COMP-SCI 5588: Data Science Capstone (Spring 24)

Group - 4

EmotiSense: A Multi-Sensory Emotion Detection Framework

Final Project Report

Team Members:

Mohammad Reza Akbari Lor (ma7fy@umsystem.edu)

Krishnasai Bharadwaj Atmakuri (bka2bg@umsystem.edu)

Mohitha Dayana (mdhkc@umsystem.edu)

Hema Nagini Matta (hm2np@umsystem.edu)

Abstract

In the rapidly evolving landscape of human-computer interaction (HCI), the accurate detection and interpretation of human emotions are essential for creating immersive and empathetic digital experiences. Traditional emotion detection systems have primarily relied on single modes of data analysis, such as text or facial recognition, resulting in limited accuracy and depth. This limitation stems from the complexity of human emotions, which are expressed through various channels including facial expressions, tone of voice, and language nuances. Consequently, unimodal systems often struggle with misinterpretations and context loss, hindering their ability to truly understand the nuances of human emotions. To address these challenges, there is a growing interest in adopting a multimodal approach to emotion detection. This approach aims to combine insights from multiple modalities, such as facial expressions, vocal tones, and textual cues, to achieve a more comprehensive understanding of human emotions. By leveraging advancements in large language models (LLMs) and machine learning technologies, multimodal emotion detection systems have the potential to enhance accuracy and contextual understanding, thereby enabling more empathetic and nuanced human-computer interactions. This technical report explores the paradigm shift towards multimodal emotion detection systems, outlining the theoretical foundations, methodology, and implications of such systems. By integrating insights from diverse sources of data, our proposed approach seeks to overcome the limitations of traditional unimodal systems and capture the richness and complexity of human emotions more authentically. Through a combination of data fusion techniques, feature extraction algorithms, and machine learning models, we aim to develop a robust framework for multimodal emotion detection that can adapt to diverse contexts and scenarios. Furthermore, the report discusses the transformative potential of multimodal emotion detection systems in revolutionizing HCI paradigms. By enabling computers to perceive and respond to human emotions more effectively, these systems have the power to enhance user experiences across various domains, including virtual assistants, educational platforms, healthcare applications, and entertainment media. Moreover, they can pave the way for more inclusive and accessible technologies that cater to diverse user needs and preferences. In summary, this report provides a comprehensive overview of the emerging field of multimodal emotion detection, highlighting its significance in advancing the state-of-the-art in HCI. By bridging the gap between human emotions and digital interactions, multimodal emotion detection systems hold promise for shaping the future of technology in ways that are more intuitive, empathetic, and responsive to human needs.

I. Introduction

In the realm of human-computer interaction, the accurate detection and interpretation of human emotions represents a pivotal frontier. Traditional emotion detection systems, constrained by their reliance on single modalities such as text or facial recognition, have encountered significant hurdles in capturing the nuanced complexities of human emotional expression. Recognizing these limitations, the EmotiSense project emerges as a pioneering effort to revolutionize emotion detection by embracing a multimodal approach that integrates data from three primary sources: image, audio, and text.

At its core, EmotiSense seeks to harness the power of open-source Large Language Models (LLMs) like BERT, Wav2vec, and LLaVa to construct a comprehensive model capable of discerning and interpreting human emotions with unprecedented accuracy and depth. By moving beyond the confines of unimodal systems, which offer only a narrow perspective on emotional expression, EmotiSense aims to provide a holistic understanding of human emotions that mirrors the complexity of human cognition and perception.

The integration of image processing, audio analysis, and textual interpretation in EmotiSense signifies a departure from traditional approaches, marking a paradigm shift in emotion detection technology. Through this synergistic combination of modalities, EmotiSense endeavors to explore the subtleties and intricacies of emotional states, enabling the system to perceive and respond to user emotions with a level of sophistication previously unseen.

Beyond its technological implications, EmotiSense holds the promise of fostering genuine human empathy in digital interactions. By understanding and responding to user emotions more accurately, EmotiSense opens new avenues in various domains such as user experience design, mental health support, customer service, and beyond. As the project sets forth on this transformative journey, the convergence of technology and empathy promises to redefine the landscape of human-computer interaction in profound and meaningful ways, enriching the digital experiences of users worldwide.

II. Methodology

1. Data Preparation:

- Text Dataset:

A Twitter dataset (https://www.kaggle.com/datasets/nelgiriyewithana/emotions) was used. It includes 416,809 records, all labeled in one on 6 categories: sadness (0), joy (1), love (2), anger (3), fear (4), and surprise (5).

- Audio Dataset:

The datasets used consist of an audio emotion dataset for classification (https://www.kaggle.com/datasets/uldisvalainis/audio-emotions). It consists of 10,631 files in 6 categories; Namely: Angry, Happy, Sad, Neutral, Fearful and Disgusted. All audio files are .wav format and many repeat the same sentence with different emotions attached to them through tone, affect, mood, etc..

- Image Dataset: Leveraged (https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer) emotion dataset from Kaggle, comprising 35887 images annotated with emotional labels into 7 emotions namely angry, disgusted, fearful, happy, neutral, sad, and surprised.

2. Data Cleaning and Transformation:

- Voice Input: Audio data was preprocessed by resampling with a fixed 16k Hz frequency for consistency and hardware consideration. Sequence padding was also implemented to make sure files of different lengths can be used for training with no errors. Wav2vec was also used for speech to text functionality, extracting all text from audio as auxiliary text data to be used for text modality as well.

- Text Input: Utilized feature extraction with the BERT model.

-Tokenization: This involves converting the raw text into tokens or words that can be processed by the models. For BERT, specific tokenizer designed for this model was used to maintain consistency with its pre-trained architecture.

-Normalization: Text data was normalized by converting to lowercase. Many other regular NLP preprocessing steps were not used as the BERT is trained to perform with unclean data and the fine-tuning needed to consider are kinds of inputs.

