Improving sentiment analysis through use of image and audio data

51126, 38134, 42159, 44766
London School of Economics and Political Science

Abstract

Traditional sentiment analysis typically relies on text alone, an approach which has limitations. In this paper, we explore *multimodal* sentiment analysis, integrating text, audio, and image data to enhance emotion detection accuracy. Our proposed model uses RoBERTa for textual sentiment analysis, a custom CNN for facial emotion recognition from images, and a fine-tuned Wav2Vec2 model for vocal emotion cues from audio. We use a *late fusion* strategy to combine predictions from all all three models based on the length of the input clip. Despite challenges in dataset availability, our experiments demonstrate that multimodal analysis outperforms single-modal approaches in sentiment classification, representing a meaningful step to emotionally intelligent AI.

1 Introduction

For the purposes of this paper, sentiment analysis will be defined as the process of categorising emotions from a given input. This input could take many forms and the categorisation of sentiment can have many uses; for example, identifying sentiment in an online investment forum could inform trading decisions, and identifying the sentiment of a person speaking real time could allow a therapy bot to provide personalised emotional support. When it comes to deep learning, the most common type of sentiment analysis is conducted on text alone; a passage is fed into the model and the model outputs a sentiment score describing said passage. However, sentiment analysis on text alone (known in the literature as sentiment analysis on a single *modality*) has limitations. Consider sarcasm for example; whilst it may be apparent from context clues in a longer text passage (indeed many models can detect it), simply listening to the tone of voice or seeing an eye roll is enough to pick it up when audio or images are provided.

In this paper, we build on the field of *multimodal* sentiment analysis; that is, using multiple *modalities* (text, audio, and image) to more accurately categorise sentiment. We demonstrate how even a relatively simple combination of three modalities can achieve a superior description of sentiment than using one modality alone. Our model takes a video clip, extracts its audio as well as an image corresponding to a facial emotion from the clip. It then uses four separate models to extract meaning, and finally combines them into an output describing the sentiment of the input as a whole. Each model is trained (or has already been pretrained) on a separate dataset, with the application of the full model being demonstrated on an unseen piece of data.

2 Background

2.1 Sentiment Analysis

Each sentiment analysis model we use in this paper outputs a vector transformed by the SoftMax function, which describes the probability an input is a certain emotion. For example, our text sentiment model outputs the following for a given input:

'fear': 0.55889, 'anger': 0.27930, 'neutral': 0.077176, 'surprise': 0.04158, 'disgust': 0.02045, 'sadness': 0.018734, 'joy': 0.003871

A first impression of this output may be that it misses many emotions; for example, where are anxiety, sarcasm and embarrassment? Some research suggests that most emotions are combinations of the fundamental six in the above output (Ekman, 1992) (neutral is also added to represent the lack of emotion). For example, anxiety could be represented by high values for fear, surprise and sadness, whilst sarcasm could be a mix of disgust and joy. For the purposes of this paper we will take it for granted that emotions can be represented as combinations of the fundamental seven in a SoftMax output, however, it should be noted there is no consensus that emotions *actually* work this way. Additionally, there is clear scope for improving the type of sentiment detected by including an 'intensity' label, though due to dataset limitations we were unable to do this.

2.2 Multimodal Sentiment Analysis

Incorporating multiple sources (image, audio and text) together for sentiment analysis is known as 'multimodal sentiment analysis.' In recent years, there has been greater interest in this subject, due to the inherent limitations behind analysing only one 'modality'. The review by Liu et. al (Lai, Hu, Xu, Ren, & Liu, 2023) mostly focuses on the feature unique to multimodal sentiment analysis; the 'fusion' of the different modalities (the modalities being audio, image and content of speech). Many of these fusion models have been extremely sophisticated; for example, RMFN uses RNNs along with an attention mechanism to update each word embedding based on audio and visual data at the time it is spoken. However, we were unable to use this approach as we could not find a suitable dataset that combined audio and visual data for sentiment analysis (one candidate- the MELD dataset- exceeded 10GB and was unable to be loaded on our devices). As a result, we opted to use a 'late fusion' approach to multimodal sentiment analysis.

