are needed, saving both time and computational resources.

Imbalanced-learn (imblearn library): Imbalanced-learn is a library specifically designed for handling imbalanced datasets, where certain classes have significantly more instances than others. It offers various techniques, including SMOTE (Synthetic Minority Oversampling Technique), to create synthetic samples for underrepresented classes. This helpsimprove the performance of models on imbalanced data.

Librosa: Librosa is a Python library for analyzing and processing audio signals. It provides tools forfeature extraction from audio data, making it invaluable for tasks involving sound analysis, such as emotion detection from speech or music.

MoviePy: MoviePy is a Python library for video editing and processing. It allows for tasks such as reading video files, extracting frames, and editing video clips. This functionality is essential for analyzing visual features in videos as part of the emotion detection system.

These libraries and frameworks collectively create a robust environment for developing

an emotion detection system. Their diverse functionalities enable the integration of various data types, including structured data, audio, and video, enhancing the overall effectiveness of the project.

VI. Results and discussion

Three machine learning models, XGBoost, Support Vector Machine (SVM) with SGDClassifier, and Gradient Boosting across multiple iterations, were compared in performance. The study demonstrated only 4 iterations, as further iterations did not lead to significant changes in the resultsthat could affect the conclusions. Four iterations were sufficient to demonstrate and compare the performance of the models, as the subsequent changes were insignificant.

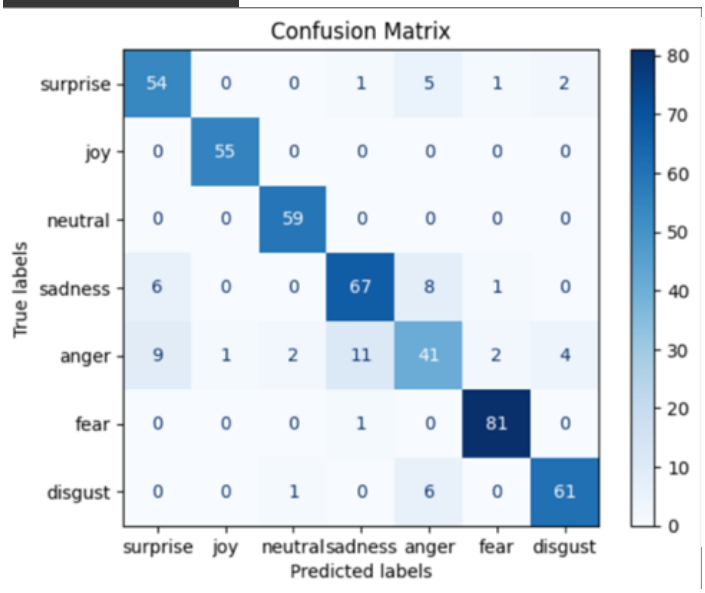
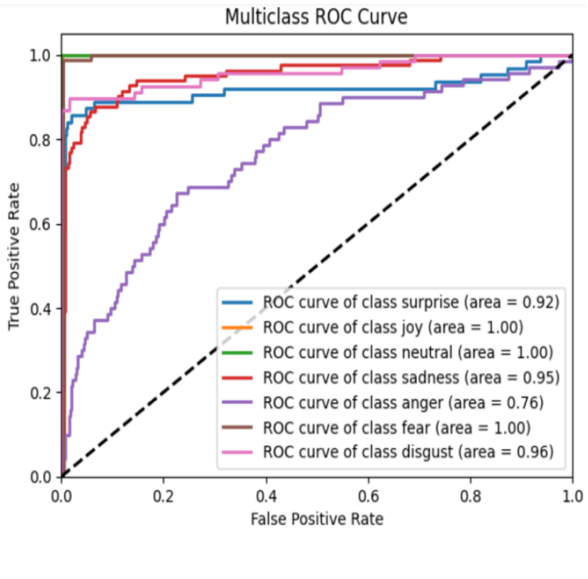
6.1. SVM with SGDClassifier

Table 2 shows the performance of the model across four iterations using the Support Vector Machine (SVM) with a Stochastic Gradient Descent (SGD) classifier for emotion classification. The first four iterations were chosen because there were no significant changes in performance beyond this point, with fluctuations in the results observed instead.

Table 2. SVM model performance

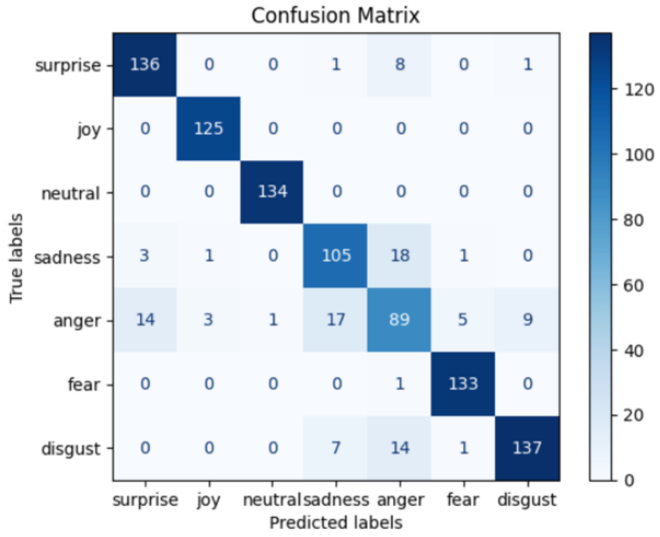
Iteration 1	Iteration 1

`Execution time: 1.21 seconds`
`Accuracy: 0.8727`
`F1-Score: 0.8697`
`Precision: 0.8685`
`Recall: 0.8727`
	`Average ROC AUC: 0.94`