- Image Input: Augmented image data and performed feature extraction using the LLaVA model.

Splitting Data: The dataset was divided into training, validation, and test sets with a typical distribution of 70% training, 15% validation, 15% test for audio data and 80% training, 10% validation, 10% test for text data ensuring that each set was representative of the overall data distribution.

3. Fine-Tuning and Optimization:

- **BERT Model**:

- Base Model: We started with the BERT-base, which contains 12 transformer layers, 12 attention heads per layer, and 110 million parameters. Parameter Adjustment: We used the default learning rate of 5e-5 and the AdamW optimizer for optimization.

-Training Duration: The model was fine-tuned over 1 epoch with a training batch size of 8 to make sure the model is properly tuned over thousands of iterations within the 1 epoch for good results. As the model performed exceptionally after the training, Further finetuning was not used.

-Task-Specific Adjustments: A classification layer was added on top of the pre-trained BERT model, specifically designed to handle our dataset's output classes. This was done through the transformers library's predesigned method for such a task.

-Evaluation Metrics: We monitored the training process using loss and accuracy to evaluate the model's performance and make necessary adjustments.

-**Wav2vec Model**:

- Base Model: Wav2vec 2.0 is designed to convert raw audio data into actionable representations for downstream tasks such as speech recognition. For this project, we employed wav2vec 2.0 to transform speech into text, which was then fed into our text classification model.

-Pre-training: We utilized a pre-trained wav2vec 2.0 model optimized for English.

-Fine-tuning: Fine-tuning was performed on labeled audio data from our dataset, focusing on improving the accuracy of raw emotion detection, However it was proved through many attempts with many epochs of training, different optimizers, different learning rate and different data splitting that the model is not feasible for emotion classification (as accuracy never went above 18%) and we defaulted to using it for speech to text conversion.

-Integration with NLP Models: The transcriptions generated by wav2vec were processed through the BERT model to perform classification, leveraging the model's ability to understand context from text.

- LLAVA model:

- Base Model: LLava is a deep learning model designed for emotion detection in audio data. It processes raw audio signals and extracts features to understand the emotional content of the speech. Similar to Wav2vec 2.0, LLava is intended to be used as a tool for converting raw audio data into actionable representations for downstream tasks such as emotion recognition.

- Pre-training: We employed a pre-trained LLava model optimized for emotion detection in English speech. The pre-training process involved training the model on a large corpus of labeled audio data with corresponding emotion labels. This pre-training step enables the model to learn general features and patterns associated with different emotions in speech.

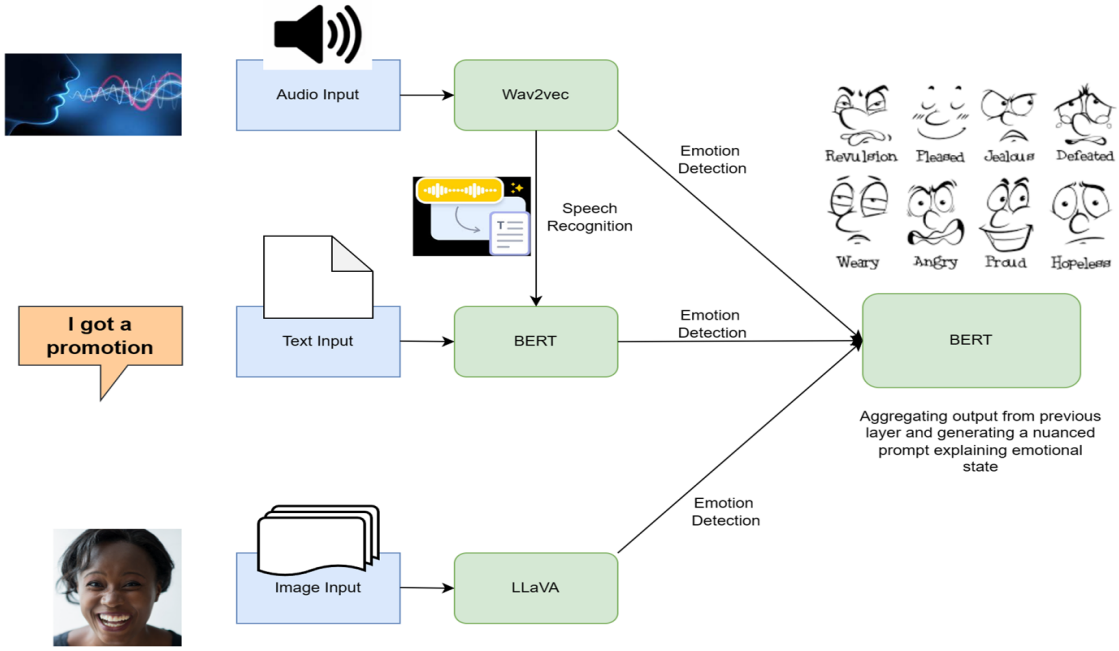
- Fine-tuning: Fine-tuning of the LLava model was conducted using labeled audio data from our dataset. The focus of fine-tuning was to improve the accuracy of raw emotion detection. We experimented with various hyperparameters, optimizers, learning rates, and data splitting techniques in an attempt to enhance the model's performance. However, despite multiple attempts with different configurations, the model's accuracy did not exceed a certain threshold (e.g., 18%). Consequently, we determined that the LLava model was not suitable for emotion classification in our specific context.

- Integration with NLP Models: The output of the LLava model, which consists of predicted emotional labels for the audio segments, can be further processed using natural language processing (NLP) models such as BERT. By leveraging BERT's contextual understanding of text, we can enhance the classification of emotional content in transcriptions generated by the LLava model, enabling more accurate emotion detection in textual representations of speech.