Late fusion is a strategy where each modality is processed independently using dedicated models, and their predictions are combined at the decision level (Yacine, Amamra, Madi, & Daikh, 2021). A benefit of this approach is that if modalities vary in structure, or some are partially missing, the model can still classify sentiment (whereas a model using an attention mechanism may rely more on a uniform data quality). In our project, we employ a late fusion pipeline on our given input- an audio file and an image corresponding to it, which are taken from a video as whole. We transcribe the text from the audio clip with the Whisper model, pass it through RoBERTa for textual sentiment classification, process the image through a CNN for visual emotion recognition, and analyse speech signals via a fine-tuned Wav2Vec2 model for vocal emotion cues. Many studies then tune the combination of outputs through the use of a neural network. However, due to computational constraints, we decided to manually set the weights of each model based on our intuition of how the modalities would interact. It should be noted this is a limitation of our model, though we still demonstrate how even a relatively simple combination of three modalities yields superior sentiment classification to one alone.

3 Methodology

The broad way our model works is represented by Figure 1. As mentioned in the background section, our combination function has not been fine-tuned based on any multimodal dataset. However, we also opted not to simply average the sentiment scores from the three modalities, for a few reasons. Firstly, the performance of our text sentiment model depends on the length of the audio. For very short sentences, our text transcription model can make mistakes, and it is also more difficult for our model to grasp meaning due to the lack of context (consider the word 'bro' on its own- without intonation, or context, its meaning is likely to be taken as neutral, though it could have a joyful, or even angry sentiment behind it). As a result, for shorter clips, we opted to weight the text sentiment model less than both the audio and image sentiments. Additionally, our CNN model was less accurate than both our audio and text sentiment models- likely as it was trained from scratch. This was especially evident for longer inputs; as we only passed one image from a given video into our CNN, our CNN was unable to pick up possibly changing facial expressions. As such, we weighted image scores less for longer clips. Finally, our audio model consistently performed the best of all 3 models; leading us to weight it the most in the combination function.

The weights we used are represented in the table below.

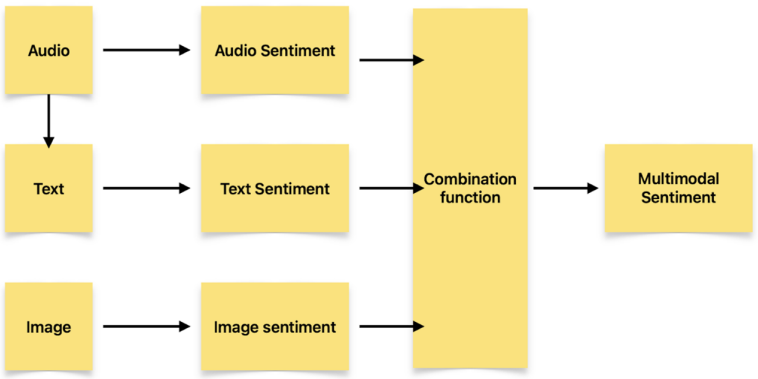

Figure 1: Complete Model

Input Length (s)	Text Weight (t)	Audio Weight (a)	Image Weight
≥ 100	0.333	0.583	0.083
≥ 10 & < 100	$(3-a)/6$	$(1+(s-10)/100)/3$	$(3-a)/6$
> 5 & < 10	0.333	0.333	0.333
≤ 5	$s/15$	$(3-t)/6$	$(3-t)/6$

Table 1: Weights Based on Sentence Length

We now discuss the architecture of the separate models building up our complete model.

3.1 Audio Transcription Model

We utilise the Whisper Tiny model for the audio transcription component, the smallest variant within the Whisper family released by OpenAI (Radford et al., 2022). Whisper is based on a transformer encoder-decoder architecture and was trained on 680,000 hours of multilingual and multitask supervised data. The Tiny model consists of approximately 39 million parameters, with four encoder layers, four decoder layers, a hidden dimension size of 384, and 6 attention heads. Whisper was particularly good at longer inputs (it can transcribe a maximum audio length of 30 seconds), but did not perform as well for shorter inputs, leading us to try and fine tune it to better pick up the words spoken in shorter audio clips (we explore fine tuning in a later section).