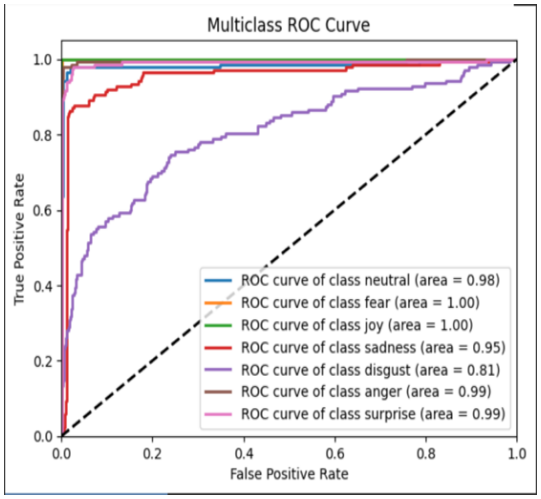
Iteration 2

```
Execution time: 2.29 seconds
Accuracy: 0.8911
F1-Score: 0.8897
Precision: 0.8893
Recall: 0.8911
```



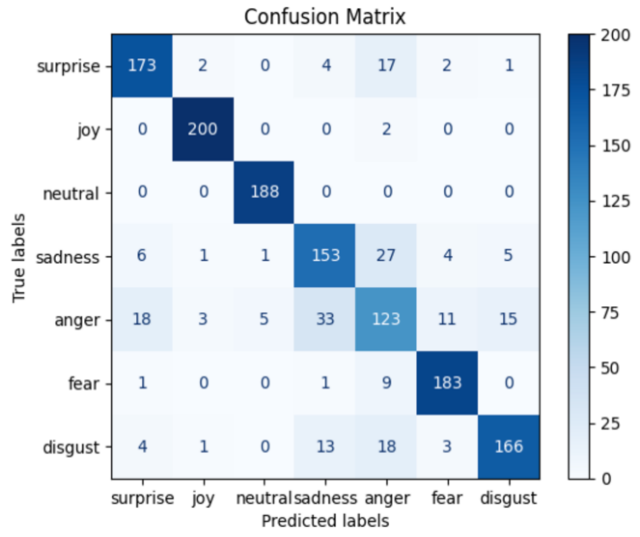
Iteration 2

```
Average ROC AUC: 0.96
```



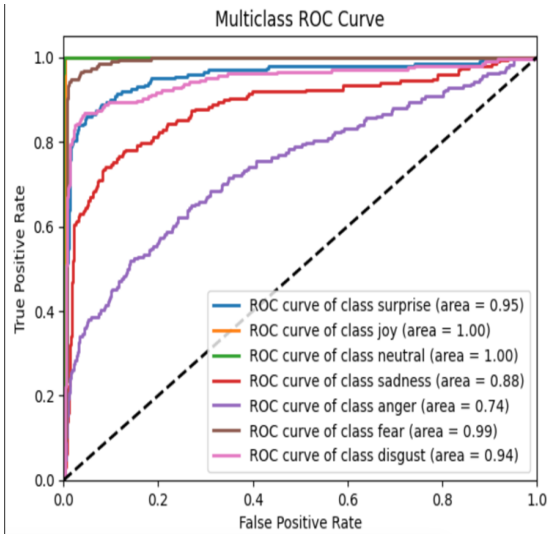
Iteration 3

```
Execution time: 3.58 seconds
Accuracy: 0.8514
F1-Score: 0.8498
Precision: 0.8492
Recall: 0.8514
```



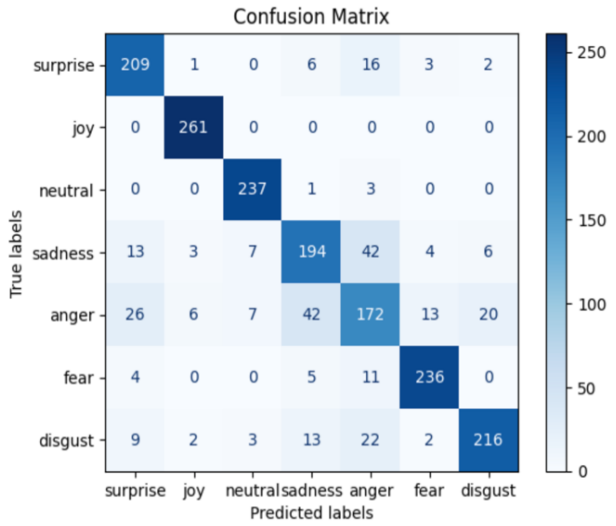
Iteration 3

```
Average ROC AUC: 0.93
```



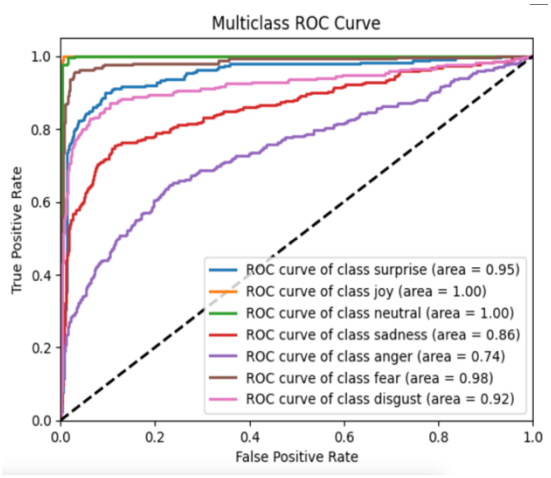
Iteration 4

```
Execution time: 2.75 seconds
Accuracy: 0.8393
F1-Score: 0.8370
Precision: 0.8363
Recall: 0.8393
```



Iteration 4

```
Average ROC AUC: 0.92
```



For emotion classification, an incremental learning approach using a Stochastic Gradient Descent (SGD) classifier for Support Vector Machines (SVM) was implemented. The best achieved results were observed in Iteration 2, with an accuracy of 0.8911 and an F1-score of 0.8897, alongside the highest average ROC AUC of 0.96, indicating excellent performance across all emotion categories. Iteration 1 also performed well, with an accuracy of 0.8727 and an average ROC AUC of 0.94.