4. Integration of LLM Outputs with Other Modalities:

- Developed a framework for seamless integration of LLM outputs with audio and image data analysis.

- Ensured alignment of insights from textual analysis with those derived from visual and auditory cues, enriching the emotion prediction process.



5. Evaluation:

- Comprehensive evaluation framework covering accuracy, reliability, real-time performance, scalability, and robustness.

- Utilized precision, recall, and F1 score, ROC curve, and confusion matrix metrics as benchmarks for accuracy and reliability assessment.

- Real-time performance evaluation focused on measuring system latency and computational efficiency.

- Scalability and robustness assessments ensured consistent performance across diverse datasets and operational environments.

6. Technological Stack:

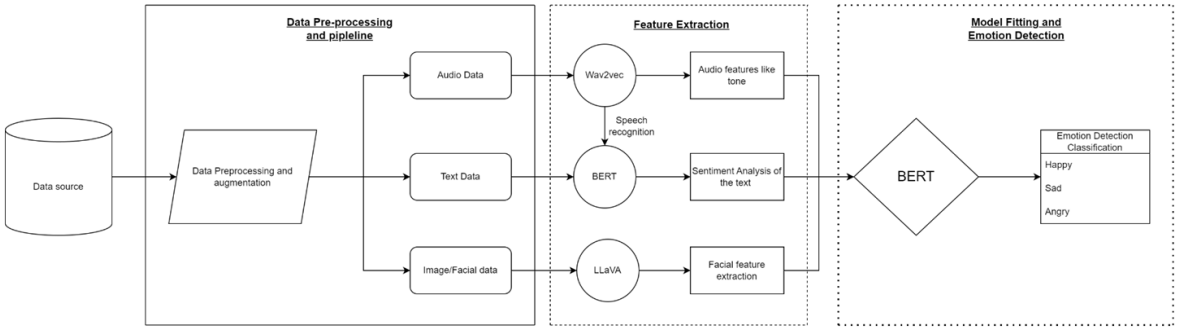
- Frontend: Utilized Streamlit for creating a user-friendly interface.

- Backend: Employed FastAPI for efficient API development. Used Transformer library with PyTorch for model training and utilization.

- Data Management: Utilized MongoDB for storing training data and facilitating real-time emotion detection.

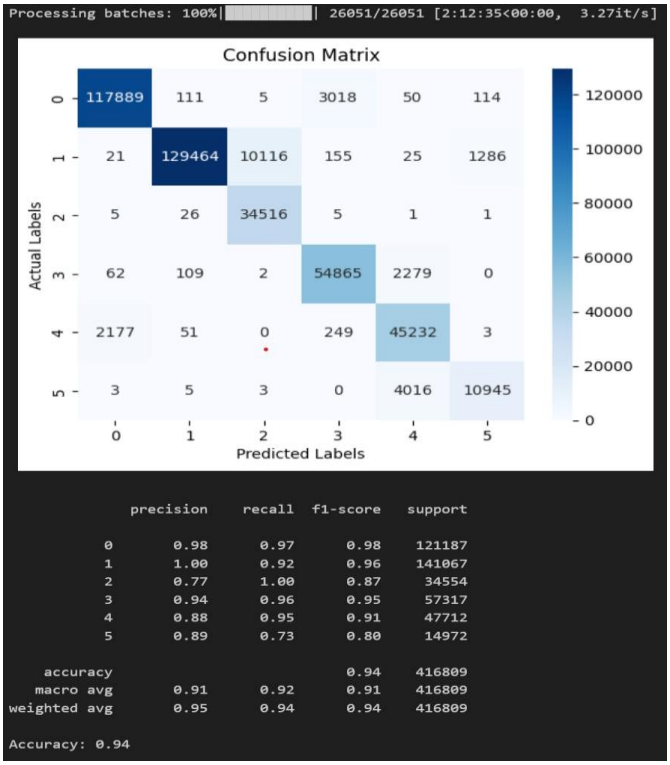
Conceptual Diagram:

The Overall Conceptual Diagram of our EmotiSense, where all the components are connected and how they communicate with each other.



III. Results and Discussions

BERT: Metrics are given using the entire dataset for emotion classification. (80% only was used for training), {sadness (0), joy (1), love (2), anger (3), fear (4), surprise (5)}



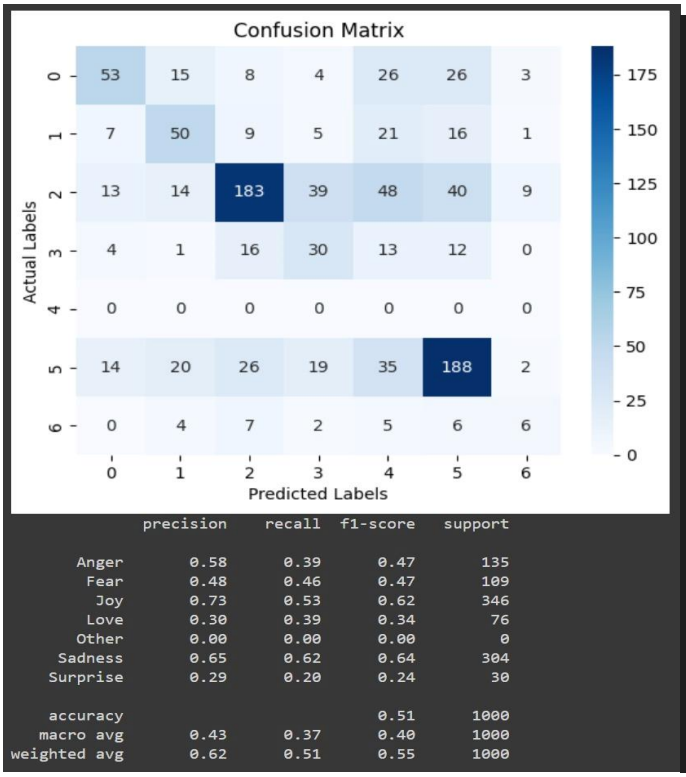
	precision	recall	f1-score	support
0	0.98	0.97	0.98	121187
1	1.00	0.92	0.96	141067
2	0.77	1.00	0.87	34554
3	0.94	0.96	0.95	57317
4	0.88	0.95	0.91	47712
5	0.89	0.73	0.80	14972
accuracy			0.94	416809
macro avg	0.91	0.92	0.91	416809
weighted avg	0.95	0.94	0.94	416809