3.2 Text Sentiment Model

The model we use for text sentiment analysis (applied on the transcription from the audio) is RoBERTa, a fine tuned version of the BERT model initially developed by Google (Devlin & Chang, 2018). BERT uses transformer architecture to perform sentiment analysis, which works by updating word embeddings (the map of tokens to numerical vectors representing sentiment) through self attention heads that take into account the context around words. For example, the word 'like' may initially be mapped to a vector with a relatively high value for the emotion joy. However, after applying the attention mechanism to the sentence around the word 'like', which could be 'I don't like pizza', the word 'like' is then mapped to a vector that more accurately reflects its negative sentiment. RoBERTa has been deeply fine tuned to pick up long range dependencies in text, as well as more nuanced sentiments, such as sarcasm. From testing the model on a variety of inputs, we found this to be true, and were consistently impressed with its output. As such, we did not believe it required more fine tuning to serve as a good baseline for comparison between multimodal and text sentiment analysis.

3.3 Facial Expression Model

Our facial emotion detection model takes in a 48 by 48 grayscale image of a facial expression from a video and outputs a SoftMax output that represents the overall emotion. The model architecture

we utilise to pick up facial emotions is a Convolutional Neural Network (CNN). CNNs differ from basic Feed Forward Networks (FFNs) through the use of a kernel (also known as a filter). The kernel is a matrix that passes over the input image and performs a convolution operation, represented by the equation below.

$$S(i,j) = (I * K)(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i+m, j+n) \cdot K(m,n) \tag{1}$$

Where *I(i, j)* is the the pixel value at position *(i,j)* in the input image, *K(m, n)*, is the weight of the kernel at position (*m,n)*, and *S(i,j)* is the output feature map at position *(i,j)*.

After applying the convolution operation, the result is typically a smaller matrix of values than the input image, though in our paper, the result of the convolution is the same size as the input image. This is achieved through a padding of 1-pixel around the input, as well as a stride of size 1. We opted not to downsample with our convolutions, as the input image is not very large and our network is deep, so downsampling risks losing too much information after each convolution.

We then pass the output from our convolution operation through a batch normalisation function, that normalises the outputs of each convolution to have a mean of 0 and a standard deviation of 1. This is to speed up training (though the exact way 'batchnorm' achieves this is disputed). After applying batchnorm we then pass those values through a ReLU activation function, which introduces non-linearity to the network and has been argued to be the best activation function for training of deeper networks (Glorot, Bordes, & Bengio, 2011). In our model we then apply another convolution and another ReLU function, in order to pick up the maximum detail possible.

After the ReLU function is applied on all inputs, the resulting output, known as a feature map, is then subject to a pooling operation, which outputs a smaller matrix than the feature map. We specifically use 'max pooling' in our paper, which replaces each region of the feature map with its maximum value, in order to retain the most prominent features of the map. Another option had been average pooling, though ever since the release of AlexNet, the consensus seems to be that max pooling works better for CNNs (Fergus & Zeiler, 2013).

For our model we repeat the pattern of Conv-Batchnorm-ReLU-Conv-ReLU-MaxPool 3 times, leading to a total of 18 layers for the first part of our CNN (there are more layers in the later FFN). Additionally, we use a total of 256 kernels to detect emotion. Facial expressions are quite complex and so we did not worry too much about overfitting, though we discuss our approach to it more in the training section of the paper.

Finally, after the 'convolutional' part of the network, the resulting output is passed into a FFN for classification, with outputs normalised on a scale of 0 to 1 via the SoftMax operation (which is the same shape as our text and audio sentiment output, allowing for easier combination). In total, the CNN we trained has 23 layers. However, despite the heavy training, the facial emotion detection model was the weakest out of all our models. We discuss why this is the case in the training section of our paper.

3.4 Voice Sentiment Analysis

Speech conveys rich emotional signals - such as tone, pitch, and pace - that are often absent in text. These features can be visualised using spectrograms or waveform graphs, revealing temporal patterns linked to emotional expression.

To capture these features, we use the Wav2Vec2 model (Baevski, Schneider, & Auli, 2019), a self-supervised framework for speech representation learning. Specifically, we fine-tune the pre-trained `wav2vec2-base` model from Hugging Face to classify emotions from raw audio clips.

As shown in Figure 2,Wav2Vec2 comprises a two-stage architecture: (1) a CNN-based feature encoder that downsamples raw waveforms into a sequence of latent feature vectors (approximately one every 20 ms), and (2) a transformer encoder that processes these representations to produce contextualised embeddings. For example, one second of audio yields about 50 hidden states, each summarising roughly 25 ms of speech, resulting in an output tensor of shape (T, 768), where T is the number of time steps.