Despite slight fluctuations, the model consistently generalized well across emotion categories, as shown by the confusion matrices and ROC curves. "Joy" and "neutral" emotions had higher precision and recall, while "anger" and proved more challenging, with lower recall and precision in some iterations (e.g., Iteration 4). This suggests that classifying ambiguous emotions remains difficult. Incremental learning with early stopping was applied to fine-tune the model and prevent overfitting. The early stopping mechanism halted training when no significant F1-score improvement was detected, and training time increased across iterations, from 1.21 seconds in Iteration 1 to 3.58 seconds in Iteration 3, reflecting the computational cost of handling more data.

The early stopping effectively refined the model, preventing overfitting and optimizing performance in Iteration 2. However, as more data was introduced in Iterations 3 and 4, performance slightly decreased, likely due to the complexity of distinguishing nuanced emotional states. ROC curves across all iterations confirmed robustness, with ROC AUC values consistently above 0.92, peaking at 0.96 in Iteration 2. Incremental learning with early stopping enabled the model to improve continuously while balancing complexity and performance. It achieved high precision and recall for easier-to-distinguish emotions. However, future work should focus on improving the classification of difficult emotions using advanced techniques such as class-specific oversampling or deep learning-based approaches.

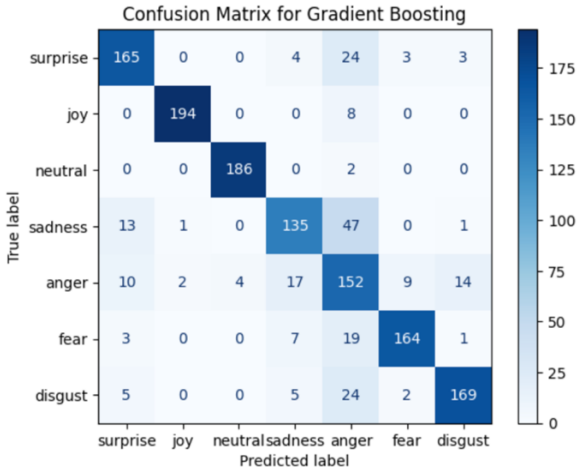
6.2. Gradient Boosting

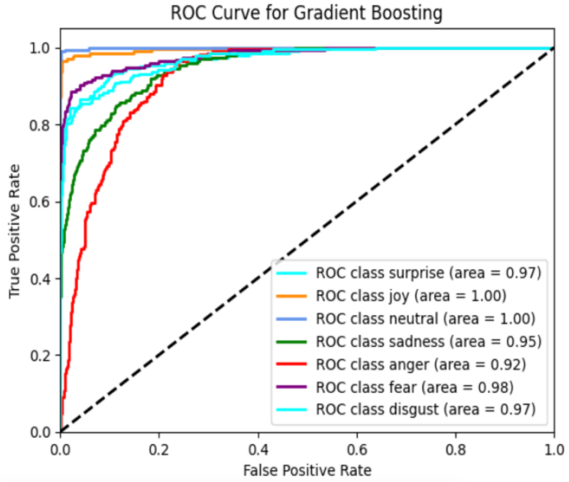
The following results presented in Table 3show the performance of the Gradient Boosting model for emotion classification

Table 3. Gradient Boosting model performance

```
Execution time: 506.38 seconds
Accuracy: 0.8363
F1-Score: 0.8409
Precision: 0.8517
Recall: 0.8363
```

```
Average ROC AUC: 0.97
```





The confusion matrix shows that the model performs well for “joy” and “neutral”, with hardly any mistakes. However, it struggles with emotions related to the same sentiment or neutral, sometimes mistaking “sadness” for “neutral” or “anger”. This highlights how tough it can be to tell apart emotions that look or sound alike.

The ROC curves show how well the model can tell emotions apart. It scored a perfect AUC of 1.00 for “joy” and “neutral”, showing great at identifying those feelings. Other emotions, like “surprise” and “disgust”, did well, too, scoring 0.97. But there was a bit of a drop for anger, which scored 0.92. This suggests the model is strong overall, but there is room for improvement in distinguishing between subtle emotions like “anger” and “sadness”.

Additionally, the micro-average and macro-average ROC AUC scores are at 0.97, confirming that the model generalizes well across all emotion categories. These findings indicate that Gradient Boosting is effective at classifying emotions with a low rate of false positives, making it a good choice for this task of recognizing different emotions.

One limitation of the Gradient Boosting model is that it does not inherently support incremental learning. Gradient Boosting operates as a batch learning algorithm, requiring the entire dataset to be available before training. It cannot incrementally update its model as new data arrives without retraining from scratch. This can be a drawback in dynamic environments where data is continuously being collected, as the model would need to be retrained entirely to

incorporate new information.