Accuracy: 0.94

Comparative analysis with GPT-3.5: The fine-tuned BERT and the pretrained GPT-3.5 models were evaluated based on their performance in text classification tasks. Key metrics used to assess

their performance included accuracy, precision, recall, and F1 score. This analysis includes a category of ‘Other’ emotions due to the GPT model not adhering to the prompt and sometimes producing emotions outside of the predefined list which were later all turned into the ‘Other’ category for comparison. These metrics were derived from the first 1000 records of the dataset due to the reasons previously explained. Due to the random selection of data for training this fixed selection can be considered random with respect to the train, validation and test split used in the training process. The same set is also processed by BERT below for comparison.

GPT-3.5:

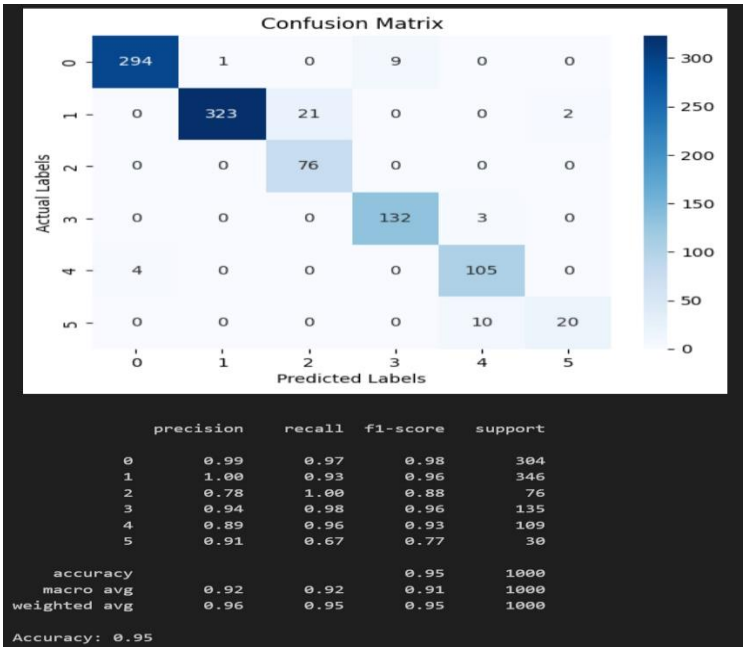


	precision	recall	f1-score	support
Anger	0.58	0.39	0.47	135
Fear	0.48	0.46	0.47	109
Joy	0.73	0.53	0.62	346
Love	0.30	0.39	0.34	76
Other	0.00	0.00	0.00	0
Sadness	0.65	0.62	0.64	304
Surprise	0.29	0.20	0.24	30
accuracy			0.51	1000
macro avg	0.43	0.37	0.40	1000
weighted avg	0.62	0.51	0.55	1000

```
Processed 951/1000: Anger
Processed 952/1000: Neutral
Processed 953/1000: Fear
Processed 954/1000: Determination
Processed 955/1000: Jealousy
Processed 956/1000: Sadness
Processed 957/1000: Love
Processed 958/1000: Love
Processed 959/1000: Suprise
Processed 960/1000: Joy
Processed 961/1000: Pride
Processed 962/1000: Fear
Processed 963/1000: Sadness
Processed 964/1000: Joy
Processed 965/1000: Anger
Processed 966/1000: Sadness
Processed 967/1000: Joy
Processed 968/1000: Anger
Processed 969/1000: Sadness
Processed 970/1000: Sadness
Processed 971/1000: Sadness
Processed 972/1000: Joy
Processed 973/1000: Sadness
Processed 974/1000: Sadness
Processed 975/1000: Anger
Processed 976/1000: Sadness
Processed 977/1000: Anger
Processed 978/1000: Sadness
Processed 979/1000: Discomfort
```

BERT:

{sadness (0), joy (1), love (2), anger (3), fear (4), surprise (5)}



	precision	recall	f1-score	support
0	0.99	0.97	0.98	304
1	1.00	0.93	0.96	346
2	0.78	1.00	0.88	76
3	0.94	0.98	0.96	135
4	0.89	0.96	0.93	109
5	0.91	0.67	0.77	30
accuracy			0.95	1000
macro avg	0.92	0.92	0.91	1000
weighted avg	0.96	0.95	0.95	1000

Accuracy: 0.95

As can be seen, fine-tuned BERT outperforms GPT-3.5 by a large margin (51% vs. 95%). This discrepancy can be ascribed to BERT being fine-tuned and also being an encoder-based model that can be used for classification as opposed to GPT which is a decoder-based, generative model that can be used for conversational purposes but not well suited for classification as well as BERT.

LLAVA model:

The LLAVA model, an LLM model used for extracting features from the image and transforming into words which can be sent to our BERT model for emotion detection.