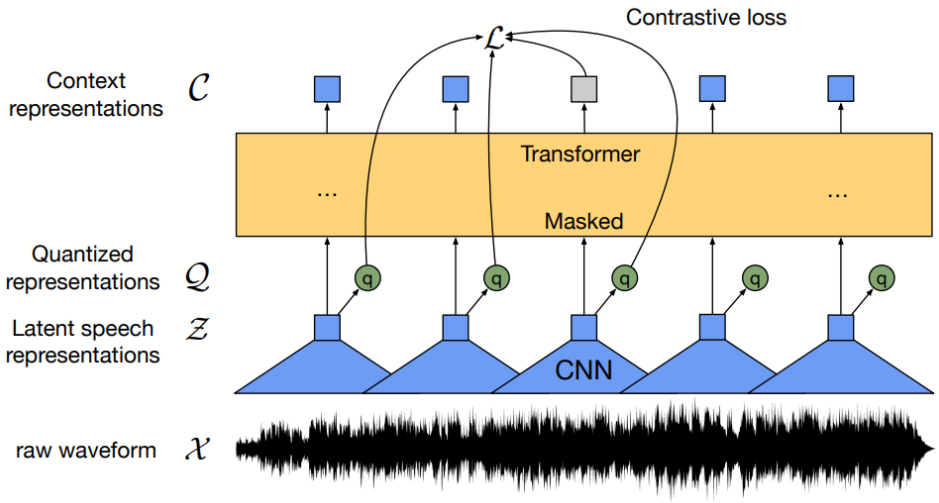

Figure 2: Overview of the Wav2Vec2 architecture. The raw waveform is passed through a CNN encoder and a Transformer, with masked predictions trained using contrastive loss.

We adopt Connectionist Temporal Classification (CTC) as the loss function during training. CTC is widely used in speech recognition tasks involving RNNs or LSTM models, where the input and output sequences are of differing lengths and alignment is unknown. It enables models to learn a probabilistic mapping between variable-length audio inputs and label sequences (e.g., characters or phonemes), without requiring explicit frame-level alignment. In Wav2Vec2, CTC allows us to directly map transformer outputs to character-level or class-level predictions via a softmax layer.

After fine-tuning(which we will discuss more in-depth in the next section), the model outputs a probability distribution over seven emotion categories for each audio sample. These predictions are later combined with outputs from the image and text branches for our complete model output. As we felt our audio model was the best of all three, we decided to weight its output the most.

4 Training and Hyperparameter Tuning

4.1 CNN Tuning and Training

Our CNN was trained from scratch to classify grayscale facial expressions from the FER-2013 dataset into seven emotional categories: angry, disgusted, fearful, happy, neutral, sad, and surprised. The FER-2013 dataset comprises 35,887 grayscale 48×48 pixel images and is inherently imbalanced, with some classes (e.g., ”happy” and ”neutral”) being overrepresented compared to others like ”disgusted”. This led our model to be less likely to predict an expression as being disgust relative to other emotions, which weakened our initial assumption that a SoftMax output could be used as a proxy for more complex emotions. This was also part of the reason behind why we weighted our CNN the least of the three sentiment models (for longer text inputs).

To improve spatial feature extraction, we resized the images to 64×64 and normalized pixel values to have a mean and standard deviation of 0.5. For training, we used a batch size of 32 and a batch size of 64 for evaluation. This distinction was intentional: smaller batches during training helped reduce GPU memory usage and improve convergence stability, while larger batches during evaluation allowed for faster inference, as gradient computation and backpropagation were not required. As mentioned earlier in the methodology section, our model is deep (23 total layers) and one method we used to address the risk of overfitting was including dropout rates of 0.5 and 0.4 in the FFN part of our CNN.

We trained the model using the AdamW optimizer, which incorporates decoupled weight decay to improve generalization. Initially, we experimented with a learning rate of 0.05, but this led to unstable training. Through the use of random search (a practical alternative to grid search given

computational constraints), we sampled values across a logarithmic scale and identified 0.001 as an optimal learning rate. Similarly, we tested several values for weight decay (0.01, 0.001, 0.0001) and found that 0.0001 offered the best regularization effect without degrading performance. Dropout rates were also tuned empirically, balancing regularization and model capacity. We further improved generalization with label smoothing (factor = 0.1) and used a cosine annealing learning rate scheduler to allow the optimizer to converge more smoothly.

