SENIOR PROJECT CN3-2 – 1/2024

Multi-Channel Sentiment Analysis

Submitted to

School of Information, Computer and Communication Technology
Sirindhorn International Institute of Technology
Thammasat University

December 2024

By

Napat	Ariyapattanaporn	6422782399
Teetawat	Bussabarati	6422782423

Advisor: Dr. Cholwich Nattee

Table of Contents

Acknowledgment

We would like to express our sincere gratitude to everyone who has contributed to the successful completion of this senior project.

First and foremost, we extend our heartfelt thanks to our supervisor, Dr. Cholwich Nattee, for his invaluable guidance, expertise, and unwavering support throughout this project. His insightful feedback and encouragement were instrumental in shaping the direction of this research and overcoming numerous challenges.

We are also grateful to the Sirindhorn International Institute of Technology (SIIT) for providing us with the necessary resources and facilities to conduct this project.

Lastly, we would like to thank all the individuals who directly or indirectly contributed to the successful completion of this project. Your support and encouragement have been invaluable.

Abstract

This project introduces a sentiment analysis system that integrates three distinct data channels: facial expression, voice tone, and speech transcription. Each channel independently analyzes sentiment, classifying emotions as either positive or negative. Through preprocessing the outputs and applying a voting mechanism, the system provides an alternative, lightweight, and more explainable approach to sentiment analysis compared to traditional multimodal techniques, which rely on early fusion within neural networks. Our approach is designed to work alongside existing multimodal methods, offering customizable edge case detection, while allowing the use of multimodal systems for more standard cases. This combination enhances flexibility and transparency in human emotion analysis, allowing researchers to tailor solutions for specific needs.

Chapter 1

1.1 Introduction

The ability to accurately detect and understand human emotions is a key factor in various technological applications, virtual avatars that dynamically adjust their facial expressions to match the user's emotions, systems that monitor mental health, and solutions for enhancing customer service experiences.

Historically, sentiment analysis systems have primarily relied on a single data channel, such as textual data, to interpret and classify emotional states. However, human emotions are complex and expressed through a combination of verbal and non-verbal cues. Traditional single-channel sentiment analysis may miss subtle emotional signals, failing to accurately reflect a person's true emotional state. For instance, someone might feel sad but mask it, or sound angry but be joking. Therefore, it is essential to consider not just the content of spoken words but also the tone of voice and facial expressions. Each of these channels provides distinct insights into emotional states:

1. Facial recognition detects visual cues that often convey emotions more immediately and universally than verbal expression.
2. Speech transcription provides semantic content, allowing for analysis of word choice and linguistic patterns associated with different emotions.
3. Voice tone analysis captures vocal features that can reveal emotions not explicitly stated in words.

For example, while text-based systems may be able to interpret the meaning behind words, they might miss the nuances conveyed through the speaker's tone or facial expressions. As noted by Nandwani and Verma in their 2021 paper, *"A review on sentiment analysis and emotion detection from text",* "… in some cases, machine learning models fail to extract some implicit features or aspects of the text …", leading to inaccurate sentiment results when solely relying on textual data.

This limitation has driven the development of traditional multimodal approaches, where early fusion techniques within neural networks combine data from multiple sources (such as text, voice tone, and facial expressions). While these multimodal systems improve the understanding of emotions, they also

introduce significant technical challenges, including handling noise, dealing with missing data, and interpreting the "black box" nature of neural networks. When one channel has a low confidence score or missing data, these systems can produce larger errors without fallback mechanisms to re-evaluate, which can affect overall accuracy. Additionally, neural networks are often opaque, making them difficult to interpret.

Our project proposes an alternative method: a voting mechanism applied after each channel has independently analyzed sentiment. This late fusion approach allows for greater interpretability and customizability, enabling researchers to handle edge cases such as conflicting emotional signals (e.g., facial expressions suggesting one emotion, while voice tone implies another). Using techniques like bagging or boosting, our voting system detects these conflicts and adjusts the final sentiment classification accordingly.

This system is not positioned as superior to state-of-the-art multimodal techniques. Instead, it serves as a complementary tool. In scenarios where multimodal systems work well, they should continue to be used. However, for edge cases and situations where there is ambiguity or conflict between signals, our voting mechanism provides a clearer, more transparent method of emotion detection. It allows users to combine both approaches: relying on multimodal systems for normal cases and utilizing our voting logic for more complex or nuanced cases.

1.2 Summary

This project aims to develop a novel sentiment analysis system that leverages the power of multiple data channels (facial expressions, voice tone, and speech transcription) to provide a more accurate and interpretable understanding of human emotions. By combining these channels through a voting mechanism, we aim to address the limitations of traditional single-channel and complex multimodal approaches. Our system offers a more transparent and flexible solution, particularly for handling edge cases where emotional signals may conflict or be ambiguous. This research contributes to the advancement of human-computer interaction, enabling the development of more empathetic and intelligent systems.

1.3 Motivation

Understanding human emotions is vital across diverse applications, from enhancing virtual interactions to improving mental health monitoring and customer service experiences. However, traditional sentiment analysis systems often rely on single data channels, such as text, which may fail to capture the complexity of emotions expressed through verbal and non-verbal cues. Multimodal approaches have emerged to address this limitation, integrating inputs like facial expressions, voice tone, and text.

While effective, these systems face challenges such as handling noisy or missing data and interpreting results due to the "black box" nature of neural networks. To address these issues, our project introduces a late fusion voting mechanism, enabling independent analysis of each channel before combining results. This approach improves interpretability and robustness, offering a valuable tool for handling edge cases and conflicting emotional signals. It complements state-of-the-art systems, enhancing their reliability in nuanced or ambiguous scenarios.

Chapter 2

2.1 Background

Emotion detection is an important area of research within affective computing, which aims to build systems that can recognize, interpret, and respond to human emotions. This technology is used in a variety of fields, such as virtual assistants, customer service tools, mental health monitoring, and human-robot interaction. Understanding human emotions is challenging because people express emotions in many different ways, such as through facial expressions, voice tone, and words. To truly understand someone's emotions, these different cues need to be analyzed together, as humans often communicate using a mix of facial expressions, speech, and tone.

In the past, many emotion detection systems have relied on single-channel models, which focus on just one source of information to detect emotions. For example, text-based models analyze the words a person uses, looking for patterns in language that suggest emotions. Facial expression recognition systems focus on analyzing a person's face, looking at features like how the mouth or eyes move to detect emotions. Finally, voice tone analysis systems examine the sound of the voice, including factors like pitch, speed, and volume, to determine whether a person is happy, angry, or sad.

While these single-channel models have improved over time, even when trained on large, high-quality datasets (such as AffectNet for facial expressions or EmoDB for voice tone), they still have significant limitations. Despite their success in research papers and on benchmark platforms like Paperswithcode.com, these models often struggle to accurately detect emotions in real-life situations.

For example, facial recognition models can identify clear emotions like happiness or anger, but they often fail to detect more subtle or mixed emotions. They also have trouble when facial expressions are unclear or hidden, such as when a person wears a mask or tries to hide their true feelings. Research by Mollahosseini et al. (2017) has shown that even state-of-the-art facial recognition systems can perform poorly on real-world data compared to their results on controlled datasets.

Similarly, speech-based emotion detection systems, which focus on features like pitch and rhythm in a person's voice, may misinterpret emotions in cases like sarcasm or humor. Humans can easily detect sarcasm or a playful tone in speech, but these signals can be hard for machines to understand. Even models trained on large, diverse speech datasets like VoxCeleb or EmoDB struggle to accurately capture emotions in more complex or subtle speech patterns. Studies by Schuller et al. (2013) suggest that voice tone alone can lead to mistakes, especially when emotions are not clearly expressed or are contradictory.

While single-channel emotion detection systems have been effective in certain cases, they still face a major issue: they do not always reflect how humans interpret emotions in the real world. These systems are often trained on large, well-structured datasets, but when they encounter real-world situations with noise, ambiguity, or conflicting signals, they may misclassify emotions or fail to identify them altogether.

These challenges highlight the need for multimodal emotion detection, where data from multiple sources—such as facial expressions, voice tone, and speech content—are combined to create a more accurate understanding of emotions. Multimodal systems are more powerful because they can use different channels of data to complement each other and provide a fuller picture of someone's emotional state. However, even multimodal systems have their own challenges, including handling conflicting signals, missing data, and noisy inputs.

Our project proposes a voting mechanism that combines the results from each channel after they have independently analyzed emotion. This late fusion approach improves the accuracy of emotion detection by considering the strengths of different data sources and resolving conflicts. For example, if facial expressions suggest happiness, but the tone of voice suggests sadness, the system can use the voting mechanism to adjust the final result. By combining the results from multiple sources, our system aims to provide a more reliable and interpretable understanding of emotions, especially in complex or uncertain situations.

Rather than replacing existing multimodal systems, our approach is meant to complement them. In cases where multimodal systems work well, they should continue to be used. However, for situations where emotions are mixed,

conflicting, or hard to interpret, our voting mechanism can provide a clearer, more transparent method for emotion detection.

2.2 Framework

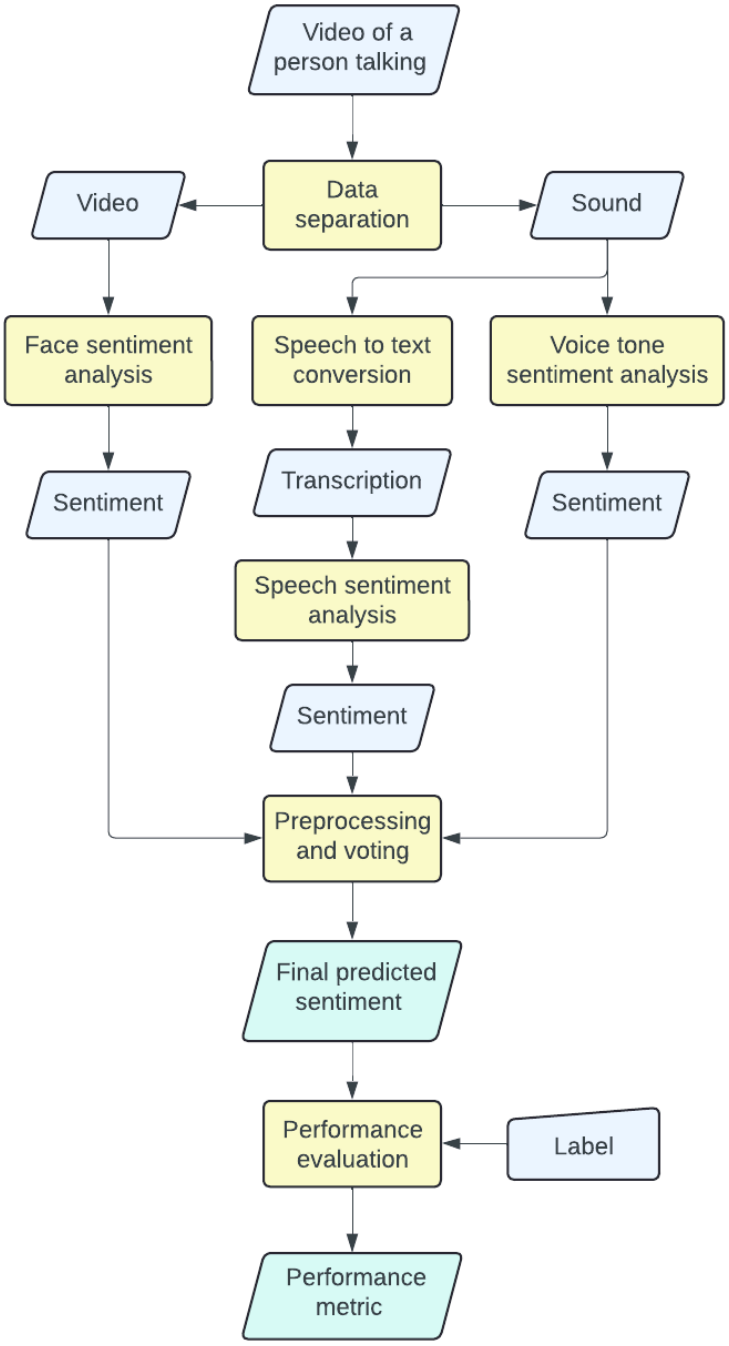


Figure 1: System Framework

This section outlines the steps and the methodology used to extract and analyze sentiment from the video, combining the outputs from each data channel to derive a final sentiment classification. The flowchart above illustrates the workflow, which is explained in detail below.

1. Video of a person talking:

   The input to the system is a short, pre-recorded video file (approximately 30 seconds) containing both visual and auditory data of a single person speaking. The video must maintain consistent communication within a single context, avoiding misclassifications arising from abrupt changes in context or tone. These restrictions are set for the first iteration of our system to ensure that the core functionality is properly established. Future iterations may expand beyond these constraints, allowing for more complex analysis as the system evolves.

2. Data separation:

   The file is separated into two data types: Video which includes the visual representation of the person, particularly their facial expressions and movements, and sound which contains the auditory signals, including both the speech and tone of voice. These separated components are then fed into their respective processing units for sentiment analysis.

3. Face sentiment analysis:

   The video is processed to analyze facial expressions using machine learning models, commonly through face mesh or facial landmark techniques. This step detects micro expressions and overall facial movements to output a sentiment classification: positive, negative, or neutral.

4. Voice tone sentiment analysis:

   The tone of voice is analyzed to identify the sentiment based on vocal features such as pitch, volume, and intonation. This analysis helps in capturing sentiment that may not be explicit in the words themselves but is conveyed through tone. Again, output of this process is a sentiment classification.

5. Speech to text conversion:

   The speech audio is processed through a speech-to-text algorithm, converting the spoken words into a textual transcription. This conversion allows for further semantic analysis, which is necessary for identifying the sentiment expressed in the speech content.

6. Speech sentiment analysis:

   Once the speech is transcribed into text, it is analyzed to identify the sentiment based on linguistic patterns, word choice, and contextual clues in the speech. Again, output of this process is a sentiment classification.

7. Preprocessing and voting:

   This process is the core of our project and will be the focus of development.

   Before the voting mechanism, the system preprocesses each sentiment output from the three models to ensure that they are relevant and properly formatted for our algorithm. This includes tasks such as filtering, feature selection, and data cleaning.

   The voting mechanism will employ a custom algorithm that weighs the contributions of each model, rather than simply using majority rule or mode. A key aspect of this mechanism will involve timestamp matching across channels, ensuring that the sentiment analysis from voice tone aligns with the corresponding transcription timestamps. This complex integration allows for a more nuanced evaluation of sentiment, as it takes into account when specific words were spoken in relation to vocal tone. Additionally, we will synchronize these analyses with facial expression timestamps as well. The output of this process is the final predicted sentiment.

8. Performance evaluation:

   The system’s performance is evaluated using the F1-Score as the key metric, which balances precision and recall to give a comprehensive measure of the system's accuracy. To assess the accuracy, we compare the final predicted sentiment to the actual sentiment label, which is either

manually determined by us reviewing the video or provided from the video source.