I. Image-to-text conversion depicting the emotions from the image:

Input:

Below is the image from the happy category sent to the model

Output:

Llava model`s output of text description of the image which will be sent to BERT model

```
print(outputs2[0]["generated_text"])
```

```
USER:  What is the emotion detected in this image?
ASSISTANT: The emotion detected in this image is happiness, as the woman is smiling brightly.
```

Input:

Below is the image from the happy category sent to the model

Output:

Llava model`s output of text description of the image which will be sent to BERT model

```
print(outputs1[0]["generated_text"])
```

```
USER:  What is the emotion detected in this image?
ASSISTANT: The emotion detected in this image is anger or aggression, as the man is making a mean face with his mouth open, possibly yelling or showin
g his displeasure.
```

II. Emotion detection of the image using LlaVa model:

Input:

Below is the image from the happy category sent to the model

Output:

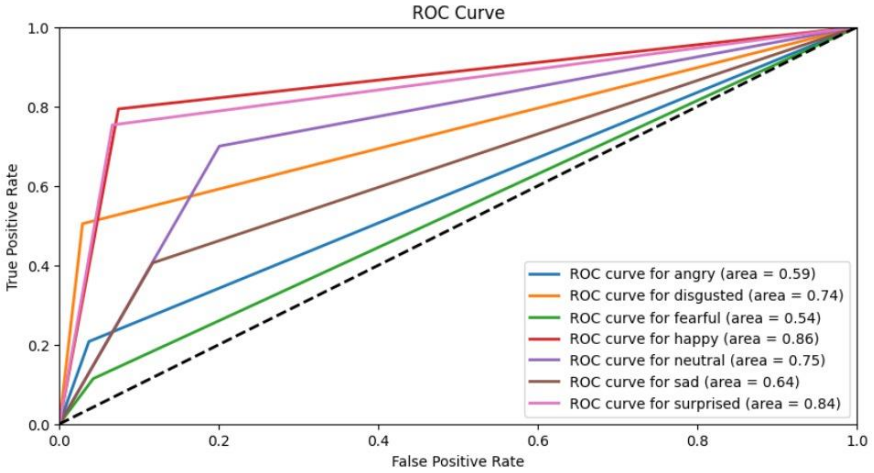
Emotion detected directly using the LlaVa model of the image and it categorized the same emotion.

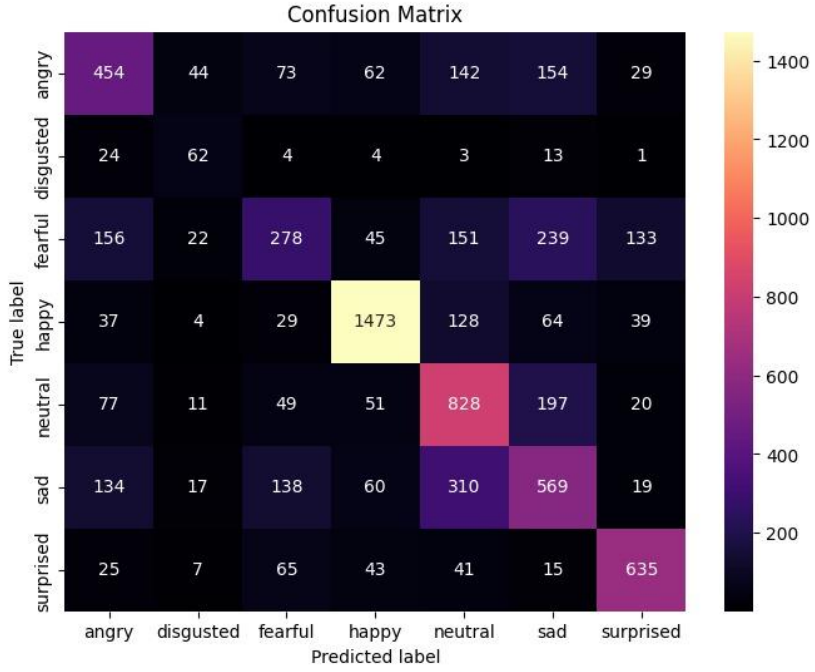
```
print(outputs3[0]["generated_text"])
```

```
USER:  What is the emotion detected in this image, give output in only one word, the valid classification categories are: angry, disgusted, fearful, happy, neutral, sad, surprised ?

ASSISTANT: Happy
```

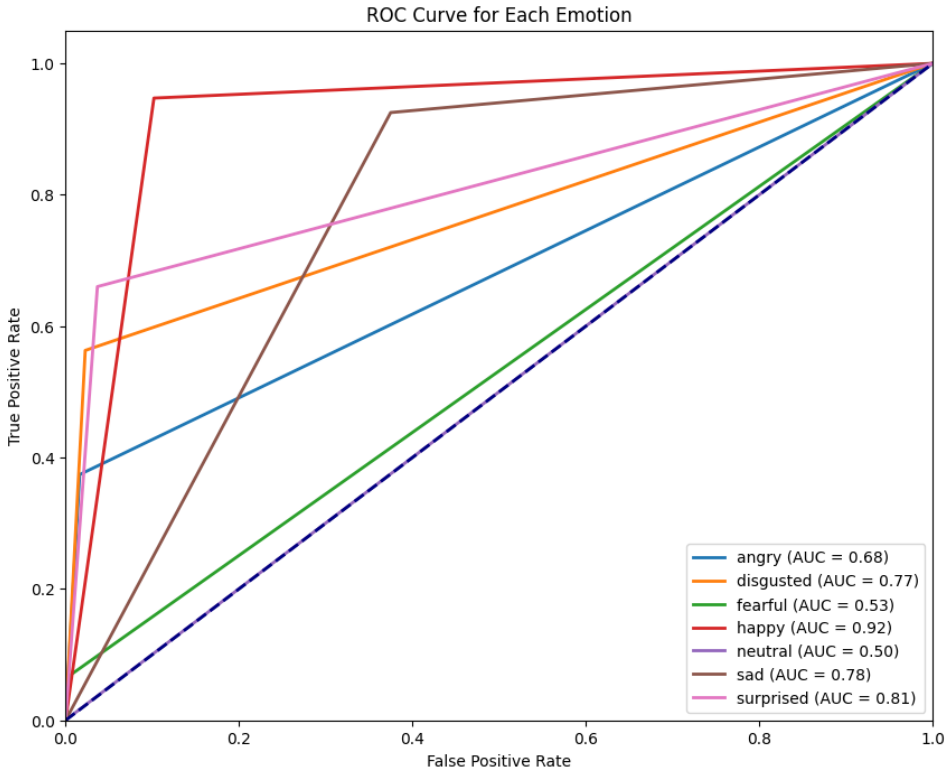
The performance of the LLAVA model on taking all the images of the dataset by giving the prompt to the pre-trained LLAVA model is quite impressive than the ResNetv50. The Comparative analysis is done based on the ROC curve obtained from the Confusion Matrix related to each of the 7 facets of the emotions (angry, disgusted, fearful, happy, neutral, sad, and surprised).