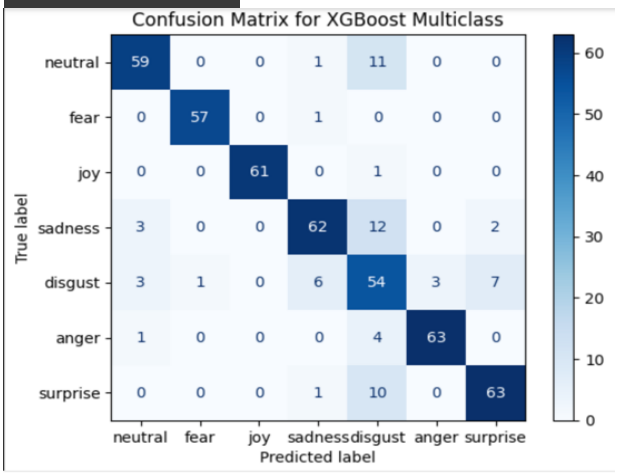
6.3. XGBoost training with learning rate adjustment

The XGBoost classifier's performance was evaluated over multiple iterations, with metrics such as accuracy, F1-score, precision, recall, and confusion matrices used to analyze classifications across seven emotion categories and presented in Table 4.

Table 4. XGBoost model performance

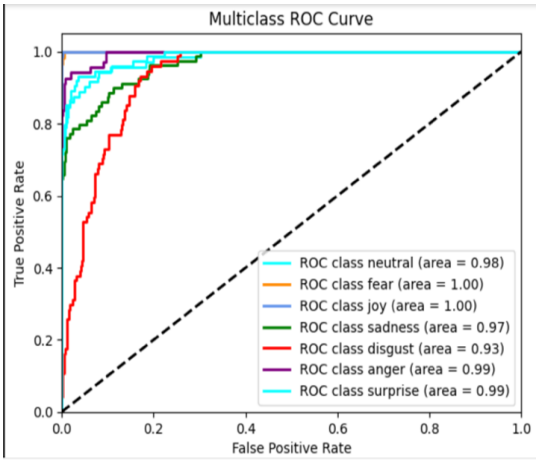
Iteration 1

```
New model trained in 540.17 seconds
Accuracy: 0.8621
F1-Score: 0.8661
Precision: 0.8736
Recall: 0.8621
```



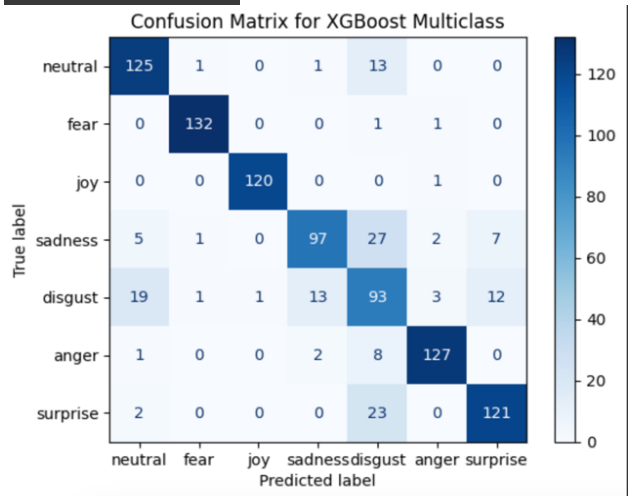
Iteration 1

```
Average ROC AUC: 0.98
```



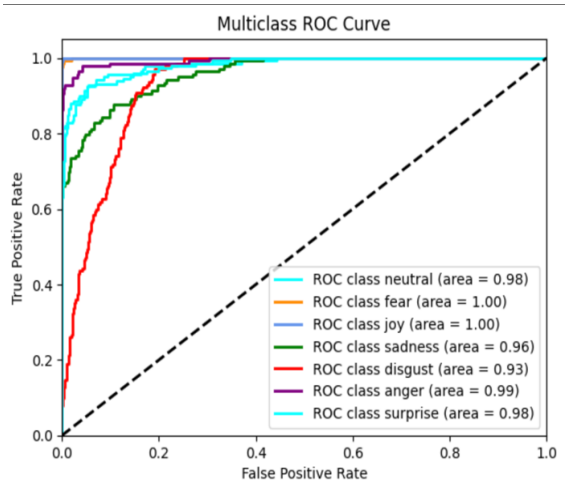
Iteration 2

```
New model trained in 66.39 seconds
Accuracy: 0.8490
F1-Score: 0.8509
Precision: 0.8568
Recall: 0.8490
```



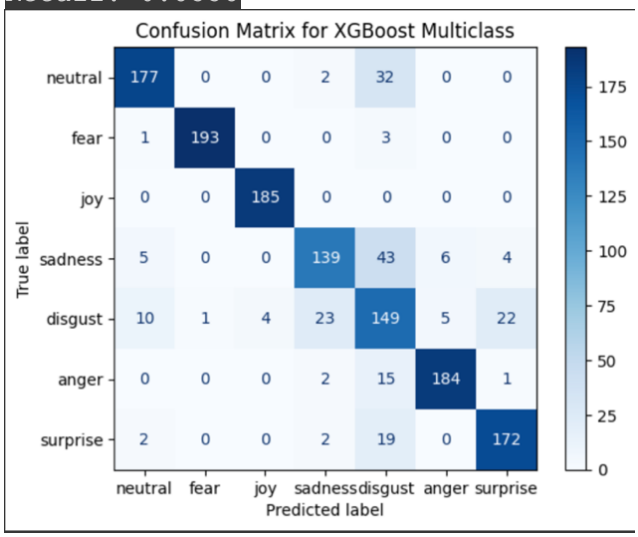
Iteration 2

```
Average ROC AUC: 0.98
```