This process allows us to reflect on the model's performance and identify any necessary adjustments to the voting algorithm. For instance, if the model underperforms in certain scenarios, we may fine-tune the weights assigned to each model to improve the final sentiment prediction. Additionally, comparing the performance of our multi-channel system to single-modal systems (e.g., using only facial expressions or speech transcription) can help validate the benefits of the multi-channel approach that it indeed offers a more accurate understanding of emotional states.

2.3 Perspective

The interaction between the users and the web application is illustrated in the Figure below

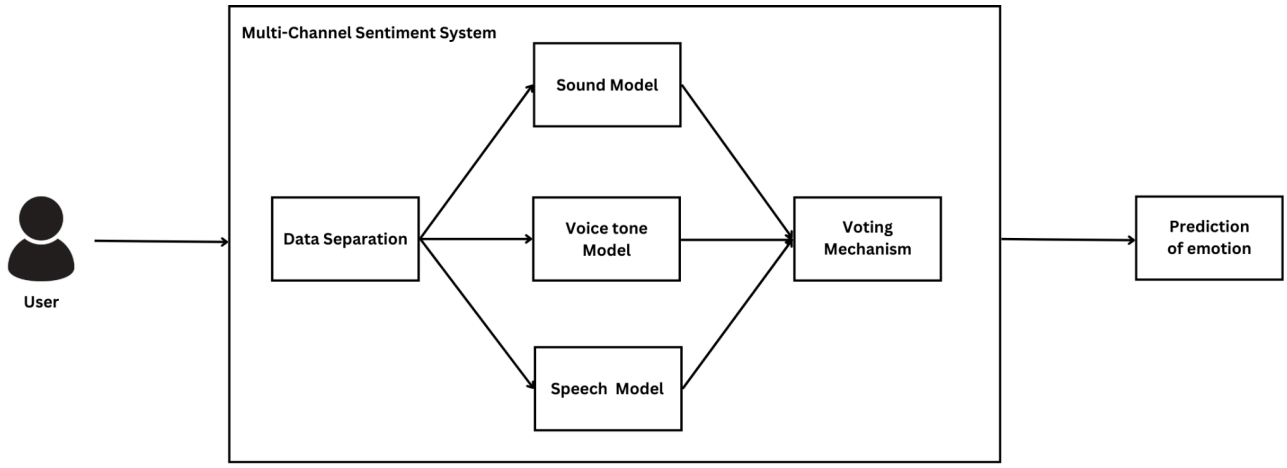


Figure 2: Perspective

From this figure, the system is operated by analyzing a user's live audio and video. The system extracts features from the voice tone, speech transcription, and facial expressions, and then uses different pre-trained machine-learning models to classify the user's overall emotion. After that, the system will weigh each result and vote for the final emotion prediction.

2.4 Requirements

The requirements of the system are listed in the following subsections.

2.4.1 Data Acquisition

DA1 The system should be able to capture and store voice recordings, facial images, and speech transcriptions from users.
DA2 The data acquisition process should be quick, ensuring that data is captured and processed promptly enabling timely emotion.

2.4.2 Data Preprocessing

DP1 The system should be preprocessed to extract relevant features such as pitch, intensity, and spectral characteristics.
DP2 Facial images should be preprocessed to detect and track facial landmarks, enabling the extraction of facial expressions.
DP3 Speech transcriptions should be cleaned and normalized to improve the quality of the input data for sentiment analysis.

2.4.3 Voting Mechanism

VM1 The system should employ a voting mechanism to combine the emotion classification results from the individual modalities.
VM2 The voting mechanism should assign weights to each modality based on their relative importance and reliability in the context of emotion detection.
VM3 The final emotion classification should be determined by our algorithm to weight with a score for each emotion conflict case.

2.4.4 Performance Evaluation

PE1 This system should be evaluated using appropriate metrics such as accuracy, precision, recall, and F1-score.
PE2 The performance of the system should be able to be compared to the existing single-channel emotion classification methods like CNN architecture.

2.5 Preliminary Results

In this section, we present preliminary results from our initial testing of 2 of the sentiment analysis models: transcriptions and voice tone. The primary objective of this analysis was to gauge the effectiveness of using only a single data type in detecting sentiment from a video.

For our testing, we prepared a dataset comprising 5 audio samples from YouTube videos, each approximately 30 seconds long, and with background noise removed using UVR. These audio clips were specifically chosen to represent a range of emotional tones and contexts. The content audio clips content are as follows:

- Penguinz0 expressing anger through a rant about U.S. immigration services (https://youtu.be/8u-_Uh89R9w?si=LrikaIISKohnbPxW)
- Logan Paul apologizing for a uploading controversial a video of a dead body (https://youtu.be/QwZT7T-TXT0?si=pJctHG1uG1dXlaTc)
- Markiplier trying to stay positive while crying grieving a friend's death (https://youtu.be/J_cxoZLPyR0?si=DG4iEZEDs5fC_Q9c)
- MrBallen narrating a love story, objectively (https://youtu.be/CBPYXcxyAOg?si=_pOui3tHBBZF3pRX)
- TommyInnit excitedly announcing his livestream plans (https://www.youtube.com/live/Op0X6a89He8?si=bcLARmFL6bQBTjMo)

Figure 3: Sound Samples

Transcription sentiment

For transcription sentiment analysis, we employed the sentiment_analyzer module from Python's Natural Language Toolkit (NLTK).

```
import speech_recognition as sr
from nltk.sentiment import SentimentIntensityAnalyzer
file_name = "negative (angry) - penguinz0 - sound"
```

Figure 4: Python's Natural Language Toolkit (NLTK)

To convert the speech audio into text, we utilized the Google Speech Recognition API, accessed through the SpeechRecognition library in Python. This API prepares the input necessary for sentiment analysis.

```
def transcribe_audio(file_path):
    # Load audio file
    recognizer = sr.Recognizer()
    with sr.AudioFile(file_path) as source:
        audio_data = recognizer.record(source)

    # Transcribe audio to text
    transcription = recognizer.recognize_google(audio_data)
    return transcription
```

Figure 5: Transcribe Audio into Text

Upon successful transcription of the audio samples, we analyzed the sentiment using the NLTK's SentimentIntensityAnalyzer. This analyzer employs a lexicon-based approach to assess sentiment, providing a score that reflects the positive, negative, and neutral sentiment contained within the transcription. Each audio sample's sentiment scores were recorded for further evaluation. With the compound score value representing an overall sentiment.

```
def analyze_sentiment(text):
    # Perform sentiment analysis
    sia = SentimentIntensityAnalyzer()
    scores = sia.polarity_scores(text)

    # Interpret the sentiment scores
    compound_score = scores['compound']
    if compound_score >= 0.05:
        sentiment = "Positive"
    elif compound_score <= -0.05:
        sentiment = "Negative"
    else:
        sentiment = "Neutral"
    return sentiment, scores
```

Figure 6: Sentiment Analysis

Next, we run the following code for each audio sample.

```
30  # Get the transcription
31  transcription = transcribe_audio(f'sounds/{file_name}.wav')
32
33  # Analyze the sentiment
34  sentiment, scores = analyze_sentiment(transcription)
35
36  # Print the results
37  print(); print(f"Transcription: {transcription}")
38  print(f"Sentiment: {sentiment}")
39  print(f"Scores: {scores}"); print()
```

Figure 7: Sentimental Analysis Each Audio

<u>negative (angry) - penguinz0</u>

Transcription: “it's not possible because there is no reason we would keep being declined they're not even giving us reasons that make any sense at all not to us not to the lawyers not anybody nobody and no other Esports team has had this problem that has international players got in lickety split mean we can't we literally can't”

Sentiment: Negative

Scores: {'neg': 0.15, 'neu': 0.85, 'pos': 0.0, 'compound': -0.8206}

<u>negative (sad) - logan paul</u>

Transcription: “they were unfiltered none of us knew how to react or how to feel I should have never posted the video I should have put the cameras down and stopped recording what we were going through there's a lot of things I should have done differently but I didn't”

Sentiment: Negative

Scores: {'neg': 0.033, 'neu': 0.967, 'pos': 0.0, 'compound': -0.1154}

<u>negative (sad) - markiplier</u>

Transcription: “in in small ways that you probably didn't know Daniel was part of what I did a big part of what I did and Daniel he did a lot of things that were often unseen and I just hope that he knew how much I respected him as a friend”

Sentiment: Positive

Scores: {'neg': 0.0, 'neu': 0.813, 'pos': 0.187, 'compound': 0.8481}

neutral (calm) - mrballen

Transcription: “Mitch was in his final semester at a college in Louisiana when he met another senior a wonderful young lady named Kayla from the instant he saw her he knew he was in love she played hard to get at first but after several months he won her over after graduation the pair stayed in Louisiana and moved in together while they got their careers off the ground two years later the pair got married and almost immediately Kayla got pregnant at the time Mitch's career was really starting to take off which allowed Kayla to stay at home and take some time off”

Sentiment: Positive

Scores: {'neg': 0.011, 'neu': 0.883, 'pos': 0.106, 'compound': 0.8885}

positive (happy) - tommyinnit

Transcription: “but today I've been in a photo shoot all day but I'm very knackered you know but I wore the red and white top and I got on the stream and I said fuck it dude I'm actually like really giddy we're going to hop on Hypixel I have some plans for us that we're going to do you guys some things that I say we need to do”

Sentiment: Positive

Scores: {'neg': 0.045, 'neu': 0.875, 'pos': 0.08, 'compound': 0.3291}

These observations highlight the limitations of using only textual data for sentiment analysis in speech. It struggled with nuanced emotions and complex contexts. Markiplier's somber speech on loss was misclassified as positive, likely due to words like "respect" and "hope" in the transcription even though he’s

crying. Similarly, MrBallen's neutral storytelling was interpreted as strongly positive, misled by terms like "wonderful" and "love." Subtle negative emotions proved challenging to identify accurately. Logan Paul's apology, while correctly labeled negative, received a weak negative score, suggesting difficulty in detecting remorse or regret when not expressed through overtly negative language. The analysis of Tommyinnit's excited announcement, though correctly classified as positive, failed to fully capture the enthusiasm evident in the audio. This highlights the system's inability to account for vocal tone and energy levels.

Voice tone sentiment

For voice tone sentiment analysis, we employed wav2vec2-lg-xlsr-en-speech-emotion-recognition, wav2vec2-large-xlsr-53 models from huggingface. The dataset used to fine-tune the original pre-trained model is the RAVDESS dataset. This dataset provides 1440 samples of recordings from actors performing on 8 different emotions in English, which are: ['angry', 'calm', 'disgust', 'fearful', 'happy', 'neutral', 'sad', 'surprised']. Unlike transcription-based sentiment analysis, which relies on lexical cues and classifies sentiment primarily into positive, negative, and neutral categories, voice tone sentiment analysis identifies a broader range of emotions directly from audio, regardless of spoken content. This allows it to capture the nuances of speech such as pitch, intonation, and energy levels.

```python
from huggingface_hub import login
from transformers import Wav2Vec2ForSequenceClassification, Wav2Vec2FeatureExtractor
import torch
import librosa
import torch.nn.functional as F
file_path = "negative (angry) - penguinz0 - sound.wav"

# Login to HuggingFace
login(token="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")

# Load model
model = Wav2Vec2ForSequenceClassification.from_pretrained(
            "ehcalabres/wav2vec2-lg-xlsr-en-speech-emotion-recognition"
        )
feature_extractor = Wav2Vec2FeatureExtractor.from_pretrained(
                        "facebook/wav2vec2-large-xlsr-53"
                    )
emotions = ['angry', 'calm', 'disgust', 'fearful', 'happy', 'neutral', 'sad', 'surprised']

# Load and preprocess audio
audio_data, sr = librosa.load("sounds/"+file_path, sr=16000)
inputs = feature_extractor(audio_data, sampling_rate=sr, return_tensors="pt", padding=True)

# Run the model and get logits (unnormalized scores)
with torch.no_grad():
    logits = model(**inputs).logits

# Get the predicted class ID
predicted_id = torch.argmax(logits, dim=-1).item()

# Apply softmax to get probabilities
probs = F.softmax(logits, dim=-1)

# Print
print();print(f"Emotion: {emotions[predicted_id]}");print()
```

Figure 8: Sentimental Analysis Voicetone

The models were applied to the same set of audio samples used in transcription sentiment analysis to evaluate the emotional tone conveyed in each speech. Below are the results:

<u>negative (angry) - penguinz0</u>

Detected Emotion Audio Tone: angry

<u>negative (sad) - logan paul</u>

Detected Emotion Audio Tone: disgust

negative (sad) - markiplier

Detected Emotion Audio Tone: happy

neutral (calm) - mrballen

Detected Emotion Audio Tone: surprised

positive (happy) - tommyinnit

Detected Emotion Audio Tone: happy

These observations highlight the limitations of using only voice tone data for sentiment analysis. While the model accurately captured strong emotions like Penguinz0's anger and TommyInnit's happiness, it struggled with more nuanced or layered emotions. Logan Paul's apology was misclassified as disgust rather than sadness, reflecting the model's difficulty in distinguishing remorse from other negative emotions. Markiplier's shaking voice, despite expressing grief, was misclassified as happy somehow, failing to detect the underlying sadness. Similarly, MrBallen's neutral storytelling was incorrectly labeled as surprised, indicating over-interpretation of subtle tonal shifts.

When comparing voice tone and transcription sentiment analysis, both modalities show significant limitations in capturing the full emotional spectrum of speech. While transcription-based analysis misinterpreted emotionally charged words, voice tone struggled with subtle or layered emotions like regret, sadness, or calmness mixed with sorrow. Relying on either modality alone proves insufficient for accurate sentiment detection, especially in real-world contexts where emotions are complex and layered.

Facial data from video can help address these gaps. Facial expressions provide valuable non-verbal cues, such as a furrowed brow or teary eyes, that can signal sadness, anger, or disgust more accurately than voice tone or transcription alone. For instance, in Markiplier's case, the sadness in his teary eyes and facial

expressions would clarify the emotional tone that neither voice nor words could fully convey.

Chapter 3

3.1 Dataset

In our project, we prioritize the use of general videos for testing rather than datasets recorded specifically for training purposes. This decision stems from the need to replicate real-world scenarios where identical facial expressions may correspond to varying emotions, and different emotions may be expressed using the same words but with differing contexts or voice tones.

Standard Individual Image Input

When evaluating models such as **Facetorch**, which is often trained on datasets like FER2013 or FER+, testing on the same or closely related datasets would not yield robust or unbiased results. Therefore, we avoid using FER-related datasets and focus on independent alternatives to assess how well these models generalize to new, unseen data.

Why Avoid FER-Related Datasets?

1. **Bias and Overfitting**:

   Models trained on FER2013 or FER+ perform disproportionately well on similar datasets due to overfitting. Such evaluations inflate results and fail to reflect real-world applicability. The example of test dataset that we use on validate before selecting the model for our project is JEFFE (Japanese Female Facial Expression), which is small non-FER dataset that has same 7 emotions with our universal dataset (FER like).

2. **Unfair Benchmarking**:

   Testing Facetorch on FER-related datasets could give it an unfair advantage over other models like OpenFace, which might not be trained on similar data.

3. **Generalization Capability**:

   Evaluating on diverse, unrelated datasets provides a clearer understanding of the model's adaptability and real-world performance.