To evaluate model performance, we tracked training loss and test accuracy over 25 epochs, plotted learning curves, generated a classification report, and constructed a confusion matrix to inspect error patterns. As expected, the model struggled more with underrepresented or visually similar categories (e.g., "disgusted" vs. "angry" and "sad" vs. "neutral"), highlighting challenges inherent not only in the dataset's class imbalance but also in the subtle nature of facial expressions, where some emotions may appear quite similar even to human observers. Nonetheless, tuning the hyperparameters helped the model achieve a final test accuracy of 62.2% which was a significant improvement over our baseline model, which consisted of a single convolutional layer and achieved just 40.9%. This accuracy placed our model among the top-performing non-pretrained CNNs for this dataset on Kaggle.

However, since the model only took in a single image from a video, along with the superior performance of the pretrained audio and text sentiment models (better than 62.2% accuracy), we weighted our image classification model the least of all three on inputs longer than 10 words.

4.2 Fine Tuning of Whisper Tiny Model (Audio Transcription)

We attempted to fine tune Whisper on the Toronto Emotional Speech Set (TESS) to better pick up short utterances. Our fine tuning attempts led to a 'word error rate' of just above 1%, however, after then testing the Whisper model on a longer audio input, the model performed worse than *before* fine tuning. As such, we decided to just use the pretrained model prior to fine tuning, accepting that the model would underperform on short utterances.

4.3 Fine Tuning Wav2Vec2 Model

We fine-tuned the `wav2vec2-base` model from Hugging Face, originally developed for self-supervised speech representation learning, to perform speech emotion classification on the TESS dataset. The raw audio files were first extracted and relabelled to include the full emotion class name. Each file was associated with an emotion label parsed from its filename using regular expressions and mapped to one of seven emotion categories.

The preprocessed data was split into training and test sets using an 80/20 split. Audio signals were visualized using waveforms and spectrograms to confirm clarity and emotion distribution. For training, we implemented a custom PyTorch dataset class that loads, resamples, and tokenizes each audio file using the `Wav2Vec2Processor`. Audio was resampled to 16 kHz and padded or truncated to a maximum length of 1 second. Labels were mapped to integer indices using a fixed label-to-id mapping.

We initialised the `Wav2Vec2ForSequenceClassification` model with `num_labels=7`, and trained it using Hugging Face's `Trainer` API with its learning rate scheduler. One benefit of its use of a dynamic learning rate to improve convergence is that we did not need to tune it manually (the different learning rates tested are mentioned in the table below). The training configuration included a batch size of 16, 3 epochs, weight decay of 0.01, and evaluation every 50 steps. Logging and checkpointing were enabled for progress tracking and reproducibility. Evaluation metrics included accuracy, precision, recall, and F1 score, computed using `scikit-learn`.

Before full training, the model was evaluated on the test set to establish a baseline. Throughout training, we recorded training loss and evaluation metrics, which were summarized in Table **??**. As shown in the table, the training loss consistently decreased from 1.66 at step 50 to 0.09 at step 400, demonstrating smooth and stable convergence. Final evaluation on the held-out test set yielded a classification accuracy of **99.8%** and an evaluation loss of 0.0308, indicating excellent generalization. These results confirm that the fine-tuned Wav2Vec2 model is highly effective at recognizing emotional cues in speech and serves as a strong auditory branch in our multimodal sentiment classification system.

	loss	grad_norm	learning_rate	epoch	step
0	1.6571	5.338749	0.000044	0.357143	50
1	0.9138	8.284115	0.000038	0.714286	100
2	0.4912	6.435124	0.000032	1.071429	150
3	0.3156	3.719413	0.000026	1.428571	200
4	0.1999	2.049409	0.000020	1.785714	250
5	0.1081	32.543797	0.000014	2.142857	300
6	0.1110	0.229513	0.000008	2.500000	350
7	0.0904	0.298211	0.000003	2.857143	400

Table 2: Training loss across steps and epochs during fine-tuning of Wav2Vec2 on TESS.