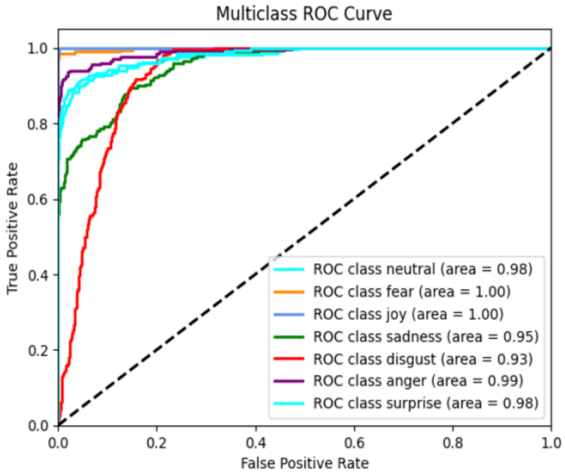
Iteration 3

```
New model trained in 127.52 seconds
Accuracy: 0.8558
F1-Score: 0.8589
Precision: 0.8657
Recall: 0.8558
```



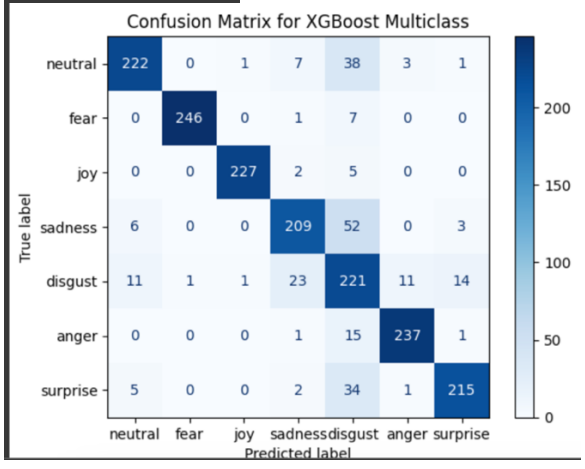
Iteration 3

```
Average ROC AUC: 0.98
```



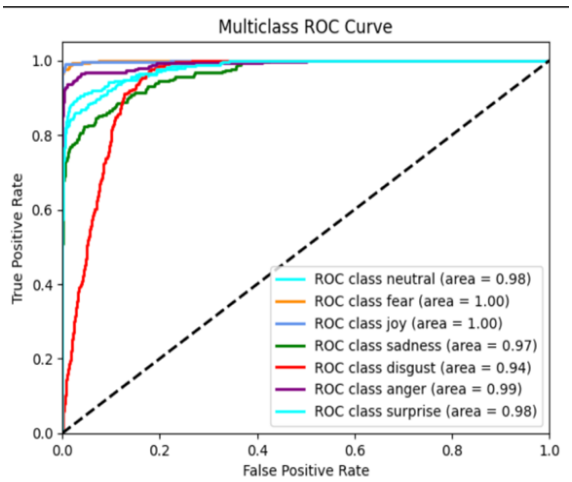
Iteration 4

```
Model fine-tuned in 117.17 seconds
Accuracy: 0.8651
F1-Score: 0.8698
Precision: 0.8806
Recall: 0.8651
```



Iteration 4

```
Average ROC AUC: 0.98
```



Iteration 1 involved training a new model from scratch, achieving an accuracy of 0.8621 and an F1-score of 0.8661 in 540.17 seconds. The confusion matrix showed balanced predictions, but confusion persisted between some emotional states, such as “joy” and “neutral” or “sadness”.

Iteration 2 fine-tuned the model, reducing training time to 66.39 seconds but slightly decreasing accuracy to 0.8490 and the F1-score to 0.8509. Confusion between emotions like “joy” and “surprise” increased.

Iteration 3 further fine-tuned the model, increasing accuracy to 0.8558 and the F1 Score to 0.8589. However, the classification time is also doubled and reached 127.52 seconds. The classification of "fear" and "anger" improved, though sadness continued to be more misclassified, especially with "disgust" and "fear".

Iteration 4 achieved an accuracy of 0.8651 and the highest F1-score of 0.8698 in 117.17 seconds, with the better recognition for "joy" and "surprise". However, "sadness" and "disgust" still showed higher misclassification rates.

Further iterations did not lead to significant changes in the results, only small fluctuations were observed, so only 4 iterations were demonstrated. Across iterations, fine-tuning slightly improved performance but showed diminishing returns after Iteration 2. While the model adapted well to new data, challenges in classifying negative emotions persisted. Further hyperparameter tuning could improve classification, particularly for overlapping emotional states. Incremental learning allowed the model to be fine-tuned with new data without retraining from scratch, significantly reducing training times while maintaining performance. Other than that, multiclass ROC curve analysis confirmed strong performance across most emotions, particularly "joy", "anger", and "surprise".

6.4. Summary

Based on the provided metrics, SVM with SGDClassifier demonstrates the highest accuracy and F1-score in its second iteration, achieving 89.1% accuracy and an F1-score of 89.0%. This indicates strong performance in emotion classification. XGBoost also shows competitive results, particularly with an average ROC AUC of 0.98, but with slightly lower accuracy and F1-scores compared to SVM. Gradient Boosting, while performing adequately, exhibits the lowest metrics across the board and struggles with closely related emotions, making it the least effective model in this comparison. It is also important to note that after fine-tuning or incremental learning the model metrics, such as the numbers in the confusion matrix, can change because new data is added with each iteration. In summary, the rankings for the models based on overall performance are as follows:

1. SVM with SGDClassifier
2. XGBoost
3. Gradient Boosting

VII. Scalability and future work

The investigated models demonstrate promising performance in emotion classification, yet scalability remains a challenge, particularly for the Gradient Boosting model, which operates as a batch learning algorithm. This limitation means the model requires the entire dataset for training, making it unsuitable for dynamic environments where data continuously arrives. Future iterations of the model could explore online learning algorithms allowing incremental updates, thus enhancing scalability and responsiveness to new data.