Selected Datasets

We chose seven video clips from popular YouTube channels such as

“Penguinz0”, “Logan Paul”, “Markiplier”, “MrBallen”, “TommyInnit”, and “Michael Reeves”. These videos were selected because they show real-life conversations with different facial expressions, voice tones, and emotional contexts. Why These Datasets Were Chosen:

1. **Realistic Emotions:**

   These videos show real, unscripted emotions in everyday situations. This is important because we want to see how well the model works in real life, not just in controlled or fake settings.

2. **Mixed Emotional Cues:**

   Emotions are complex and can come from not just facial expressions but also what’s said and how it’s said. These videos let us test the model in situations where emotions need to be understood from all these different clues.

3. **Different Types of Emotions:**

   The videos show a wide range of emotions, such as happiness, surprise, frustration, sarcasm, and excitement. This helps us see how well the model can handle different emotions.

4. **Variety in Personalities and Content:**

   The selected YouTubers each have their own style and personality, which means the emotions in their videos are very different. For example, Markiplier’s videos are often funny with lots of exaggerated emotions, while MrBallen’s videos can be more serious and suspenseful. This variety helps test how well the model can handle different kinds of emotions.

By using these real-world, natural videos, we can better understand how well the model works in everyday conversations, where emotions are not always easy to see or categorize.

3.2 Preprocessing

Since we can’t feed raw video into each model, we need to preprocess it into another suitible file first before input into modalities

Preprocessing for Facial Expression Modality

For the facial expression analysis, the raw video file in .mp4 format needs to be processed into individual frames. The facial expression model relies on detecting emotions from still images, so we begin by extracting a frame from the video every second. This means that for each second of the video, a corresponding .jpg image will be saved. Each image represents a snapshot of the person's face, allowing the model to analyze and recognize facial emotions at that specific moment in the video. By setting the frame extraction rate to **one frame per second**, we ensure that each second of the video is represented by a still image, making it easier to analyze emotions that correspond to each moment. The resulting images are named sequentially (e.g., frame_0001.jpg, frame_0002.jpg, etc.) and stored for further analysis. These images will later be fed into the facial expression model for emotion recognition.

Preprocessing for Transcript and Voice Tone Modality

Both transcription and voice tone analysis rely on extracting the audio from the video, as they are both focused on understanding emotions conveyed through speech. The first step is to extract the audio from the .mp4 video file and save it as a .wav file. This format ensures high-quality, uncompressed audio that is ideal for speech processing.

3.3 Model

3.3.1 Facial Expression

Overview

Facial expression analysis is a core modality in our multi-channel sentiment analysis framework. This channel captures visual emotional cues, integrating two state-of-the-art models: **Facetorch** and **OpenFace**. These models were chosen to address different strengths: Facetorch excels in static frame analysis, while OpenFace is suited for dynamic, temporal video analysis.

Model Selection and Strategy

The key criteria for selecting facial expression models were:

- Their suitability for emotion recognition tasks, focusing on subtle

expressions and temporal analysis.

- Availability of pre-trained models to reduce development time and simplify integration.
- Balancing strengths in both static and dynamic data handling.

At first, we consider using the DeepFace model, a deep learning model developed by Facebook AI Research with high accuracy in identifying individuals from large-scale datasets, varying lighting conditions, poses, and facial expressions. However, DeepFace's primary focus is on identifying individual identities rather than recognizing and classifying emotions. As a result, it may not be optimally suited for tasks involving subtle emotional cues, such as micro-expressions or nuanced facial movements.

In the end, after we explored more models and based on the criteria above, we came across these options:

1. **Facetorch (FER-like)**: A Python library designed for face analysis trained on various FER-like datasets but not explicitly mentioned if it is FER+ and FER2013 in Facetorch documentation. We use Facetorch as the baseline model for our facial expression analysis due to its capability to identify emotions like happiness, sadness, anger, surprise, fear, disgust, and neutral expressions. This model is well-suited for tasks where static expressions are the focus, such as identifying facial expressions from individual photos or isolated video frames.

   Although It does well at detecting emotions in single images, it might have trouble with videos that contain subtle temporal changes.

   Implementation: Download “facetorch” library to access pre-trained models: EfficientNet-B0 and B2 then use the library's face detection and emotion classification functions on individual video frames.

2. **OpenFace:** A framework with advanced features for dynamic expression analysis in videos and facial landmark tracking. Which have Action Units (AUs) that OpenFace generates to represent specific muscle movements and temporal information on emotional transitions.

Implementation: Installing necessary dependencies like “dlib” and “OpenCV”. Then, preprocess the video data for face detection and pass it to OpenFace functionalities for landmark tracking, AU extraction, and emotion classification across video frames.

Limitations of Single-Channel Models in Facial Expression Recognition

While many single-channel models for facial expression recognition (FER) are trained on state-of-the-art datasets, such as “FER+”, “FER2013”, or other reputable emotion datasets, these models still face inherent limitations when applied to real-world scenarios. These limitations are particularly evident when analyzing single frames of facial expressions, which may lack the temporal context needed to fully understand the underlying emotion. Even with the best datasets and highly reputable models, such as “Facetorch”, “DeepFace”, or “AffectNet”, there are several reasons why single-frame models are not perfect at detecting emotions accurately.

1. **Lack of Temporal Context**

   A key issue with single-frame models is that they often treat each image as an isolated snapshot, without considering the temporal evolution of emotions. Human emotions, especially in dynamic contexts, are rarely captured in one frame. For example, the transition from a neutral face to a smile, or from a surprised expression to a fearful one, involves subtle changes in facial muscles over time. Single-frame models, even those trained on large, state-of-the-art datasets, may fail to capture these transitions, leading to misclassification or a lack of nuance in the emotion detection.

2. **Ambiguity in Expression**

   Emotions are not always easily discernible from a single frame of a face. For instance, someone may express mixed emotions, like being slightly surprised while also showing signs of discomfort, which is hard to capture in a single image. Models trained on large datasets may still struggle in these situations, as they typically focus on detecting one primary emotion per frame, which can lead to incorrect conclusions when multiple emotions are present

simultaneously or when the expression is too subtle for the model to pick up.

3. **Model Overfitting to Dataset Bias**

   Even state-of-the-art models trained on highly diverse datasets can become biased by the particular types of faces, lighting conditions, and backgrounds present in their training data. In real-world scenarios, human faces can present varied expressions under diverse conditions (lighting, angle, motion blur, etc.), and these variations may not always be well-represented in the training set, causing a gap in model performance when tested on new, unseen data. Despite advances in datasets, models trained on such data might still fail to detect emotions accurately in diverse real-world environments.

Given these challenges, it becomes clear that relying solely on single-frame analysis may not always yield reliable or accurate results in emotion detection.

Figure 9 : Angry penguinz0 face

The image above illustrates a significant limitation of single-frame emotion detection models. Even when the subject, such as the YouTuber Penguinz0, is clearly expressing “anger” (shouting in frustration about a situation like US

immigration policy regarding his esports team), the model may misclassify the emotion as “surprise” due to certain facial features:

- **Eyes nearly closed**: Often associated with surprise when accompanied by wide-open eyes in some contexts, but this image shows eyes squinting, a possible sign of anger.
- **Mouth wide open**: This is commonly labeled as surprise by single-frame models because of its association with a sudden reaction or shock.

Without considering context or the video's temporal dynamics, the emotion recognition model lacks the ability to accurately determine the intent behind the facial expression. This reinforces the need for temporal data analysis (e.g., OpenFace or video-segment annotation), which could account for how the expression evolved, audio cues, and the subject's tone of voice to classify it correctly as anger.

Workflow for Facial Expression Analysis:

Facial expression analysis is integral to multimodal emotion detection, where temporal dynamics and robust mapping to universal emotion categories are essential. This section outlines the methodologies for two key approaches: **FaceTorch** and **OpenFace**, highlighting their integration and interoperability within our project.

A. FaceTorch Workflow

FaceTorch is primarily designed for static frame analysis, excelling in tasks where individual images are used for emotion recognition. However, its limitation lies in the lack of temporal dynamics, requiring preprocessing to align with video-based datasets. Steps in the FaceTorch Workflow:

1. **Frame Extraction:**
    - Video data is preprocessed by extracting frames at fixed 1-second intervals to match the temporal annotation granularity of FERPlus.
2. **Emotion Recognition:**

- Each frame is passed through the pre-trained FaceTorch model (EfficientNet variants).
- The output includes emotion predictions and confidence scores.

3. **Temporal Alignment:**
    - Frame-level results are aggregated into 1-second intervals for compatibility with FERPlus annotations and evaluation.

Strengths:

- High accuracy for static emotion recognition.
- Easily adaptable for image datasets.

Limitations:

- Lack of temporal awareness restricts dynamic emotion analysis.

B. OpenFace Workflow

OpenFace introduces temporal dynamics into emotion analysis through **Action Units (AUs)** and facial landmark tracking, making it particularly suited for video-based datasets. Steps in the OpenFace Workflow:

1. **Feature Extraction:**
    - OpenFace's FeatureExtraction module processes video frames, outputting a CSV file containing:
        - AU intensities (e.g., AU06_r, AU12_r).
        - Confidence scores.
        - Frame metadata (timestamps, landmarks).
    - **Example Command:**
        - ./FeatureExtraction -f video.mp4 -out_dir output/
2. **Temporal Aggregation:**

- Frame-level outputs are grouped into 1-second intervals to align with FERPlus annotations.
- For each interval:
    - The most frequent predicted emotion is selected.
    - Confidence scores are averaged.

3. **Emotion Mapping:**
    - AUs are mapped to universal emotion categories (Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral) using predefined heuristics.
4. **Accuracy Evaluation:**
    - Predictions are compared against ground truth labels from the FERPlus-like JSON format.
    - Results include:
        - Image path.
        - Ground truth emotion.
        - Predicted emotion.
        - Confidence score.
        - Accuracy flag (True/False).

Example CSV Output:

Image	Ground Truth	Prediction	Confidence	Correct
frame_0000.jpg	Neutral	Sad	0.98	False

Strengths:

- Dynamic emotion analysis through AUs.
- Temporal consistency for video-based datasets.

Limitations:

- Requires preprocessing for universal mapping compatibility.

Comparison and Integration

Both FaceTorch and OpenFace offer unique strengths, complementing each other in a multimodal system:

Feature	FaceTorch	OpenFace
Input Type	Static Frames (Images)	Dynamic Frames (Videos)
Emotion Granularity	Single Frame	Temporal Dynamics
Preprocessing Requirement	High (Frame Extraction)	Moderate (CSV Parsing)
Mapping to Universal Emotion List	Direct	Rule-based Heuristics
Synchronization	Interval-based Alignment	Interval-based Aggregation

By integrating the static accuracy of FaceTorch with the temporal dynamics of OpenFace, the workflow achieves a robust and synchronized emotion analysis pipeline, suitable for multimodal voting systems.

3.3.2 Voice Transcription

For the transcription modality, we have selected the Hugging Face Emotion Detection model, specifically the DistilRoBERTa model fine-tuned for emotion classification. This model was chosen after evaluating several options, and it emerged as the best choice for our project based on several key factors. Reasons for Choosing This Model:

1. **State-of-the-Art Preprocessing and Architecture:**

The model is built on DistilRoBERTa, a transformer-based architecture derived from RoBERTa, which is a robust pre-trained model that has been fine-tuned for emotion detection. DistilRoBERTa is a distilled version of RoBERTa, making it smaller, faster, and more efficient, while still retaining much of RoBERTa's performance. This architecture is known for its ability to handle complex language understanding tasks, such as emotion detection, by capturing subtle nuances in text, making it particularly well-suited for our needs.

2. **Fine-Tuned for Emotion Detection:**

   One of the most important factors in our selection process was the model's fine-tuning on emotion-labeled datasets. The emotion-english-distilroberta-base model was trained specifically to detect emotions in English text, including emotions like happiness, sadness, anger, surprise, etc. This makes it directly applicable to our task of detecting the emotions in video transcripts. The model's ability to recognize and classify emotions based on context and wording aligns perfectly with the needs of our project, where emotions are often conveyed through subtle shifts in speech.

3. **Accuracy and Reliability:**

   During our testing phase, we experimented with several emotion detection models from Hugging Face, and the DistilRoBERTa model outperformed others in terms of both accuracy and consistency. This model exhibited the most reliable results in predicting emotions from text, showing a good balance between precision and recall across multiple emotional categories. Its ability to generalize well across different types of text and contexts made it stand out from other models that struggled with ambiguous or subtle emotional expressions.

4. **Lightweight and Efficient:**

   Since our project involves processing large video datasets, model efficiency is crucial. DistilRoBERTa offers a significant reduction in model size compared to other transformer models (like the original RoBERTa), making

it faster and less computationally expensive while maintaining a high level of performance. This efficiency is especially important when processing large volumes of text from video transcripts, allowing us to process data in a timely manner without sacrificing accuracy.

3.3.3 Voice Tone

For the voice tone modality, we have selected the **Hugging Face Wav2Vec 2.0 Emotion Recognition model**. This model was chosen after testing and evaluation against other voice emotion recognition models. It proved to be the most suitable for our needs, offering high accuracy, reliability, and efficiency in detecting emotions from speech. Reasons for Choosing This Model

1. **Advanced Audio Processing Architecture**: The model is built on **Wav2Vec 2.0**, a state-of-the-art architecture for processing raw audio waveforms. Wav2Vec 2.0 is particularly effective at learning speech representations directly from audio data without requiring extensive preprocessing or feature extraction. Its ability to handle raw audio input makes it highly effective for emotion detection tasks, as it can capture subtle variations in voice tone, pitch, and intensity that are essential for determining emotional states.

2. **Fine-Tuned for Emotion Detection**: This model has been fine-tuned specifically for emotion recognition from speech, leveraging emotion-labeled datasets to understand the nuances of human emotions conveyed through voice. The fine-tuning process ensures that the model can detect a wide range of emotions, including happiness, sadness, anger, and fear, with a high degree of accuracy. Its specialization for this task makes it an ideal choice for analyzing voice tone in our project.

3. **High Accuracy in Testing**: During our testing phase, the **wav2vec2-emotion-recognition** model consistently outperformed other models in recognizing emotions from voice data. It exhibited strong performance in detecting both explicit emotions (e.g., anger, joy) and more subtle emotional cues (e.g., confusion, surprise) across a variety of audio samples. Its precision in classifying emotions, even in challenging cases like

overlapping or noisy audio, made it a standout choice.

3.4 Syncing Result

The syncing process is needed to combine the data from three different modalities—facial expressions, voice tone, and transcripts into coherent partitions. These partitions represent consistent time intervals within the video data, allowing the voting mechanism to make emotional predictions for each period of the video.

3.4.1 Input Data

1. **Facial Expressions**
    - Source: “facial_expression/FERPlus/data/output/file_name”
    - Format: Each line contains the frame number, detected emotion, and the model’s confidence score.
    - Example: frame_00001.jpg, Happy, 0.85
        - Frame: frame_00001.jpg (used to infer timing)
        - Emotion: Happy
        - Confidence: 0.85
2. **Voice Tone**
    - Source: “voice_tone/file_name”
    - Format: JSON array where each entry contains the time, detected emotion, and confidence.
    - Example: { "time": 1.5, "emotion": "Angry", "confidence": 0.75 }
3. **Transcription**
    - Source: “voice_transcription/file_name”
    - Format: JSON array where each entry represents a sentence with its timestamp, emotion, and confidence.