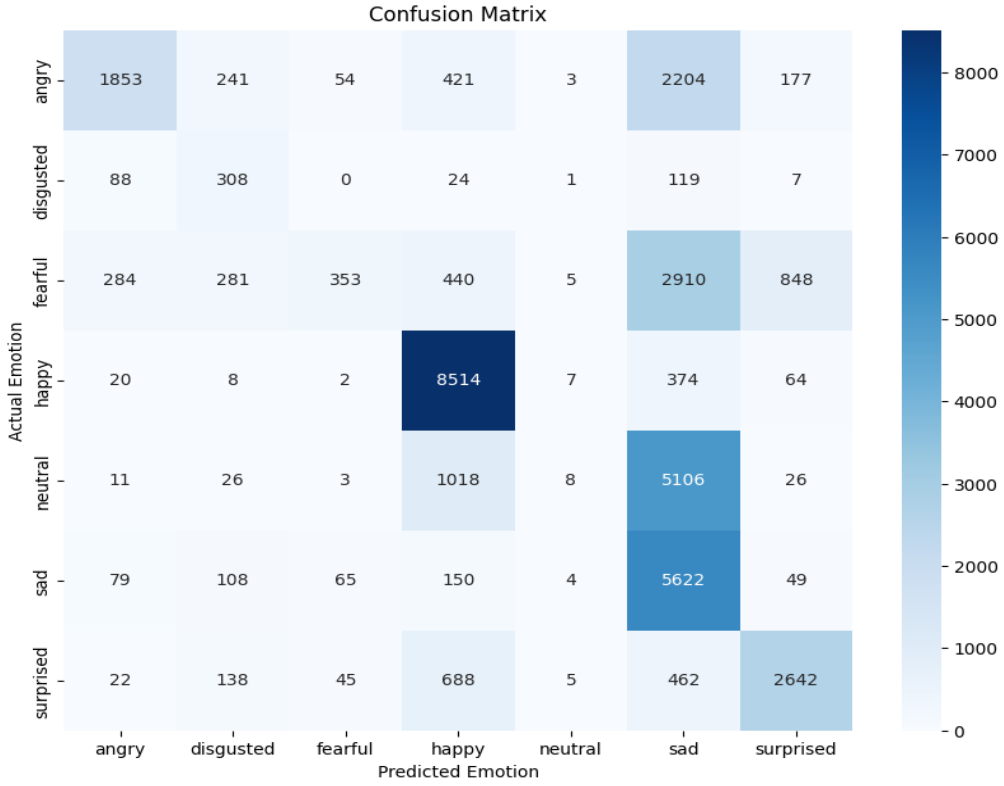
ResNet50v2:





LLAVA model:


ROC Curve for Each Emotion
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
angry (AUC = 0.68)
disgusted (AUC = 0.77)
fearful (AUC = 0.53)
happy (AUC = 0.92)
neutral (AUC = 0.50)
sad (AUC = 0.78)
surprised (AUC = 0.81)


Confusion Matrix
Actual Emotion
Predicted Emotion
angry
disgusted
fearful
happy
neutral
sad
surprised
1853 241 54 421 3 2204 177
88 308 0 24 1 119 7
284 281 353 440 5 2910 848
20 8 2 8514 7 374 64
11 26 3 1018 8 5106 26
79 108 65 150 4 5622 49
22 138 45 688 5 462 2642
8000
7000
6000
5000
4000
3000
2000
1000
0

Waw2vec Mdoel :

The Logistic Regression Model showed promising results when utilizing MFCC Features, achieving respectable classification accuracy and F1-score. Similarly, CNNs demonstrated effectiveness in leveraging MFCC Features. Hence, for autoencoders, only MFCCs are utilized as features.

When it comes to autoencoders, Variational Autoencoders excel in reconstructing audio samples with lower reconstruction loss compared to traditional autoencoders. This superiority stems from their ability to capture the distribution of happy audio samples within a 128-dimensional latent space.

We have taken some standard machine learning models like logistic regression, CNN, autoencoders, and variational encoders to evaluate the performance of different audios of actors of different ages.

The results we got for each machine learning model performances for audio emotion detection are presented here:

Logistic Regression: MFCCs

```
scores, cmatrix = LogisticRegressionPipeline(X1,y1)
```