To further improve the accuracy of emotion classification, especially for overlapping emotional states, future work could focus on the following strategies:

Feature Engineering: Developing additional features that better capture the nuances of emotions could enhance model performance. Techniques like dimensionality reduction and selecting more discriminative features might prove beneficial.

Data Augmentation: Increasing the diversity and volume of training data through augmentation techniques could help models generalize better and reduce overfitting. This could include generating synthetic data for underrepresented emotions.

Ensemble Methods: Combining multiple models through ensemble techniques may improve classification accuracy and robustness. Techniques like stacking or blending could use thestrengths of each model to produce superior results.

Deep Learning Approaches: Investigating deep learning frameworks, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), may yield further improvements in emotion recognition from audio and visual inputs. These architectures can capture more complex patterns in data, potentially outperforming traditional machine learning models.

Hyperparameter Tuning: More **careful** hyperparameter tuning can help boost model performance, especially for tricky emotional states. Automated methods like Bayesian optimization could make this tuning process even better.

Cross-domain Validation: Testing how the model performs across different datasets or domainsis a good way to check if it is robust and can work well in various situations. Implementing thesestrategies can further optimize models for real-world applications, leading to more accurate emotion classification in dynamic and diverse settings.

VIII. Conclusion

In conclusion, the comparison of SVM with SGDClassifier**,** Gradient Boosting, and XGBoost highlights the strengths and weaknesses of each model in emotion classification. In this experiment, SVM exhibited excellent precision and recall, particularly in identifying distinct emotions, although it faced challenges in classifyingmore nuanced emotional states. The Gradient Boosting model, while powerful, faced limitations due to its batch-learning nature, making it less adaptable to real-time data. The XGBoost demonstrated effective fine-tuning capabilities, allowing for quicker training times while maintaining strong performance metrics.

The performance of the three machine learning models—SVM with SGDClassifier, Gradient Boosting, and XGBoost—was assessed across multiple iterations, focusing on key metrics such asaccuracy, F1-score, precision, recall, and ROC AUC. Table 2 provides an overview of the results,illustrating the strengths and limitations of each model in emotion classification tasks.

Despite these challenges, all models achieved quite high accuracy and AUC scores across the MELD dataset, indicating their potential for practical application in emotion recognition tasks. The findings underscore the importance of iterative model refinement, and future enhancements focusing on scalability, feature engineering, and advanced learning techniques are crucial for improving overall performance and adaptability. As emotion recognition systems become

increasingly integrated into various technologies, ongoing research and development will be vital to meet the growing demands for accurate and responsive models.

Table 5. Results overview

Model	Iter. №	Accuracy, %	F1-Score, %	Precision %	Recall %	ROC AUC %	Time, sec	Comments
SVM with SGD Classifier	1	87.3	86.8	86.9	87.3	94	1.2	Strong initial performance across all metrics demonstrates the model's effectiveness. Despite high accuracy, someconfusion among emotions indicates room for improvement.
	2	89.1	89.0	88.9	89.1	96	2.3	
	3	85.1	85.0	84.9	85.1	93	3.6	
	4	83.9	83.7	83.6	83.9	92	2.8	
Gradient Boosting	1	83.6	84.1	85.2	83.6	97	506.4	The model achieves reasonable accuracy but struggles with closely related emotions, suggesting that feature enhancement could improve classification performance.
XGBoost	1	86.2	86.6	87.4	86.2	98	540.2	XGBoost shows strong metrics, indicating solid effectiveness in emotion detection; however, challenges in distinguishing similar emotional states remain a concern.
	2	84.9	85.1	85.7	84.9	98	66.39	
	3	85.6	85.9	86.6	85.6	98	127.5	
	4	86.5	87.0	88.1	86.5	98	117.2	

Finally, this research has effectively addressed the questions posed regarding emotion detectionthrough data fusion:

How does data fusion perform in emotion recognition tasks?

The study part based on the literature review demonstrates that the fusion of audio, video, and textual data significantly enhances the performance of emotion recognition tasks. By using the strengths of each modality, the combined model achieves higher accuracy compared to models using a single data source because in real life people can express nuanced emotions, use sarcasm, etc. The integration allows a more comprehensive understanding of emotional expressions, leading to improved detection capabilities.

Which machine learning models are most effective in handling multimodal data for emotiondetection?

The research evaluates the performance of three models on the MELD dataset: Support Vector Machine (SVM) with SGDClassifier, Gradient Boosting, and XGBoost.

- **SVM with SGDClassifier** achieved the highest accuracy in the second iteration, excelling in recognizing emotions such as “joy” and “neutral,” but faced difficulties with “anger.”

- **Gradient Boosting** showed reliable performance for specific emotions but struggled with distinguishing similar emotional states and lacks support for incremental learning.

- **XGBoost** demonstrated strong overall performance across all metrics, effectively classifying a range of emotions, although it encountered challenges with overlapping emotional states.

In summary, XGBoost and SVM were the most effective models among the three tested on the MELD dataset for handling multimodal data in emotion detection, offering complementary strengths for this task.