- Example: { "time": 2.0, "transcript": "I can't believe this!", "emotion": "Disgust", "confidence": 0.90 }

3.4.2 Partitioning Process

The syncing process is based on the timestamps in the transcripts. Each transcript defines a partition that spans from the end of the previous sentence to the timestamp of the current sentence. The syncing steps are as below:

1. **Initialize Partitioning**
   - Start time for the first partition is 0.0 seconds.
2. **Iterate Over Transcript Entries**
   - Each transcript entry defines the end time of the current partition.
   - The start time of the partition is the end time of the previous entry.
3. **Populate Modalities into Partitions**
   - **Facial Expressions:**
     - Match the frame times to the current partition based on the frame number.
     - Frames falling within the start and end times are added to the partition.
   - **Voice Tones:**
     - Match the timestamps of voice tone entries to the current partition.
     - Entries within the start and end times are added to the partition.
4. **Save Partition**
   - Each partition includes:
     - **Transcript:** The text, detected emotion, and confidence.
     - **Facial Expressions:** Frames, detected emotions, and confidences for all frames within the time window.
     - **Voice Tone:** Detected emotions and confidences within the time window.
5. **Update Start Time**
   - The end time of the current partition becomes the start time for the next.

3.4.3 Output

The result is a structured JSON file (output.json) containing synchronized data for each partition. Example output:


```
[
  {
    "partition": "0-6s",
    "transcript": {
      "text": " So what we came across that day in the woods was obviously
      unplanned and the reactions you saw on tape were raw. They were
      unfiltered",
      "emotion": "Neutral",
      "confidence": 0.40619438886642456
    },
    "facial_expression": [
      {
        "frame": "0000.jpg",
        "emotion": "Sad",
        "confidence": 0.19
      },
      …
    ],
    "voice_tone": [
      {
        "time": 5,
        "emotion": "Fear",
        "confidence": 0.5330254435539246
      }
    ]
  },
  …
]
```


3.5 Voting Mechanism

3.5.1 Key Concepts

3.5.1.a Weight for each Model

The code assigns a weight to each modality, representing its importance in

the final decision:

- Transcript (text or speech content): 50% weight (0.5)
- Facial Expression: 30% weight (0.3)
- Voice Tone: 20% weight (0.2)

These weights show how much influence each modality has when determining the final emotion. The total weight is calculated dynamically based on the available data for each partition.

3.5.1.b Reason

The **transcript** is assigned the highest weight of **0.5**, and here's why:

- **Single Emotion per Sentence**:

  In each partition, the transcript corresponds to a single sentence or a segment of speech. This allows for a direct mapping between the sentence's meaning and the emotion expressed. Since each sentence typically conveys a single emotion, it's easier to pinpoint and confidently determine the sentiment based on the words and their context.

- **Context and Meaning**:

  Emotions in language are often tied to the context of the conversation. The words used, the sentence structure, and the overall meaning can give a clearer indication of the emotional state. For example, the sentence "I am so happy to see you!" clearly expresses a positive emotion of joy, and the context in which it's said helps to eliminate ambiguity. Therefore, the transcript is highly reliable for detecting emotions, especially since it often provides strong clues in context.

- **Ease of Determining Emotion**:

  Compared to facial expressions and voice tone, it is easier to determine emotion from text because the meaning of words and the context they provide are directly linked to emotional expression. Sentences with clear

emotional cues are easier to analyze and classify with confidence, making the transcript a strong input for emotion detection.

For these reasons, the **transcript** is given a higher weight of **0.5** because it provides a reliable, clear, and context-rich signal for determining emotion, especially when there's no ambiguity in the words used.

The **facial expression** modality is assigned a weight of **0.3**, which is the second-highest weight, and here's the rationale for this choice:

- **More Reliable Than Voice Tone**:

  While voice tone can sometimes be ambiguous or difficult to interpret accurately, facial expressions are generally more stable and universally understood across cultures. People's faces provide strong emotional cues through movements of muscles and facial features, such as smiles, frowns, raised eyebrows, and other subtle expressions that signal emotions like happiness, anger, sadness, or surprise.

- **Many Models Developed for Facial Expression Recognition**:

  There is a vast body of research and many advanced models designed specifically to analyze facial expressions. These models tend to be quite reliable and have been trained on large datasets to detect emotions from facial features. This makes facial expression a strong predictor of emotion, especially in real-world scenarios where people's faces tend to show clear emotional cues.

- **Tendency for Multiple Results**:

  One challenge with facial expression analysis is that it can sometimes yield multiple results for a given segment, especially if the person is displaying mixed or subtle emotions. For example, a person might look frustrated while smiling, creating conflicting signals. Because of this, facial expressions might not always be as clear-cut as the transcript in determining a single emotion. This variability in results is one reason facial expression is given a slightly lower weight than the transcript.

Despite these challenges, **facial expression** remains an important and reliable source of emotional data, which is why it is given a weight of **0.3**.

The **voice tone** modality is given the lowest weight of **0.2**. Here's why:

- **More Variability and Ambiguity**:

  Voice tone analysis can be more challenging because the same words can be spoken in various tones, which can significantly change the emotion conveyed. For instance, a sarcastic tone might make a sentence that would normally be happy sound angry or confused. On the other hand, a monotone voice might make a sentence sound emotionally neutral, even if the words suggest a particular emotion.

- **Less Developed Models**:

  Although there are advanced models for detecting emotions from voice tone, they are generally less robust than facial expression models. Detecting emotions from voice tone is more complex because it involves analyzing pitch, speed, tone, and rhythm. Voice tone can sometimes be harder to interpret in isolation, and various environmental factors (e.g., background noise, and microphone quality) can interfere with the accuracy of emotion recognition.

- **Greater Risk of Outlier Predictions**:

  Voice tone tends to be more affected by outlier predictions. For example, if a person speaks in an unusual tone or is speaking under stress, the voice tone recognition model might incorrectly identify an emotion, leading to a higher chance of errors. Since voice tone can be more easily misinterpreted than facial expressions or text, it is assigned the smallest weight of **0.2**.

However, voice tone is still an important modality because it can provide additional clues, especially when combined with facial expressions and transcript data. It can help detect emotions like anger, sarcasm, or sadness that may not be as clearly conveyed through text or facial expressions alone.

3.5.1.c Emotional Categories

In real-life scenarios, people do not typically transition abruptly between extreme emotions, such as being happy and sad, within the span of a single sentence. Emotional states are usually more stable and follow a natural flow over time. When conflicting emotions are detected within the same partition, it raises a red flag that the detected conflict might not represent genuine human emotional behavior. Instead, it may indicate **noise, ambiguity, or outliers** in one or more modalities. To counter that we categorize result emotion into these 3 types:

- Negative Emotions: ["Angry", "Disgust", "Fear", "Sad"]
- Positive Emotions: ["Happy", "Surprise"]
- Neutral Emotion: "Neutral" (does not contribute to conflict)

With this categorization, a conflict that occurs when both positive and negative emotions are present in the list of emotions for a partition will be easier to determine. This leads to more manageable weight adjusting through the proportion of conflicting emotions (positive vs. negative). Higher conflict results in greater weight reduction.

3.5.2 Steps of the Algorithm for Adjusting Weight

1. **Count Positive and Negative Emotions:**
    - Iterate through the list of emotions for the given modality (facial expression or voice tone).
    - Count how many are classified as positive and how many as negative.
2. **Check for Conflict:**
    - If both positive and negative counts are greater than zero, a conflict exists.
3. **Compute Conflict Ratio:**
    - Calculate the ratio of the smaller group (either positive or negative)

to the total count of positive and negative emotions:

$$conflict\_ratio = \frac{min(positive\_count, negative\_count)}{positive\_count + negative\_count}$$

- The higher the conflict ratio, the greater the level of emotional conflict.

4. **Adjust Weight:**

   - The weight is reduced based on the conflict ratio:

$$adjusted\_weight = 1 - \text{conflict\_ratio}$$

5. **ApplyAdjust Weight:**

   - Multiply the adjusted weight by the predefined weight for that modality (facial expression or voice tone) when aggregating emotions.

3.5.3 Example for Adjusting Weight

Emotion List for Voice Tone:

[{"emotion": "Happy", "confidence": 0.7},

{"emotion": "Sad", "confidence": 0.6},

{"emotion": "Fear", "confidence": 0.8}]

Modality Weight: 0.2

Step-by-Step:

1. **Count Emotions:**

   - Positive emotions: 1 (Happy)

   - Negative emotions: 2 (Sad, Fear)

2. **Check Conflict:**

   - Both positive and negative counts > 0, so a conflict exists.

3. **Compute Conflict Ratio:**

$$conflict\_ratio = \frac{min(1,2)}{1+2} = \frac{1}{3} \approx 0.33$$

4. **Adjust Weight:**

$$adjusted\_weight = 1 - 0.33 \approx 0.67$$

5. **Apply Adjust Weight:**

   - Original weight: 0.2
   - Adjusted weight: 0.2 * 0.67 ≈ 0.134

Thus, the weight for voice tone is reduced to 0.134 for this partition due to the emotional conflict.

After all the results are calculated for each modality, the scores are normalized by dividing by the total weight. This ensures that the final scores are comparable across different partitions, regardless of how much data is available for each modality.

Once the scores are normalized, the emotion with the highest score is selected as the final emotion for that partition. The corresponding confidence score (how confident the system is in the selected emotion) is also recorded.

Finally, the results are printed, showing the final emotion and confidence for each partition. For further result after voting can be found in Appendix.

3.5.4 Inter-chunk Outlier Handler

This function smooths out transitions between different emotions in a list of partitioned results. It ensures that short, transient emotional states are either merged into the previous emotion or replaced by a "Neutral" state, providing more realistic emotion in the sequence, since human emotion does not normally change from positive to negative and back to positive immediately.

Step-by-Step:

1. **Initialization**:

   - The function initializes an empty list refined_results to store the final results.

- current_emotion is set to None to start tracking the first emotion in the sequence.
- current_streak is an empty list used to accumulate results with the same emotion.

2. **Processing Each Partition Result**:
    - The function iterates over each result in the partition_results list.
    - For each result, the emotion (final emotion) is checked.
3. **Handling "Neutral" Emotions**:
    - If the emotion is "Neutral":
        - If there is an ongoing streak, it is finalized by adding the streak results to refined_results.
        - The current streak is reset to start fresh, and the "Neutral" result is appended to refined_results as it is.
        - This ensures that "Neutral" emotions are only inserted after completing any previous streak of the same emotion.
4. **Handling Valid Emotions (Non-Neutral)**:
    - If the emotion is not "Neutral", the function checks if the current emotion matches the ongoing streak (current_emotion).
    - If the emotion is the same as the current one, it is added to the current streak.
    - If the emotion is different from the current one:
        - The function finalizes the previous streak:
            - If the streak is shorter than the stability threshold, each result in the streak is changed to "Neutral".
            - The results of the finalized streak are added to refined_results.

- The function starts a new streak with the current emotion and adds the current result to this new streak.

5. **Finalizing the Last Streak**:

   - After the loop finishes processing all results, the function ensures that the final streak is processed.
   - Similar to previous streaks, if the final streak is shorter than the stability threshold, the results are changed to "Neutral". If it is long enough, the streak remains intact.
   - This final streak is then added to refined_results.

The output is a list of refined emotion results where short transitions are replaced by "Neutral"

3.5.5 Example for Inter-chunk Outlier Handling

Output:

- Partition: 0-6s, Final Emotion: Sad, Confidence: 0.37
- Partition: 6-10s, Final Emotion: Sad, Confidence: 0.63
- Partition: 10-12s, Final Emotion: Happy, Confidence: 0.40
- Partition: 12-17s, Final Emotion: Fear, Confidence: 0.46
- Partition: 17-21s, Final Emotion: Neutral, Confidence: 0.36
- Partition: 21-24s, Final Emotion: Neutral, Confidence: 0.23

After Adjust:

- Partition: 0-6s, Final Emotion: Sad, Confidence: 0.37
- Partition: 6-10s, Final Emotion: Sad, Confidence: 0.63
- Partition: 10-12s, Final Emotion: Neutral, Confidence: 0.40
- Partition: 12-17s, Final Emotion: Fear, Confidence: 0.46

- Partition: 17-21s, Final Emotion: Neutral, Confidence: 0.36
- Partition: 21-24s, Final Emotion: Neutral, Confidence: 0.23

Chapter 4

4.1 Model Evaluation

This section provides a detailed evaluation of two facial expression recognition models: **Facetorch** (Model v1 and v2) and **OpenFace**, tested on both the JEFFE dataset and a prepared video dataset. The analysis compares their accuracy and highlights strengths, weaknesses, and performance trends.

4.1.1 Facetorch Evaluation

4.1.1.a JEFFE Dataset Results

Facetorch Models v1 and v2 were evaluated using the JEFFE dataset, a static dataset with labeled images for 7 universal emotions: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral.

Results:

Model	**Validation Accuracy**
Model v1	21.13%
Model v2	1.41%

Observations:

- Model v1 performed significantly better than Model v2 on the JEFFE dataset.
- Model v2 accuracy is close to zero, suggesting possible model underfitting or improper weight initialization.

4.1.1.b Prepared Video Dataset Results

Facetorch (Model v1 and v2) was tested on a prepared dataset consisting of video frames extracted from different video clips.

Results:

Video	Model v1 Accuracy	Model v2 Accuracy
penguinz0	21.43%	10.71%
michael_reeves	30.00%	0.00%
tommyinnit	47.37%	10.53%
mrballen	13.33%	0.00%
markiplier_part1	31.25%	0.00%
markiplier_part2	3.12%	0.00%
logan_paul	34.62%	65.38%

Observations:

- **Model v1** outperformed Model v2 in most video cases, achieving accuracy between **21.43% and 47.37%**.
- **Model v2** showed inconsistent performance, achieving **0.00% accuracy** in several videos. However, it performed better on the "logan_paul" video with **65.38% accuracy**.

Insights:

- Model v1 is more stable across different videos but still underperforms on videos with subtle or ambiguous emotions.
- Model v2 may be overfitting to specific data but failing to generalize.

4.1.2 OpenFace Evaluation

OpenFace, a pre-trained model, was evaluated on the prepared video dataset to analyze its performance on dynamic emotional recognition.

Results:

Video	OpenFace Accuracy
logan_paul	34.62%
markiplier_part1	43.75%
markiplier_part2	9.38%
michael_reeves	70.00%
mrballen	96.67%
penguinz0	21.43%
tommyinnit	36.84%

Observations:

- OpenFace achieved superior accuracy compared to Facetorch, especially on videos like “mrballen” (**96.67%**) and “michael_reeves” (**70.00%**).
- Performance dropped for videos with ambiguous expressions or complex dynamics, such as “markiplier_part2” (**9.38%**).

Insights:

- OpenFace is more robust for analyzing emotional dynamics in videos.
- Its pre-trained features generalize well, particularly on clear facial expressions.