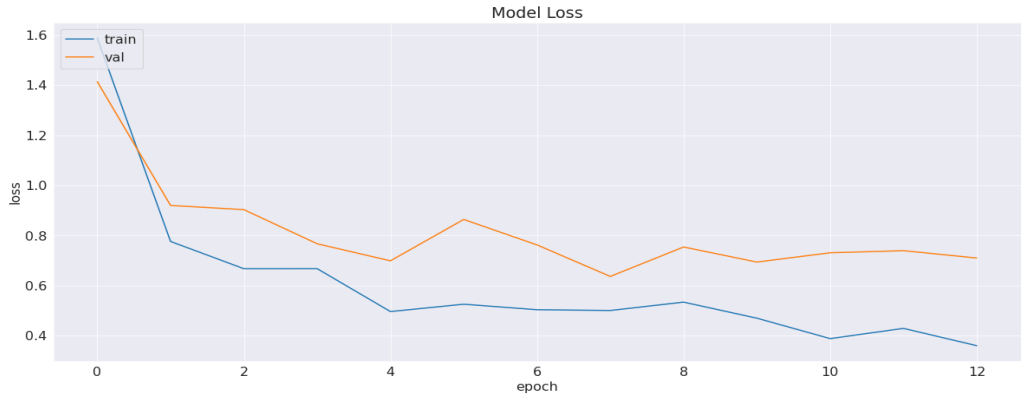
```
Training Performance
              precision    recall  f1-score   support

           0       0.97      0.99      0.98       154
           1       0.99      0.97      0.98       153

    accuracy                           0.98       307
   macro avg       0.98      0.98      0.98       307
weighted avg       0.98      0.98      0.98       307

----------------------------------------
Test Performance
              precision    recall  f1-score   support

           0       0.82      0.87      0.85        38
           1       0.86      0.82      0.84        39

    accuracy                           0.84        77
   macro avg       0.84      0.84      0.84        77
weighted avg       0.85      0.84      0.84        77

----------------------------------------
5-Folds Scores:  [0.64935065 0.74025974 0.67532468 0.68831169 0.80263158]
----------------------------------------
5-Folds Average Score:  0.7111756664388242
```

Logistic Regression: Mel Spectrogram

```
scores, cmatrix = LogisticRegressionPipeline(X2,y2)
```

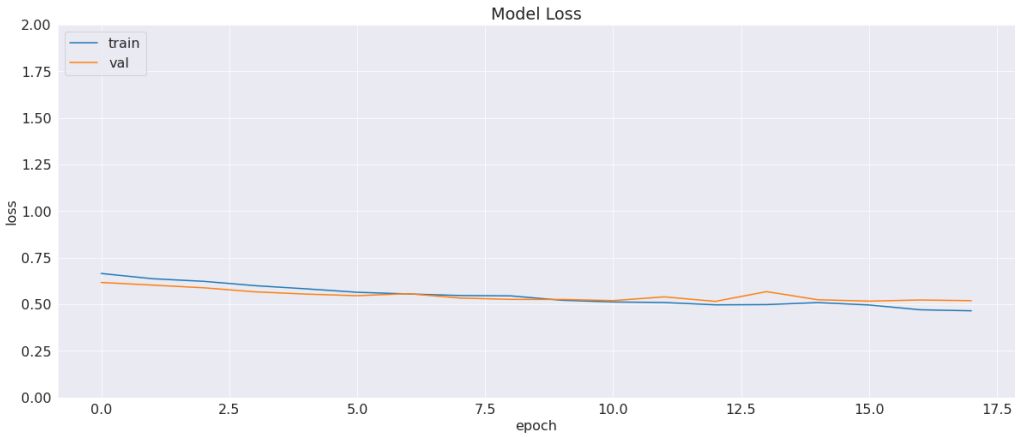
```
Training Performance
              precision    recall  f1-score   support

           0       0.90      0.68      0.77       154
           1       0.74      0.93      0.82       153

    accuracy                           0.80       307
   macro avg       0.82      0.80      0.80       307
weighted avg       0.82      0.80      0.80       307

----------------------------------------
Test Performance
              precision    recall  f1-score   support

           0       0.74      0.61      0.67        38
           1       0.67      0.79      0.73        39

    accuracy                           0.70        77
   macro avg       0.71      0.70      0.70        77
weighted avg       0.71      0.70      0.70        77

----------------------------------------
5-Folds Scores:  [0.67532468 0.74025974 0.67532468 0.72727273 0.71052632]
----------------------------------------
5-Folds Average Score:  0.7057416267942583
```

CNN: MFCCs



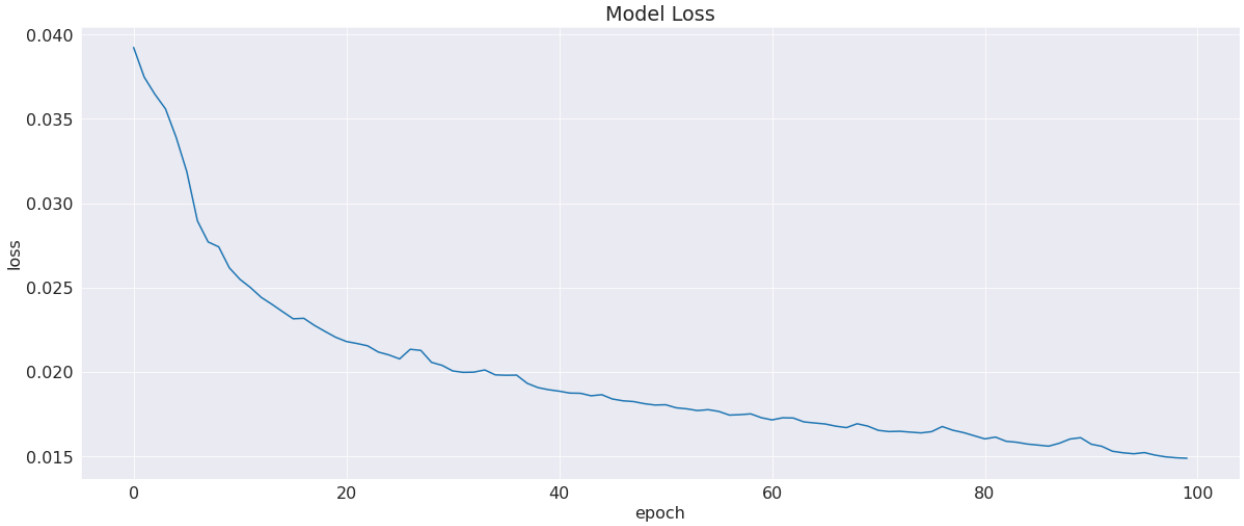

```
              precision    recall  f1-score   support

           0       0.70      0.55      0.62        38
           1       0.64      0.77      0.70        39

    accuracy                           0.66        77
   macro avg       0.67      0.66      0.66        77
weighted avg       0.67      0.66      0.66        77
```