5 Results

We tested our complete model on 2 pieces of (unseen) data. The first was a clip from Breaking Bad (*I am the one who knocks*), whilst the second was a clip from the show Invincible (*I am so lonely*). Both data were emotional pieces of dialogue; the Breaking Bad clip is a clearly angry and spiteful tone, whereas the Invincible clip is more ambiguous; we felt it was mostly sad, but also with hints of fear and disgust as well.

The speech transcription of the Breaking Bad clip produced the following results:

'You clearly don't know who you're talking to. So let me clue you in. I am not in danger, Skyler. I am the danger. A guy opens his door and gets shot and you think of that of me. Now I am the one who knocks.'

The one typo here was that 'Now', should actually be 'No.' Additionally, the transcription does not include possible punctuation that could help the text model grasp meaning, most evidently there should be a question mark after 'you think that of me.' Our text sentiment model (RoBERTa) then produced the following output: (rounded to 3dp)

'fear': 0.559, 'anger': 0.279, 'neutral': 0.077, 'surprise': 0.042, 'disgust': 0.020, 'sadness': 0.019, 'joy': 0.004

- which weighted fear too much. This demonstrated how using a single modality can be subject to errors. Our facial emotion model then took in an image of Walter (the speaker in the Breaking Bad clip) at a particularly emotional moment. It produced the following output:

'anger': 0.950, ' 'disgust': 0.003, 'fear': 0.003, 'joy': 0.027, 'neutral': 0.010, 'sadness': 0.004

- which appropriately weighted anger as being the strongest emotion. Finally, our audio sentiment model produced the following output:

'anger': 0.737, 'disgust': 0.026, 'fear': 0.056, 'joy': 0.019, 'neutral': 0.021, 'sadness': 0.048, 'surprise': 0.093

Again, detecting anger as the most prominent emotion. The predictions of emotion by each modality are represented in Figure 3. It demonstrates a key strength of multimodal sentiment analysis; when one modality 'goes wrong' the others can correct it.

The Invincible clip's predictions are represented in Figure 4.This was interesting to us, as it demonstrated how all 3 modalities picked up *different* aspects of emotion, all of which could be argued to be the 'true emotion' to some extent. The final output is ambiguous about the 'correct' emotion, but this aligns with our own ambiguous feelings about the emotion of the clip. In a way, using multimodal sentiment analysis prevented our model from being too confident about a certain emotion being the 'true emotion'.

Taken together, the model demonstrated the power of multimodal sentiment analysis, even with a basic combination function. However, it should be noted that we only tested it on two examples and there may be some examples for which the model performs poorly. A clear extension of this project would be to fine tune the combination function based on a multimodal sentiment dataset; though we still feel our relatively crude function demonstrates the benefits of using multiple modalities for sentiment analysis.

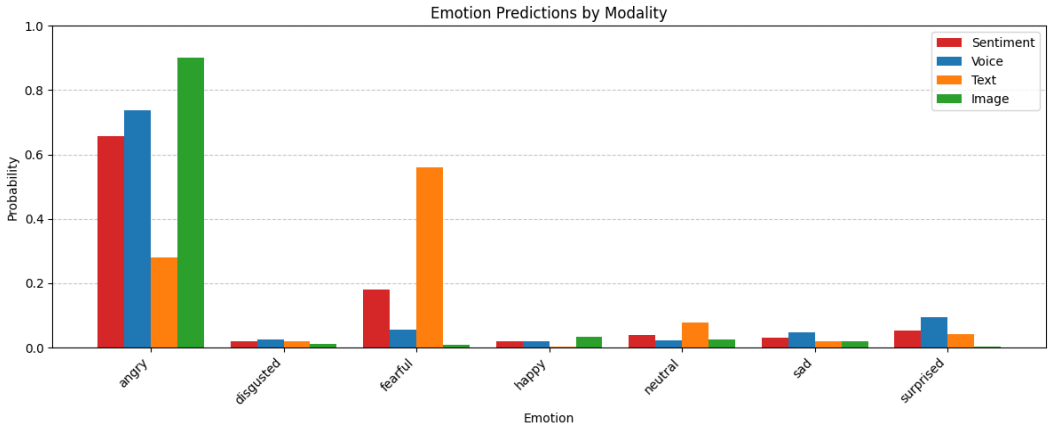

Figure 3: Sentiment analysis on Breaking Bad Clip. Although the text output is slightly wrong it is corrected by the other modalities