4.1.3 Comparative Analysis

Aspect	Facetorch (Model v1)	Facetorch (Model v2)	OpenFace
Input Type	Static images	Static images	Dynamic video frames
JEFFE Dataset Accuracy	21.13%	1.41%	N/A

Prepared Dataset Accuracy	21-47%	0-65%	9-96%
Performance Consistency	Moderate	Inconsistent	High
Strengths	Handles static images	N/A	Robust for videos
Weaknesses	No temporal analysis	Poor generalization	Requires high compute

Key Insights

1. **Facetorch**:
    - Model v1 performs better than Model v2 but struggles with subtle emotions.
    - Model v2 shows poor consistency, with notable improvement only in the logan_paul video.
    - FaceTorc performs well on individual static frames, offering a simpler setup and high accuracy on clear, static emotions.
2. **OpenFace**:
    - Outperforms Facetorch across most videos, particularly in clear, dynamic emotional transitions.
    - Its ability to analyze facial dynamics makes it superior for real-world use cases.
    - OpenFace excels in dynamic analysis by tracking temporal facial muscle movements and action units (AUs) over time, making it ideal for detecting subtle transitions and micro-expressions.
3. **General Trends**:
    - OpenFace is better suited for **dynamic emotion recognition** in videos.

- Facetorch is more suitable for static, image-based datasets.

Real-World Applicability:

- Real-world emotions are rarely straightforward. People may hide their feelings, pause to take a breath mid-anger, or express subtle micro-expressions before fully revealing their emotions. OpenFace's ability to detect muscle movement transitions provides an advantage over static-frame models like FaceTorch.
- Conversely, FaceTorch's ability to process static images quickly is beneficial when dynamic tracking is unnecessary or computational resources are limited.

Loosely Coupled Voting Mechanism

The decision to implement a **voting mechanism** rather than relying on a single model or a tightly integrated multimodal system proves advantageous. This approach enables the system to dynamically switch between models depending on the use case or confidence scores.

By combining outputs from multiple models, the system can address edge cases, such as conflicting signals, ambiguous expressions, or noisy inputs, which state-of-the-art multimodal methods often struggle with.

Here's how the system handled the switch seamlessly:

1. **Fixed Interval Aggregation**:
    - Both models provided outputs for specific timestamps or frames. Regardless of their processing differences, the system aggregated or aligned their results into fixed 1-second intervals.
    - **Post-Processing Flexibility**: Even though OpenFace requires temporal analysis, and Facetorch processes each frame independently, the system normalized their outputs into the same format (emotion categories and confidence scores). This ensured compatibility with the voting mechanism.
2. **Result Synchronization**:

- The universal format allowed the system to align outputs from the facial expression channel with the other modalities (voice tone and transcription). This alignment was achieved without any dependency on the underlying model architecture.
- **Outcome Consistency**: Both Facetorch and OpenFace delivered comparable results in terms of the emotional classification required for the syncing mechanism. This highlights the modularity and robustness of the framework.

4.2 Addressing State-of-the-Art Limitations

Accuracy vs. Real-World Understanding

- Current state-of-the-art systems prioritize accuracy on benchmark datasets but struggle with real-world complexities, such as sarcasm, mixed emotions, or ambiguous expressions.

- The voting mechanism prioritizes interpretability and conflict resolution, making it more applicable to nuanced scenarios where human emotions are layered and complex.

Challenges in Early Fusion Multimodal Systems

- Multimodal systems often combine data from different channels too early, leading to issues when one channel's data is missing or noisy.

- The voting mechanism, by contrast, ensures that each channel contributes independently to the final decision, reducing the impact of incomplete or noisy inputs.

4.3 Advantages of the Loosely Coupled Voting Mechanism

4.3.1 Feasibility and Practical Applicability

Traditional multimodal systems often focus on a limited number of channels (e.g., text and voice tone or text and facial expressions). Research shows that most multimodal frameworks are not designed to handle more than two channels effectively. Extending these systems to integrate three or more channels (as in this

project: facial expression, voice tone, and transcription) requires significant re-engineering, which is resource-intensive and technologically challenging.

- **Advantage**: The voting mechanism bypasses this limitation by allowing each channel to be processed independently before combining their results through post-processing. This makes the approach more feasible and scalable for real-world applications.

- **Application**: Emotional analysis in mental health monitoring or customer service, where multi-channel integration is critical but constrained by computational and budgetary limitations.

4.3.2 Accessibility of Black-Box Models

Many state-of-the-art models in facial recognition, voice tone analysis, and natural language processing are proprietary (API-only services) or deeply integrated within a closed system. Accessing the internal weights or architecture for fine-tuning is often impossible without explicit permissions.

- **Advantage**: By treating these models as black boxes and only utilizing their outputs, the voting mechanism avoids the need for invasive model integration. This allows researchers and practitioners to use cutting-edge tools without violating licensing restrictions or investing in costly development.

- **Application**: Integration of industry-grade tools like Google Cloud's NLP API or Amazon Rekognition for emotion detection without needing to re-train or modify these systems.

4.3.3 Flexibility and Modularity

The sentiment analysis industry is highly dynamic, with frequent advancements in state-of-the-art models. Multimodal systems are tightly coupled, meaning that replacing one channel's model often requires re-training the entire framework, which can be expensive and time-consuming.

- **Advantage**: The loosely coupled nature of the voting mechanism allows seamless swapping of individual models for each channel. As new and better

models emerge, they can be incorporated into the system without affecting the overall architecture.

- **Application**: Businesses and research projects can stay up-to-date with cutting-edge developments, enabling more accurate emotion detection without overhauling the entire system.

A clear demonstration of this system's flexibility can be seen in the **facial expression channel** of our demo. In the project, we showcased the ability to switch between **Facetorch** and **OpenFace**, two distinct facial expression analysis models. Despite their different methods of processing (Facetorch excels in static frame analysis, while OpenFace uses dynamic video analysis with action units), both models ultimately produced results that were compatible with our universal format.

4.3.4 Enhanced Robustness through Multi-Channel Fusion

Relying on a single channel (e.g., text or voice tone) for sentiment analysis often fails to capture the complexities of human emotions. By combining outputs from multiple channels, the voting mechanism improves the reliability of the final sentiment prediction, even when one channel is ambiguous or noisy.

- **Advantage**: Multi-channel fusion mitigates the weaknesses of individual modalities, leading to a more balanced and accurate sentiment analysis. This robustness is especially valuable in real-world applications where data quality can vary significantly.

- **Application**: Emotion analysis in video conferencing tools, where background noise may distort voice tone or facial expressions might be partially obscured.

4.3.5 Conflict Resolution in Edge Cases

Traditional multimodal systems often struggle with conflicting signals (e.g., a happy tone of voice but a sad facial expression). Such conflicts can lead to incorrect predictions, as multimodal architectures typically lack transparent mechanisms to handle ambiguity.

- **Advantage**: The voting mechanism explicitly addresses conflicts by assigning weights to channels based on their reliability and the context of the analysis. This provides a clearer and more interpretable resolution to contradictory signals.

- **Application**: Situations where emotional ambiguity is common, such as detecting sarcasm or understanding mixed emotions in customer feedback.

4.3.6 Simplified Training and Maintenance

Early fusion multimodal systems require training complex neural networks that integrate data from multiple channels. This demands large datasets, extensive computational resources, and expertise in deep learning.

- **Advantage**: The voting mechanism eliminates the need for joint training by processing each channel independently. This reduces computational overhead and simplifies the overall development pipeline.

- **Application**: Small to medium-sized enterprises or research labs with limited access to high-performance computing resources can adopt this method more easily.

4.3.7 Transparency and Interpretability

Multimodal neural networks are often criticized for their "black-box" nature, where the decision-making process is opaque and difficult to interpret. This lack of transparency can hinder trust and usability in applications where explainability is critical.

- **Advantage**: The voting mechanism offers a more transparent framework by clearly showing how individual channel outputs contribute to the final decision. This enhances interpretability and trust in the system’s predictions.

- **Application**: Healthcare applications, where clinicians require a clear understanding of how an emotion detection system reached its conclusions.

4.3.8 Resilience to Missing or Noisy Data

Multimodal systems are highly sensitive to missing or noisy data in one of

the channels, which can significantly degrade their performance. For example, poor lighting might affect facial recognition, or background noise could distort voice tone analysis.

- **Advantage**: The voting mechanism handles such scenarios gracefully by down-weighting or excluding unreliable channels during post-processing. This ensures the system remains functional even in challenging conditions.

- **Application**: Remote interviews or online education platforms, where environmental factors can disrupt data quality.

4.3.9 Applicability Across Domains

While multimodal systems often cater to specific tasks (e.g., customer sentiment or healthcare diagnostics), the voting mechanism is highly versatile and can be adapted for diverse use cases.

- **Advantage**: This adaptability makes the system suitable for a wide range of applications, from entertainment to workplace productivity tools.

- **Application**: Personal assistants like Alexa or Siri could benefit from this mechanism to enhance their emotional understanding in diverse interactions.

Conclusion

This project developed a new way to analyze emotions by combining facial expressions, voice tone, and spoken words using a voting system. Unlike traditional methods that blend data early and can struggle with noise or conflicts, our system keeps each channel separate and combines the results later. This makes it easier to understand how the final emotion is decided and ensures more reliable results.

Our tests showed that this approach works well. Facial expression models, like Facetorch and OpenFace, gave useful insights, with OpenFace handling changes over time better and Facetorch doing well with still images. Voice tone and word analysis added extra details, showing the value of using multiple types of data together.

The voting system also proved to be flexible. It can handle disagreements between channels and adjust to missing or unclear data, making it a practical solution for real-world use. This system can help in areas like mental health monitoring, customer service, or improving interactions between people and technology.

Despite voting system strengths, the system does not claim to surpass state-of-the-art multimodal techniques in all scenarios. Instead, it provides a complementary solution, excelling in situations where modularity, interpretability, and robustness are paramount. Future iterations can further refine this framework by incorporating adaptive weighting strategies, exploring additional channels, and addressing nuanced emotional states like sarcasm or mixed emotions.

While the system does not claim to surpass the performance of tightly integrated multimodal solutions in controlled environments, it excels in handling nuanced edge cases, ambiguous signals, and scenarios where traditional methods struggle. This positions the system as a valuable complement to existing technologies, bridging the gap between theoretical advancements and practical applicability.

Future Work

From the results obtained, it is evident that while the performance is not flawless, the method shows clear improvements over the best single-channel sentiment analysis approach, which primarily relies on facial expressions. By leveraging a multi-channel approach, the system effectively mitigates the risk of biased outcomes inherent to single-channel methods. This is achieved through dynamic weighting of multiple channels, allowing the system to adaptively consider different inputs.

In the future, there is significant potential to refine this approach by analyzing deeper into specific human emotions. This would involve studying the intricate and often subtle patterns that define nuanced emotional states. Recognizing these nuanced patterns could lead to a more robust and accurate categorization of complex emotions. For instance, emotions like sarcastic anger, which may outwardly appear as happiness, could be more accurately identified through a combination of a happy facial expression, a neutral vocal tone, and an angry transcription of spoken words. Similarly, hidden sadness, an emotion that might superficially appear neutral or even positive, could be detected through the interplay of subtle signals such as a happy or neutral facial expression, a sad vocal tone, and either sad or neutral spoken content.

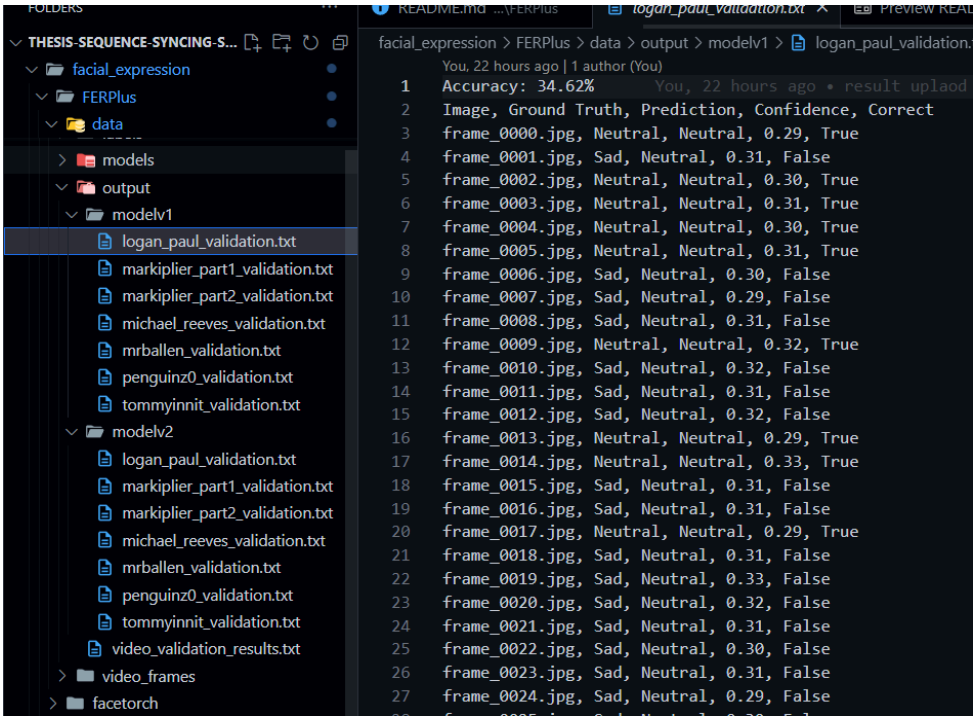
By understanding these subtleties, the system could be enhanced to recognize and appropriately classify these layered emotional states. Such improvements would strengthen the system's ability to handle the complexities of real-world emotional expressions, where emotions are rarely isolated or straightforward but instead often appear as blends of multiple states.