CNN: Mel Spectrogram



	precision	recall	f1-score	support
0	0.53	0.42	0.47	38
1	0.53	0.64	0.58	39
accuracy			0.53	77
macro avg	0.53	0.53	0.53	77
weighted avg	0.53	0.53	0.53	77

Autoencoder:



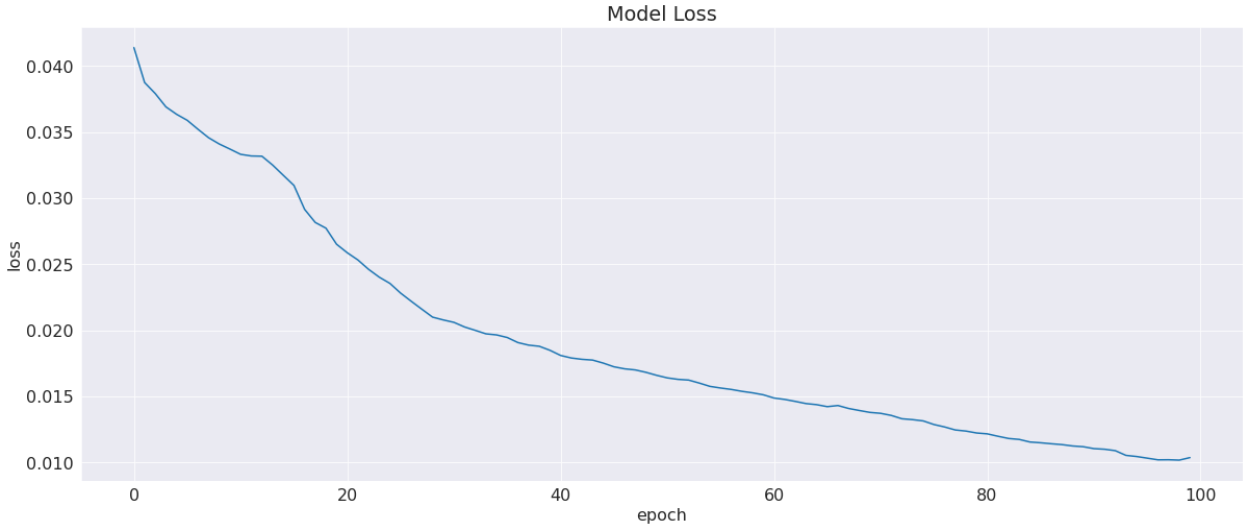

```
model.evaluate(X_happy,X_happy)
```

```
6/6 [==============================] - 0s 4ms/step - loss: 0.0148
0.014842587523162365
```

```
model.evaluate(X_sad,X_sad)
```

```
6/6 [==============================] - 0s 4ms/step - loss: 0.0193
0.019280999898910522
```

Variational Autoencoders:




```
model.evaluate(X_happy,X_happy)
```

```
6/6 [==============================] - 0s 7ms/step - loss: 0.0098
0.009753730148077011
```

```
model.evaluate(X_sad,X_sad)
```

```
6/6 [==============================] - 0s 6ms/step - loss: 0.0235
0.023536866530776024
```

IV. Conclusion

In conclusion, EmotiSense stands as a pioneering endeavor in the realm of emotion detection systems, leveraging the synergistic power of large language models (LLMs) such as BERT, Wav2vec, and LLaVa. By seamlessly integrating textual, auditory, and visual data modalities, our framework transcends the limitations of traditional unimodal approaches, offering a holistic understanding of human emotions in digital interactions.

Our project represents a significant advancement, addressing challenges like data synchronization, model complexity, data imbalance, and interpretability with innovative solutions. Through meticulous development and refinement, EmotiSense not only enhances accuracy and contextual understanding but also fosters genuine human empathy in digital interactions.

Beyond technological innovation, EmotiSense holds promising implications for various domains, including user experience design, mental health support, and customer service. As we continue to optimize the framework, future research endeavors will explore avenues for enhancing model robustness, scalability, and real-time performance.

Crucially, user-centric evaluations and deployment in real-world scenarios will validate the efficacy of EmotiSense, ensuring its seamless integration into diverse applications. Ultimately, our project stands as a testament to the convergence of technology and empathy, heralding a new era

of emotionally intelligent computing that empowers users and fosters deeper human connections in the digital age.

V. Future Work

While EmotiSense represents a significant advancement in multi-modal emotion detection, there are several avenues for future exploration and enhancement:

1. Personalization and Adaptation: Explore methods for personalizing EmotiSense to individual users' unique emotional expressions and preferences, enabling more tailored and effective emotion detection and response.

2. Exploring Additional Modalities: Investigate the integration of additional modalities such as physiological signals (e.g., heart rate variability, skin conductance) and behavioral cues (e.g., gestures, body language) to capture a more comprehensive picture of human emotions.

3.Longitudinal Analysis and Contextual Understanding: Conduct longitudinal studies to analyze changes in emotional states over time and understand how context influences emotional expression and perception. Develop algorithms for contextual emotion detection that consider situational factors, social dynamics, and environmental cues to provide more nuanced and contextually relevant emotion predictions.

4.Cross-cultural Adaptation and Generalization: Investigate methods for adapting EmotiSense to different cultural contexts and linguistic nuances, ensuring that the emotion detection models generalize well across diverse populations and languages. Collaborate with experts in psychology, sociology, and cultural studies to develop culturally sensitive emotion detection frameworks that account for cultural variations in emotional expression and interpretation.