IX. References

[1] Ekman, P. (1992). An argument for basic emotions. *Cognition and Emotion*, 6(3-4), 169-200.https://doi.org/10.1080/02699939208411068

[2] Calvo, R. A., & D'Mello, S. (2010). Affect detection: An interdisciplinary review of models, methods, and theirapplications. IEEE Transactions on Affective Computing, 1(1), 18-37. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5520655

[3] Picard, R. W. (2000). *Affective computing*. MIT press. Retrieved from https://cs.uwaterloo.ca/~jhoey/teaching/cs886-affect/papers/Picard AffectiveComputing/9780262281584_chap6.pdf

[4] Soleymani, M., Lichtenauer, J., Pun, T., & Pantic, M. (2017). A multimodal database for affect recognition andimplicit tagging. *IEEE Transactions on Affective Computing*, *3*(1), 42-55. DOI: 10.1109/T-AFFC.2011.25

[5] Sharma, A., Sharma, K. & Kumar, A. Real-time emotional health detection using fine-tuned transfer networkswith multimodal fusion. *Neural Comput & Applic* **35**, 22935–22948 (2023). https://doi.org/10.1007/s00521-022- 06913-2

[6] Baltrusaitis, T., Ahuja, C., & Morency, L. P. (2019). Multimodal machine learning: A survey andtaxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(2), 423-443. doi:10.1109/tpami.2018.2798607

[7] Jiehui Jia, Huan Zhang, Jinhua Liang, *Bridging Discrete and Continuous: A Multimodal Strategy for ComplexEmotion Detection.* arXiv preprint arXiv:2409.07901. 2024. Available at: https://doi.org/10.48550/arXiv.2409.07901

[8] Rafael Pereira, Carla Mendes, José Ribeiro, Roberto Ribeiro, Rolando Miragaia, Nuno Rodrigues, Nuno Costa, António Pereira, *Systematic Review of Emotion Detection with Computer Vision and Deep Learning.* Sensors 2024,24(11), 3484. https://doi.org/10.3390/s24113484

[9] Kenton, J. D. M. W. C., & Toutanova, L. K. (2019, June). Bert: Pre-training of deep bidirectional transformersfor language understanding. In *Proceedings of naacL-HLT* (Vol. 1, p. 2). https://doi.org/10.48550/arXiv.1810.04805

[10] Hosen, M. H. (2023). *Multimodal emotion recognition in conversation*. Kaggle. Version 1. MultimodalEmotionLines Dataset (MELD), MELD_files. Copied from Pratap, A. Retrieved from https://www.kaggle.com/code/hosen42/multimodal-emotion-recognition-in-conversation

[11] Poria, S., Cambria, E., Bajpai, R., & Hussain, A. (2017). A review of affective computing: From unimodalanalysis to multimodal fusion. *Information Fusion*, 37, 98-125.

https://doi.org/10.1016/j.inffus.2017.02.003

[12] Poria, S., Cambria, E., Hazarika, D., & Majumder, N. (2020). Multimodal sentiment analysis: Addressing keyissues and setting future directions. *IEEE Transactions on Affective Computing*, 12(2), 17-34. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9260964

[13] Jiang, Y., Li, W., Hossain, M. S., Chen, M., Alelaiwi, A., & Al-Hammadi, M. (2020). A snapshot research and implementation of multimodal information fusion for data-driven emotion recognition. *Information Fusion*, *53*, 209-221. https://www.sciencedirect.com/science/article/pii/S1566253519301381

[14] Pepa, L., Spalazzi, L., Capecci, M., & Ceravolo, M. G. (2021). Automatic emotion recognition in clinicalscenario: a systematic review of methods. *IEEE Transactions on Affective Computing*, *14*(2), 1675-1695. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9618863

[15] Vistorte, A. O. R., Deroncele-Acosta, A., Ayala, J. L. M., Barrasa, A., López-Granero, C., & Martí-González, M. (2024). Integrating artificial intelligence to assess emotions in learning environments: a systematic literaturereview. *Frontiers in Psychology*, *15*, 1387089. https://doi.org/10.3389/fpsyg.2024.1387089

[16] Zepf, S., Hernandez, J., Schmitt, A., Minker, W., & Picard, R. W. (2020). Driver emotion recognition forintelligent vehicles: A survey. *ACM Computing Surveys (CSUR)*, *53*(3), 1-30. https://dl.acm.org/doi/pdf/10.1145/3388790

[17] Zadeh, E. K., & Alaeifard, M. (2023). Adaptive Virtual Assistant Interaction through Real-Time SpeechEmotion Analysis Using Hybrid Deep Learning Models and Contextual Awareness. *International Journal of Advanced Human Computer Interaction*, *1*(1), 1-15. https://www.ijahci.com/index.php/ijahci/article/view/9

[18] Naveenkumar, M., & Kaliappan, V. K. (2019, November). Audio based emotion detection and recognizing tool using mel frequency based cepstral coefficient. In *Journal of Physics: Conference Series* (Vol. 1362, No. 1, p. 012063). IOP Publishing. DOI 10.1088/1742-6596/1362/1/012063

[19] Garg, U., Agarwal, S., Gupta, S., Dutt, R., & Singh, D. (2020, September). Prediction of emotions from theaudio speech signals using MFCC, MEL and Chroma. In *2020 12th international conference on computational intelligence and communication networks (CICN)* (pp. 87-91). IEEE. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9242635

[20] Yang, S., & Chong, X. (2021). Study on feature extraction technology of real-time video acquisition based ondeep CNN. *Multimedia Tools and Applications*, *80*(25), 33937-33950. https://link.springer.com/content/pdf/10.1007/s11042-021-11417-7.pdf