Appendix

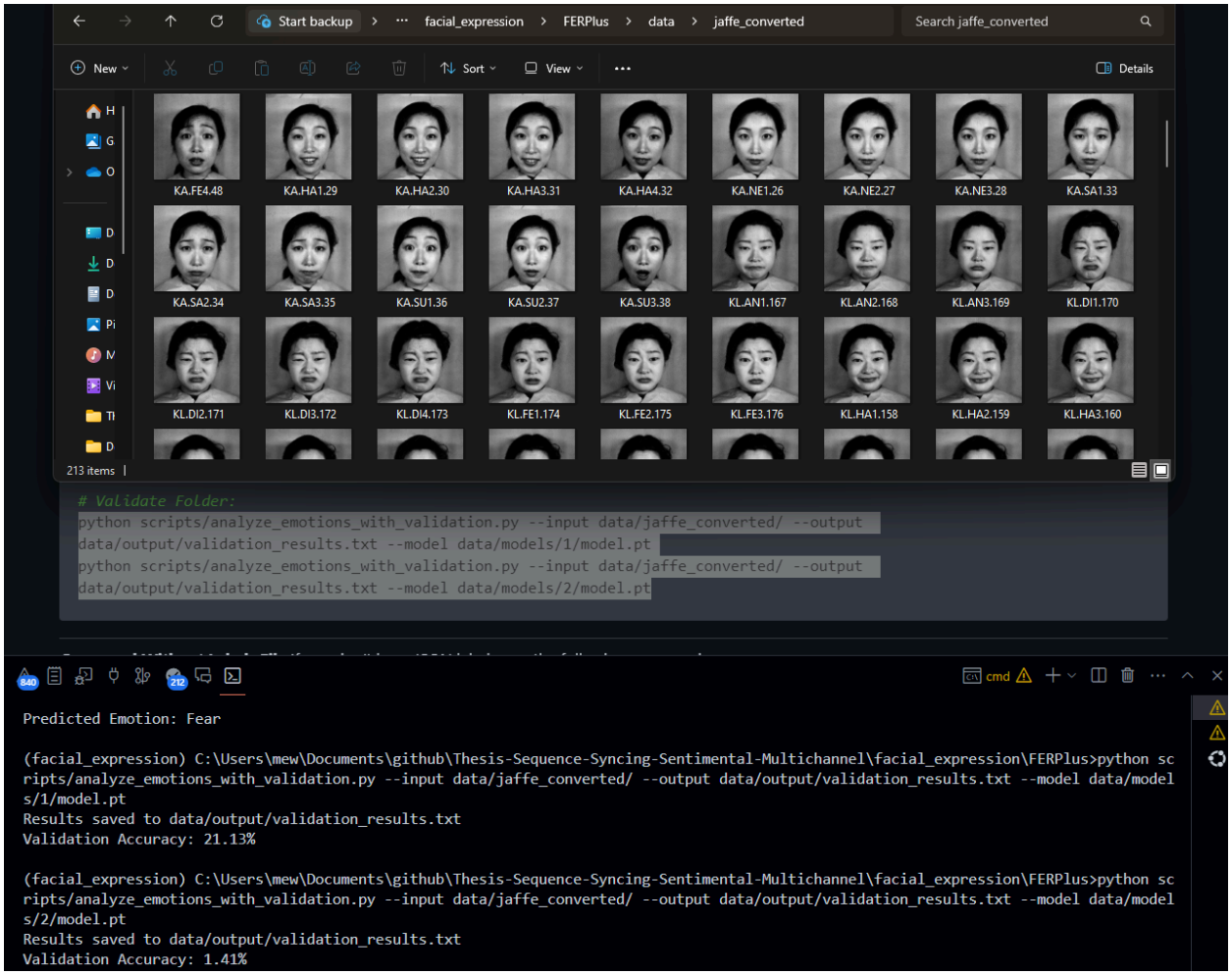
Link to our code and running script examples for replication of the experiment: https://github.com/mewakinHub/Thesis-Sequence-Syncing-Sentimental-Multichannel

```
(facial_expression) C:\Users\mew\Documents\github\Thesis-Sequence-Syncing-Sentimental-Multichannel\facial_expression\FERPlus>   valida
te_all.bat
model v1
Results saved to data\output\penguinz0_validation.txt
Validation Accuracy: 21.43%
Results saved to data\output\michael_reeves_validation.txt
Validation Accuracy: 30.00%
Results saved to data\output\tommyinnit_validation.txt
Validation Accuracy: 47.37%
Results saved to data\output\mrballen_validation.txt
Validation Accuracy: 13.33%
Results saved to data\output\markiplier_part1_validation.txt
Validation Accuracy: 31.25%
Results saved to data\output\markiplier_part2_validation.txt
Validation Accuracy: 3.12%
Results saved to data\output\logan_paul_validation.txt
Validation Accuracy: 34.62%
model v2
Results saved to data\output\penguinz0_validation.txt
Validation Accuracy: 10.71%
Results saved to data\output\michael_reeves_validation.txt
Validation Accuracy: 0.00%
Results saved to data\output\tommyinnit_validation.txt
Validation Accuracy: 10.53%
Results saved to data\output\mrballen_validation.txt
Validation Accuracy: 0.00%
Results saved to data\output\markiplier_part1_validation.txt
Validation Accuracy: 0.00%
Results saved to data\output\markiplier_part2_validation.txt
Validation Accuracy: 0.00%
Results saved to data\output\logan_paul_validation.txt
Validation Accuracy: 65.38%
```

FaceTorch Log Result (FER like)



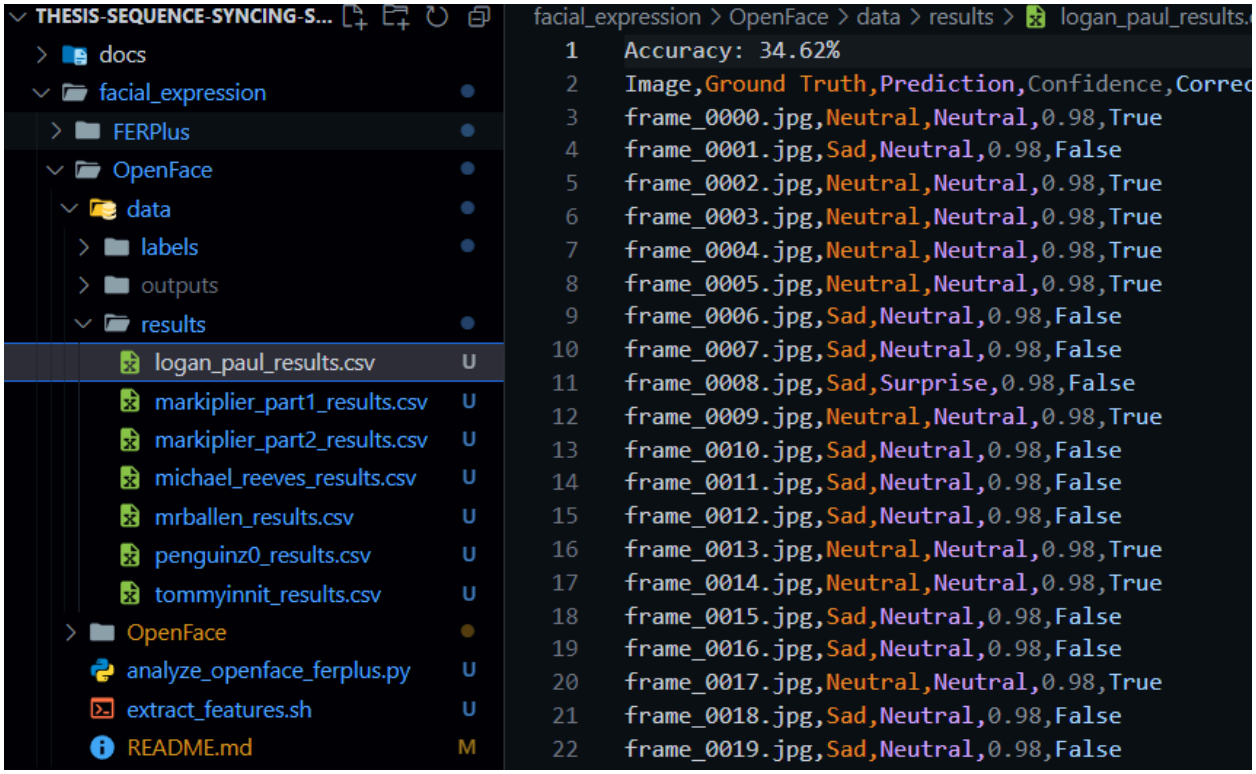
FaceTorch File Result (FER like)



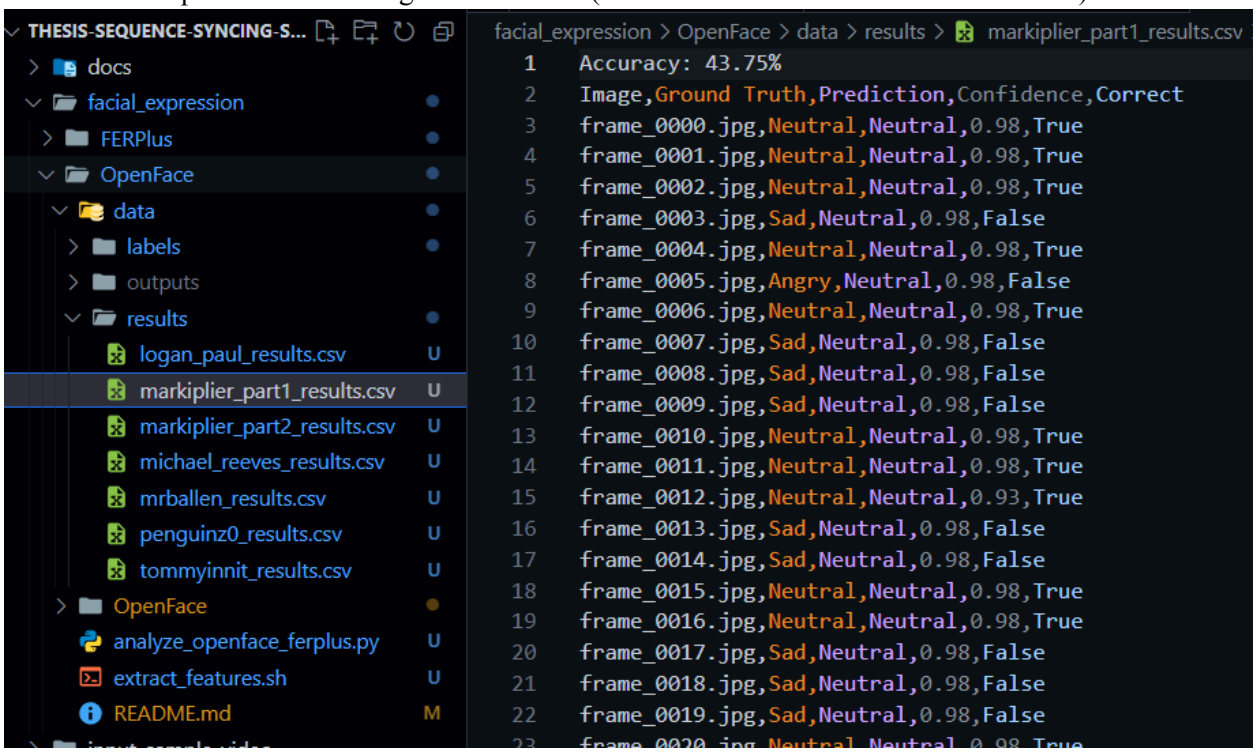
Evaluation on JEFFE(Japanese Female Facial Expression) Dataset

```
(facial_expression) C:\Users\mew\Documents\github\Thesis-Sequence-Syncing-Sentimental-Multichannel\facial_
\OpenFace>python analyze_openface_ferplus.py
Processing: logan_paul
Results saved to ./data/results\logan_paul_results.csv with Accuracy: 34.62%
Processing: markiplier_part1
Results saved to ./data/results\markiplier_part1_results.csv with Accuracy: 43.75%
Processing: markiplier_part2
Results saved to ./data/results\markiplier_part2_results.csv with Accuracy: 9.38%
Processing: michael_reeves
Results saved to ./data/results\michael_reeves_results.csv with Accuracy: 70.00%
Processing: mrballen
Results saved to ./data/results\mrballen_results.csv with Accuracy: 96.67%
Processing: penguinz0
Results saved to ./data/results\penguinz0_results.csv with Accuracy: 21.43%
Processing: tommyinnit
Results saved to ./data/results\tommyinnit_results.csv with Accuracy: 36.84%
```

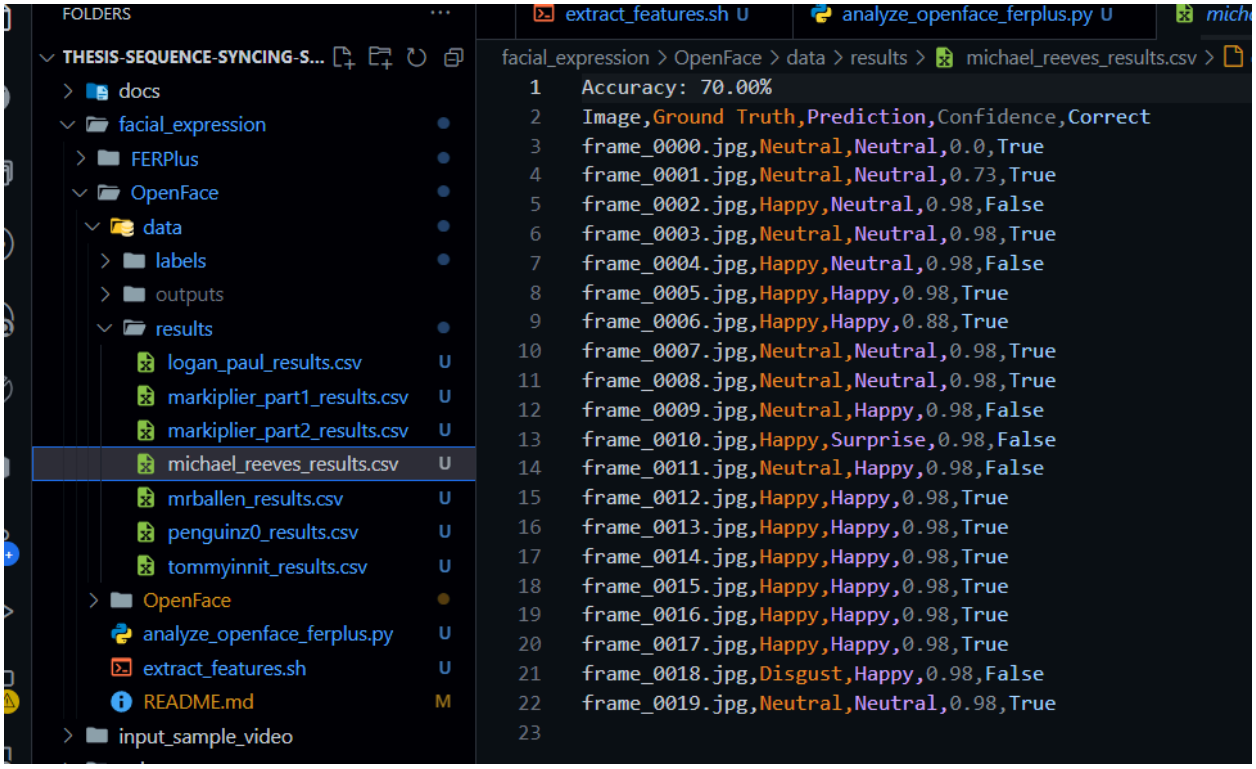
OpenFace model Log Result (alternative switched model of FaceTorch)