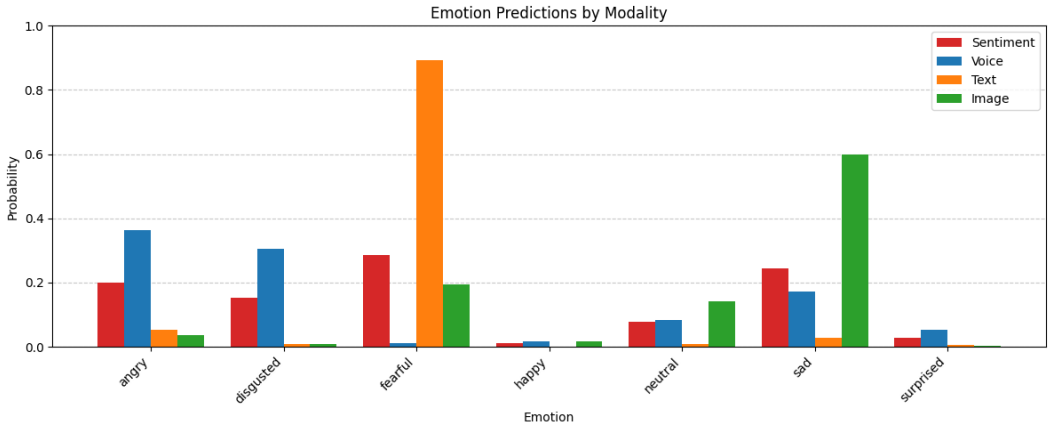

Figure 4: Invincible Clip Sentiment Results

6 Conclusion

This project demonstrates the power and potential of multimodal sentiment analysis by combining three distinct modalities - text, audio, and image, into a unified framework for emotion detection. Through the integration of RoBERTa for textual sentiment, Wav2Vec2 for vocal emotion cues, and a CNN for facial expression recognition, we show that even a simple late fusion strategy can outperform single-modality models, particularly in emotionally complex or ambiguous scenarios.

The importance of this work lies in its ability to move beyond the limitations of text-only sentiment analysis, addressing real-world cases where emotion is conveyed through tone and expression just as much as through words. Our model's success in detecting nuanced emotions like anger or sadness across various clips reinforces the strength of this approach, and it highlights the robustness that arises when multiple modalities can compensate for each other's weaknesses.

Nevertheless, limitations remain. Our use of a SoftMax output of 7 key emotions may miss certain emotions like 'determination', which could require an intensity label to fully convey. Additionally, we were constrained to 30-second audio inputs, meaning our analysis does not account for emotional shifts over longer periods, something future work could explore by tracking evolving sentiments in longer conversations or videos. The use of fixed weights in our fusion function, rather than a learned combination, also limits adaptability, though it serves as a proof of concept that multimodal integration meaningfully improves performance.

Looking forward, the project could be extended in several exciting directions: training the fusion model on a large-scale multimodal dataset, incorporating additional modalities such as physiological signals, or developing real-time sentiment tracking in dynamic environments. Ultimately, this project represents a small step towards emotionally intelligent AI.

7 References

Patrick von Platen. wav2vec2-base. https://huggingface.co/patrickvonplaten/ wav2vec2-base. Accessed: 2025-05-03.

Baevski, A., Schneider, S., & Auli, M. (2019, October). vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations.

Devlin, J., & Chang, M.-W. (2018, November 2). Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing. Retrieved from Google Research: https://research.google/blog/open-sourcing-bert-state-of-the-art-pre-training-for-natural-language-processing/

Ekman, P. (1992). Facial Expressions of Emotion: New Findings, New Questions. Psychological Science, 34-38.

Fergus, R., & Zeiler, M. (2013). Visualizing and Understanding Convolutional Networks. Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. AISTATS.

Lai, S., Hu, X., Xu, H., Ren, Z., & Liu, Z. (2023). Multimodal sentiment analysis: A survey. Displays.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022, December). Robust Speech Recognition via Large-Scale Weak Supervision.

Yacine, S., Amamra, A., Madi, M., & Daikh, S. (2021). Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition. Machine Vision and Applications.