[21] Abdallah, T. B., Elleuch, I., & Guermazi, R. (2021). Student behavior recognition in classroom using deeptransfer learning with VGG-16. *Procedia Computer Science*, *192*, 951-

960. https://doi.org/10.1016/j.procs.2021.08.098

[22] Uymaz, H. A., & Metin, S. K. (2022). Vector based sentiment and emotion analysis from text: A survey. *Engineering Applications of Artificial Intelligence*, *113*, 104922. https://doi.org/10.1016/j.engappai.2022.104922

[23] Bharti, S. K., Varadhaganapathy, S., Gupta, R. K., Shukla, P. K., Bouye, M., Hingaa, S. K., & Mahmoud, A. (2022). Text-Based Emotion Recognition Using Deep Learning Approach. *Computational Intelligence and Neuroscience*, *2022*(1), 2645381. https://doi.org/10.1155/2022/2645381

[24] Zadeh, A., Mao, C., Shi, K., Zhang, Y., Liang, P. P., Poria, S., & Morency, L. P. (2019). Factorized multimodal transformer for multimodal sequential learning. *arXiv preprint arXiv:1911.09826*. https://arxiv.org/pdf/1911.09826

[25] Njoku, J. N., Caliwag, A. C., Lim, W., Kim, S., Hwang, H., & Jung, J. (2022). Deep learning based data fusion methods for multimodal emotion recognition. *The Journal of Korean Institute of Communications and Information Sciences*, *47*(1), 79-87. Retrieved from https://www.researchgate.net/profile/Judith-Njoku-2/publication/358947897_Deep_Learning_Based_Data_Fusion_Methods_for_Multimodal_Emotion_Recognition/links/621ed1fe7106690c08532302/Deep-Learning-Based-Data-Fusion-Methods-for-Multimodal-Emotion- Recognition.pdf

[26] Xie, J., Wang, J., Wang, Q., Yang, D., Gu, J., Tang, Y., & Varatnitski, Y. I. (2023). A multimodal fusion emotion recognition method based on multitask learning and attention mechanism. *Neurocomputing*, *556*, 126649. https://doi.org/10.1016/j.neucom.2023.126649

[27] Jair Cervantes, Farid Garcia-Lamont, Lisbeth Rodríguez-Mazahua, Asdrubal Lopez, A comprehensive survey on support vector machine classification: Applications, challenges and trends, Neurocomputing, Volume 408, 2020, (pp. 189-215). https://doi.org/10.1016/j.neucom.2019.10.118

[28] Anowar, F., & Sadaoui, S. (2021, June). Incremental Learning with Self-labeling of Incoming High- dimensional Data. In *Canadian AI*. Retrieved from https://assets.pubpub.org/wp5fbf35/21621481311278.pdf

[29] Badirli, S., Liu, X., Xing, Z., Bhowmik, A., Doan, K., & Keerthi, S. S. (2020). Gradient boosting neural networks: Grownet. *arXiv preprint arXiv:2002.07971*. https://doi.org/10.48550/arXiv.2002.07971

[30] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794). https://doi.org/10.1145/2939672.293978

[31] Kumar, K. (2024, June 17). *Incremental learning with XGBoost: Examples and insights*. Medium. Retrieved from https://medium.com/@kirankumar_61999/insights-from-

training-xgboost-models-incrementally- dd4dddfe1457

[32] Bnojavan. (2022). *EmoReact: A multimodal emotion dataset of children* [Data set]. GitHub. Retrieved fromhttps://github.com/bnojavan/EmoReact

[33] CMU-MultiComp-Lab. (2023). *CMU-Multimodal SDK: Tools for multimodal data analysis* [Software]. GitHub.Retrieved from https://github.com/CMU-MultiComp-Lab/CMU-MultimodalSDK

[34] S. Poria, D. Hazarika, N. Majumder, G. Naik, R. Mihalcea, E. Cambria. MELD: A Multimodal Multi-PartyDataset for Emotion Recognition in Conversations. (2018). https://doi.org/10.48550/arXiv.1810.02508

[35] Chen, S.Y., Hsu, C.C., Kuo, C.C. and Ku, L.W. EmotionLines: An Emotion Corpus of Multi-PartyConversations. arXiv preprint arXiv:1802.08379 (2018). https://doi.org/10.48550/arXiv.1802.08379

[36] Greenacre, M., Groenen, P. J., Hastie, T., d'Enza, A. I., Markos, A., & Tuzhilina, E. (2022). Principal component analysis. *Nature Reviews Methods Primers*, *2*(1), 100. https://doi.org/10.1038/s43586-022-00184-w

[37] Ghojogh, B., Ghodsi, A., Karray, F., & Crowley, M. (2021). Uniform manifold approximation and projection(UMAP) and its variants: tutorial and survey. *arXiv preprint arXiv:2109.02508*. https://doi.org/10.48550/arXiv.2109.02508

[38] Wang, S., Dai, Y., Shen, J., & Xuan, J. (2021). Research on expansion and classification of imbalanced databased on SMOTE algorithm. *Scientific reports*, *11*(1), 24039. Retrieved from https://www.nature.com/articles/s41598-021-03430-5.pdf

[39] Hasib, K. M., Iqbal, M. S., Shah, F. M., Mahmud, J. A., Popel, M. H., Showrov, M. I. H., ... & Rahman, O. (2020). A survey of methods for managing the classification and solution of data imbalance problem. *arXiv preprintarXiv:2012.11870*. https://doi.org/10.3844/jcssp.2020.1546.1557