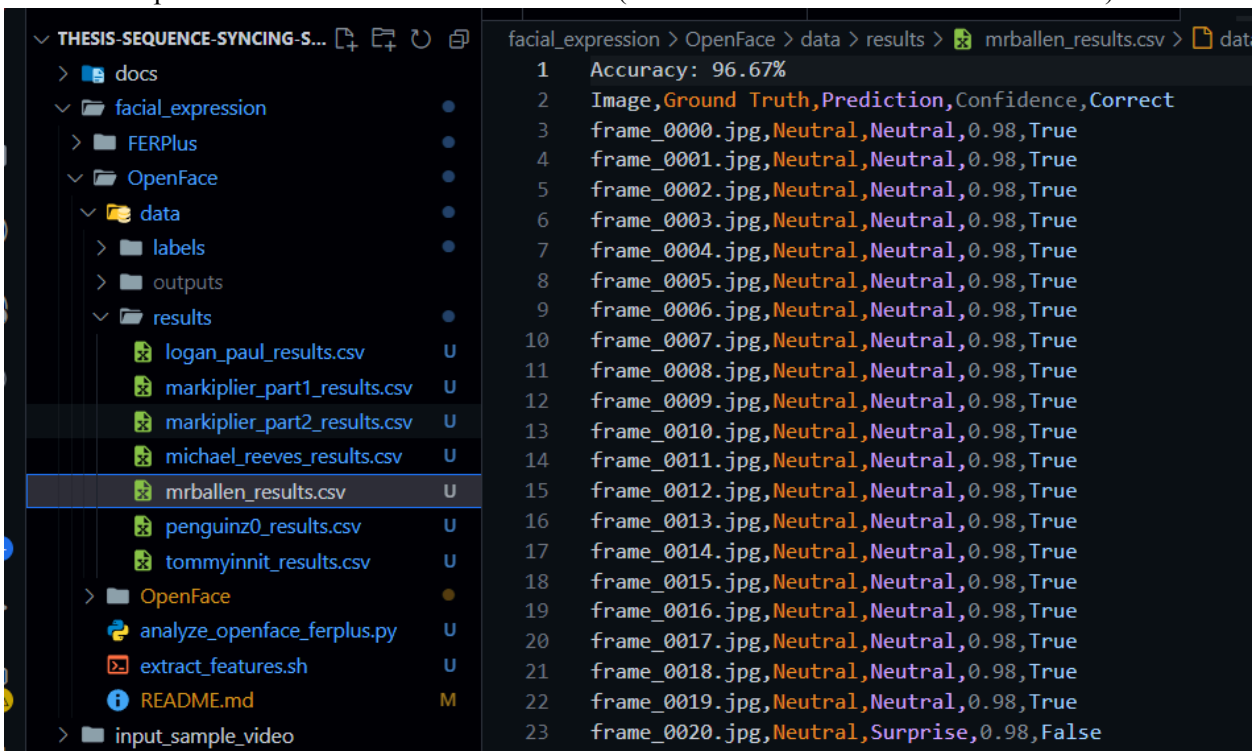
OpenFace model Logan Paul Result (alternative switched model of FaceTorch)



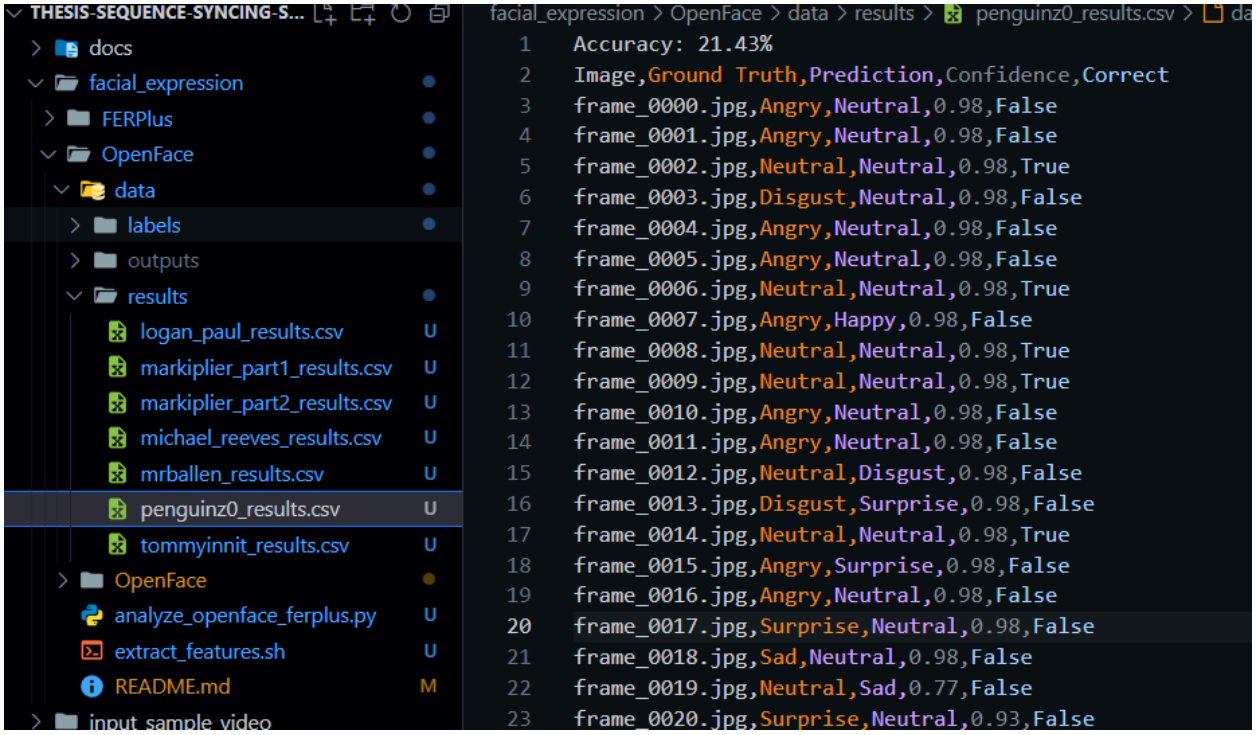
OpenFace model Markiplier Result (alternative switched model of FaceTorch)



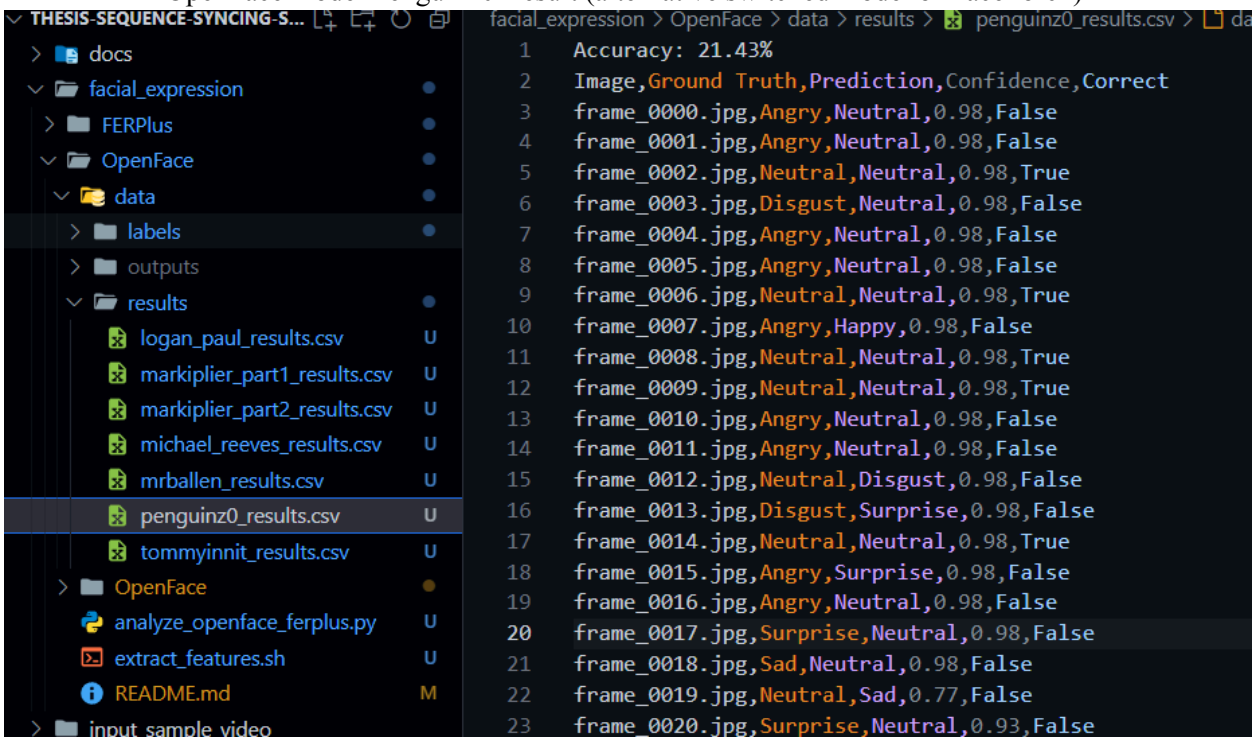
OpenFace model Michael Reeves Result (alternative switched model of FaceTorch)



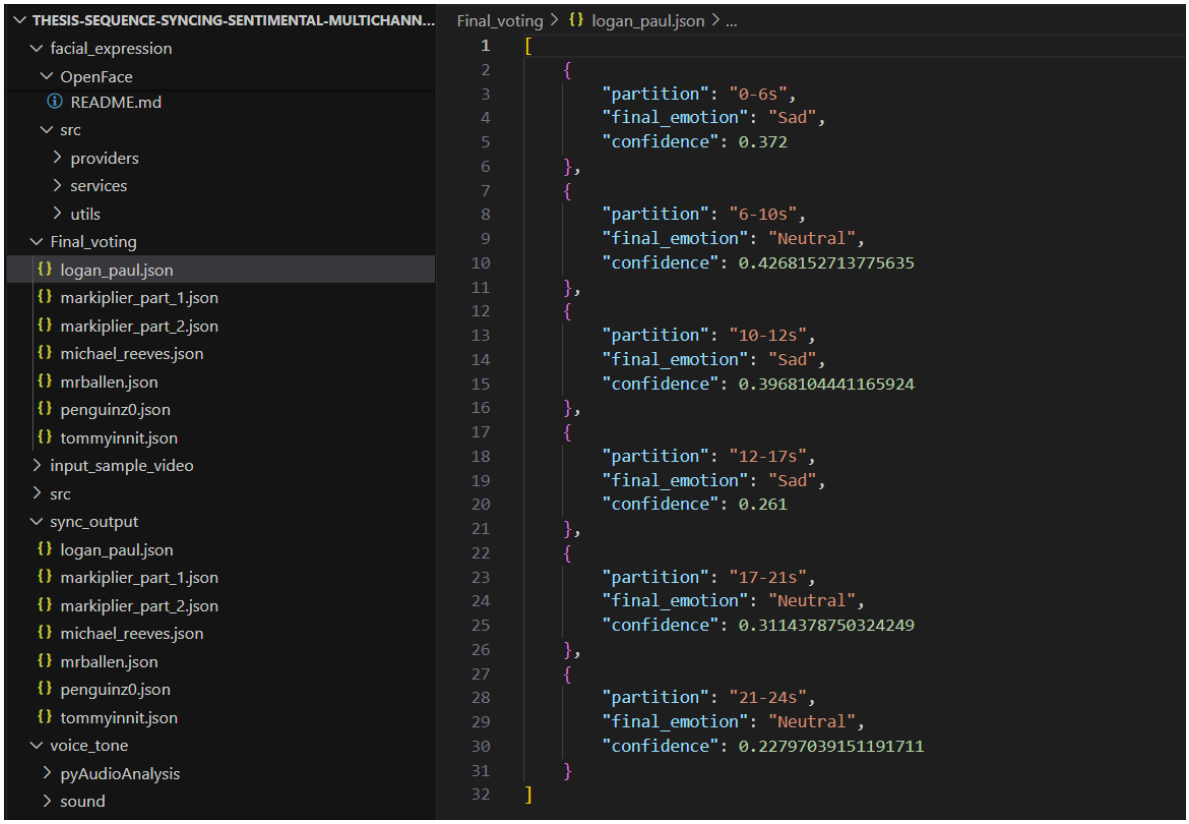
OpenFace model Mrballen Result (alternative switched model of FaceTorch)



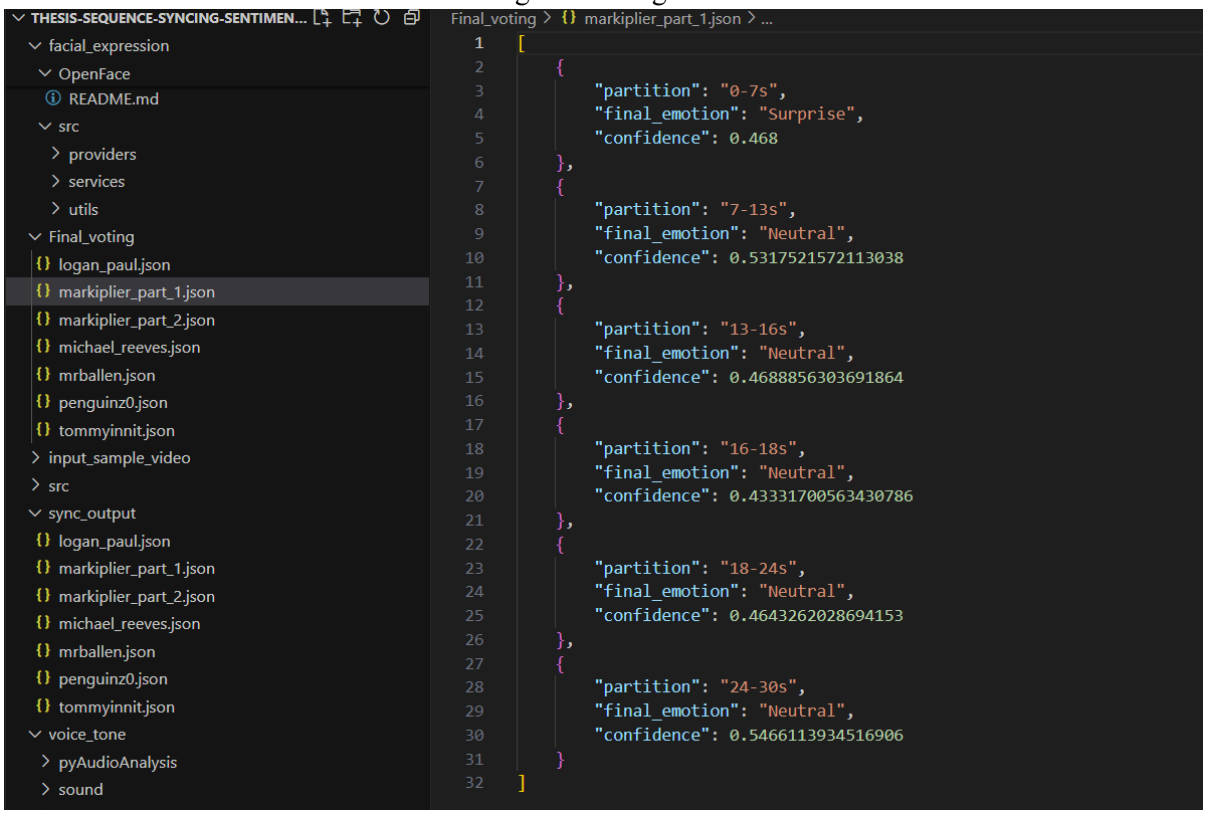
OpenFace model Penguinz0 Result (alternative switched model of FaceTorch)



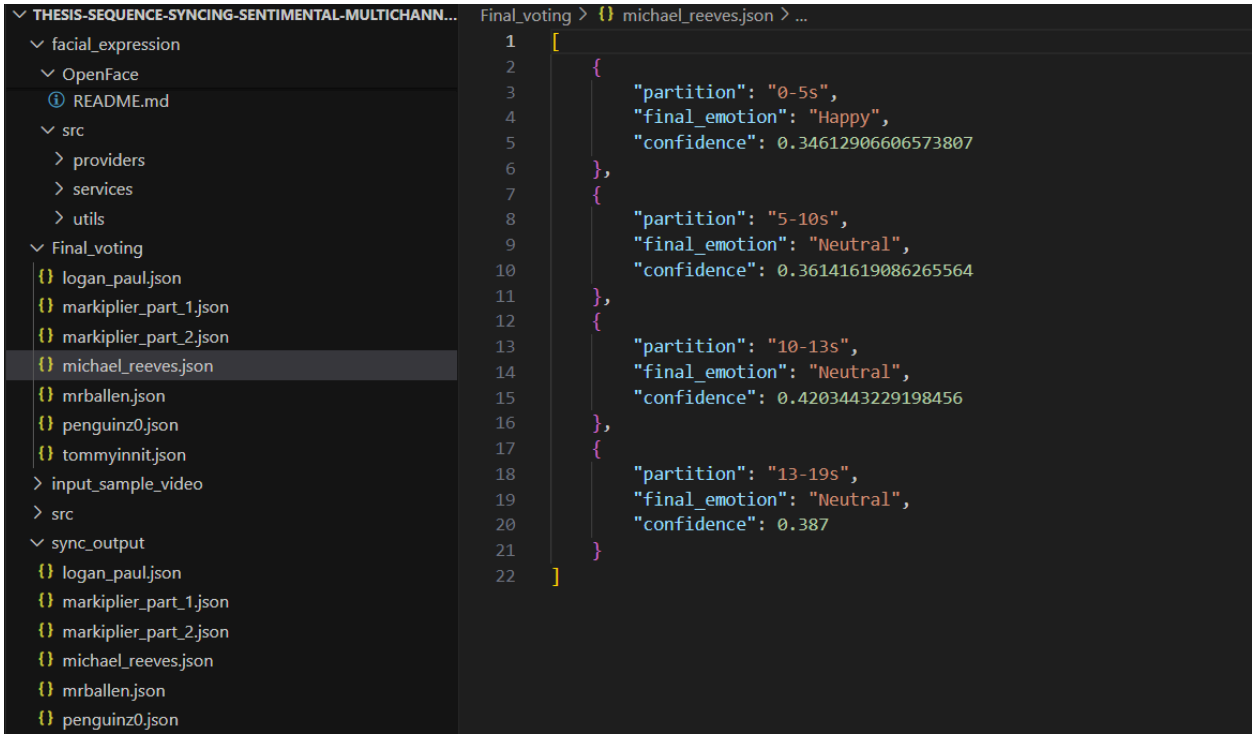
OpenFace model Tommyinnit Result (alternative switched model of FaceTorch)



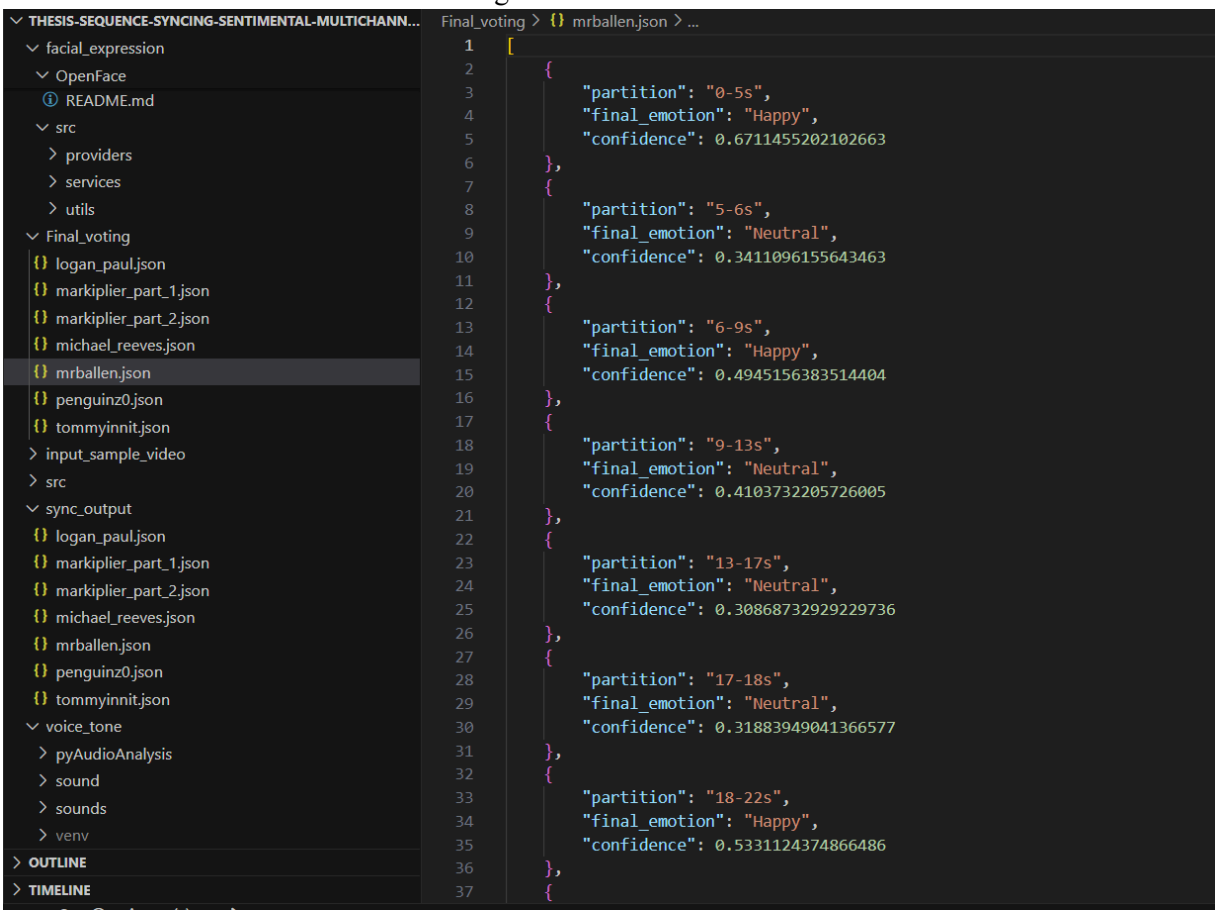
Voting Result Logan Paul



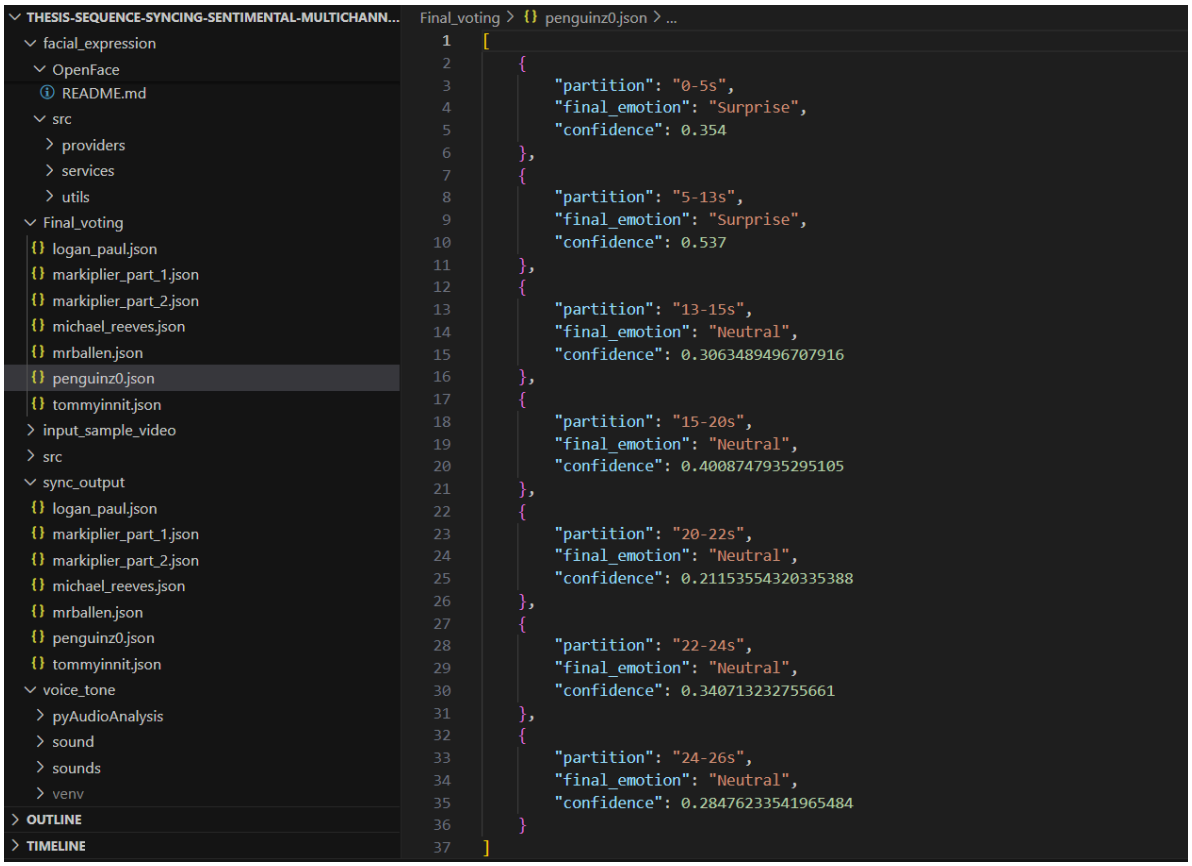
Voting Result Markiplier



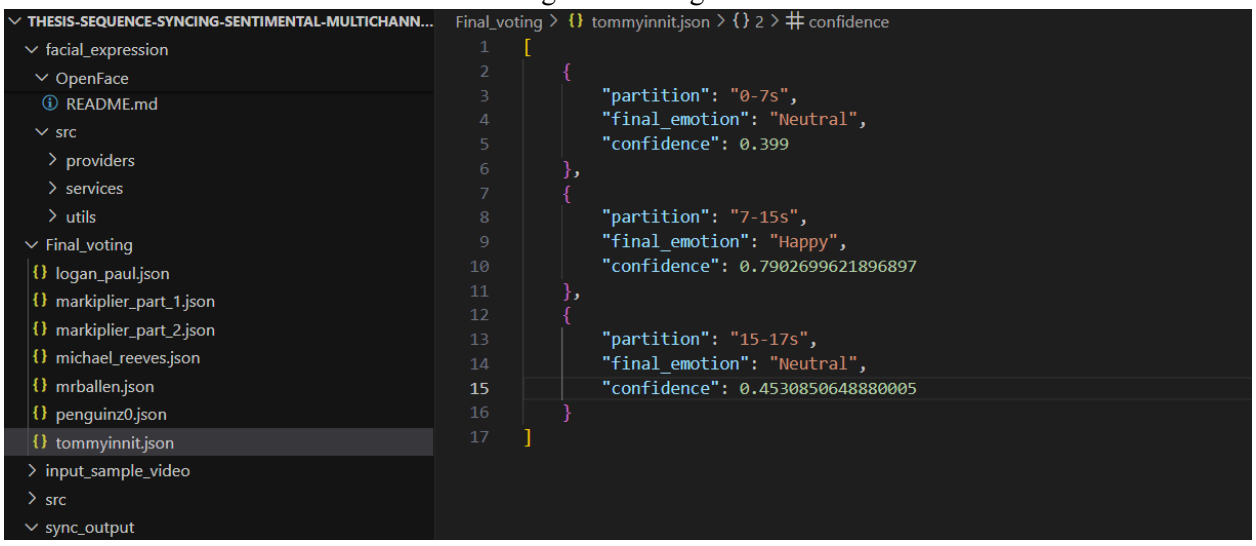
Voting Result Michael Reeves



Voting Result Mrballen



Voting Result Penguinz0



Voting Result Tommyinnit

Bibliography

1. Poria, S., Majumder, N., Mihalcea, R., & Hovy, E. (n.d.). Emotion recognition in conversation: Research challenges, datasets, and recent advances. Retrieved from https://paperswithcode.com/paper/emotion-recognition-in-conversation-research

2. Seunghyun Yoon, Seokhyun Byun, & Kyomin Jung. Multimodal Speech Emotion Recognition Using Audio and Text [Papers With Code]. Retrieved from https://paperswithcode.com/paper/multimodal-speech-emotion-recognition-using

3. Tzirakis, P., Trigeorgis, G., Nicolaou, M. A., Schuller, B., & Zafeiriou, S. (n.d.). End-to-end multimodal emotion recognition using deep neural networks. Retrieved from https://paperswithcode.com/paper/end-to-end-multimodal-emotion-recognition

4. Sangineto, A., Liu, H., & Xu, L. (2020, September). Nonparallel emotional speech conversion using VAE-GAN. In INTERSPEECH 2020 (pp. 1043-1047). International Speech Communication Association. Retrieved from https://paperswithcode.com/paper/nonparallel-emotional-speech-conversion

5. Hazarika, D., Poria, S., Zimmermann, R., & Mihalcea, R. (2019). Conversational transfer learning for emotion recognition. Retrieved from https://paperswithcode.com/paper/emotion-recognition-in-conversations-with

6. Schuller et al. (2013). AVEC 2013 - The continuous audio/visual emotion and Depression Recognition Challenge. (n.d.). Retrieved from https://www.researchgate.net/publication/262157517_AVEC_2013_-_The_continuous_AudioVisual_Emotion_and_depression_recognition_challenge

7. Hartmann, J. (n.d.). *Emotion English DistilRoBERTa base* [Hugging Face Model]. Retrieved from https://huggingface.co/j-hartmann/emotion-english-distilroberta-base

8. Dpngtm. (n.d.). *Wav2Vec2 Emotion Recognition* [Hugging Face Model]. Retrieved from https://huggingface.co/Dpngtm/wav2vec2-emotion-recognition

9. Ehcalabres. (n.d.). Wav2Vec2-Large-XLSR-EN Speech Emotion Recognition [Hugging Face Model]. Retrieved from https://huggingface.co/ehcalabres/wav2vec2-lg-xlsr-en-speech-emotion-recognition

10. Facebook. (n.d.). *Wav2Vec2-Large-XLSR-53* [Hugging Face Model]. Retrieved from https://huggingface.co/facebook/wav2vec2-large-xlsr-53

11. Bird, S., Klein, E., & Loper, E. (n.d.). Natural Language Toolkit (NLTK) [Software]. Retrieved from https://www.nltk.org/

12. Gajarsky, T. (n.d.). Facetorch: A facial expression recognition library [Computer software]. GitHub. Retrieved from https://github.com/tomas-gajarsky/facetorch

13. Gajarsky, T. (n.d.). Facetorch demo app [Demo application]. Hugging Face. Retrieved from https://huggingface.co/spaces/tomas-gajarsky/facetorch-app

14. Papers With Code. (n.d.). Facial Expression Recognition (FER) dataset benchmarks. Retrieved from https://paperswithcode.com/task/facial-expression-recognition?page=3

15. Baltrusaitis, T., Robinson, P., & Morency, L.-P. (2016). OpenFace: An open source facial behavior analysis toolkit. Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV). Retrieved from https://github.com/TadasBaltrusaitis/OpenFace