Voice Emotion Classification using Reprogrammed GPT-2 with Patch Embeddings

Internship Project Report
Submitted in partial fulfilment of the requirements for the award of the degree of

Master of Technology in Artificial Intelligence

Submitted by

Aijaz Mustafa
Roll No: 2411AI64

Under the Guidance of

Professor Jimson Mathew

Department of Computer Science and Engineering
Indian Institute of Technology
July, 2025

Acknowledgment

First, I would like to express my sincere gratitude to my project supervisor, **Professor Jimson Mathew**, for their continuous guidance, insightful feedback, and unwavering support throughout the course of this project. Their expertise and encouragement were instrumental in shaping the direction and success of this work.

I am also very grateful to the PhD research scholar **Naseem Babu** for providing a conducive environment and valuable resources that enabled the completion of this project. A special thanks goes to my fellow batchmates and friends for their collaboration, discussions, and moral support during the development and testing phases. Last, I am deeply grateful to my family for their patience, motivation, and belief in my work throughout my academic journey. This project would not have been possible without the collective support and encouragement of all those mentioned above.

Abstract

Human emotions play a pivotal role in communication and interaction, making automatic emotion recognition from speech a valuable task in human-computer interaction, mental health monitoring, and virtual assistant technologies. Traditional voice emotion recognition systems rely heavily on handcrafted features and shallow classifiers, often failing to generalise across varied speech patterns and emotions. This project introduces a novel deep learning approach that reprograms a pre-trained Generative Pre-trained Transformer 2 (GPT-2) [1], initially developed for natural language processing, for voice emotion classification.

Our method transforms speech signals into time-frequency representations (e.g., MFCC), followed by patch embedding and a reprogramming layer to make the audio-compatible inputs suitable for GPT-2. The model learns complex emotional patterns embedded in speech by leveraging the robust contextual understanding of the transformer architecture. Experiments were conducted on standard datasets such as TESS [2], and the proposed model was evaluated using metrics such as accuracy, precision, recall, and F1-score.

Contents

Chapter 1

Introduction

Emotions are central to human communication and influence how we convey and interpret information. Recognising emotions from voice, known as Voice Emotion Classification(VEC), has emerged as a crucial task in human-computer interaction, with applications in virtual assistants, mental health diagnostics, and adaptive learning systems. Traditional emotion recognition systems rely on handcrafted acoustic features combined with classical classifiers like Support Vector Machines or Hidden Markov Models. While these methods offer moderate performance, they struggle to capture the nuanced, hierarchical patterns inherent in emotional speech, especially across speakers and languages.

Recent advances in deep learning have enabled automatic feature extraction through architectures like Convolutional and Recurrent Neural Networks. However, these models often require large domain-specific datasets and still fall short in capturing long-range dependencies and contextual nuances. This project introduces a novel framework that reprograms a pre-trained GPT-2 language model, initially designed for textual data, to classify emotions from speech. Audio signals are converted into MFCC [3] and processed through patch embedding and reprogramming layers, enabling compatibility with the GPT-2 architecture. This allows the model to leverage the transformers' contextual depth and generalisation power for the audio domain. The primary objective is to develop and evaluate a deep learning pipeline that uses GPT-2 in a reprogrammed fashion for accurate voice emotion classification. Experimental results demonstrate significant improvements over traditional models, highlighting the potential of cross-domain transformer re-usability.

1.1 Large Language Models (LLMs) for Audio-based Emotion Classification

Large Language Models (LLMs) are deep neural architectures designed initially for natural language understanding and generation [4, 5, 6]. These models are typically built on the transformer architecture, introduced by Vaswani et al. [7], which leverages a self-attention mechanism to capture local and global dependencies across sequences.

Unlike recurrent neural networks, transformers are highly parallelizable and scalable, making them well-suited for handling long sequences. Two primary paradigms dominate LLM training:

1. **Masked Language Modelling (MLM):** Used in models like BERT [8, 9], where random tokens are masked and the model learns to predict them using surrounding

context. This method is effective for classification and understanding tasks.

2. **Autoregressive Language Modelling (ALM):** Employed by models like GPT [8], where the model learns to predict the next token in a left-to-right manner, excelling in generative tasks such as text generation, dialogue systems, and sequential modelling.

Although LLMs were initially built for text data, their modular and generalizable architecture has enabled their application in non-text domains. This project adapts a GPT-2-based autoregressive language model for speech-based emotion recognition. Since LLMs are not inherently designed for audio signals, we introduce a patch embedding and reprogramming pipeline to convert Mel-frequency cepstral coefficients (MFCCs) into a format compatible with the transformer input space.

Recent advancements have seen LLMs evolve into *multimodal models* (e.g., Flamingo, LLaVA, GPT-4V) [10, 11], capable of processing and aligning across modalities such as text, images, and audio. These developments have opened the door to innovative uses of LLMs in tasks such as speech recognition, audio classification, and emotion decoding.

Using the representational power of LLMs and applying a domain-bridging reprogramming approach, we extend their utility to the domain of affective computing. This enables efficient learning from limited emotional speech data and offers potential for applications in real-time emotion-aware systems, assistive technology, and human-computer interaction. These developments have opened the door to innovative uses of LLMs in tasks like speech recognition, audio classification, and emotion decoding. By leveraging the representational power of LLMs and applying a domain-bridging reprogramming approach, we extend their utility to the domain of affective computing. This enables efficient learning from limited emotional speech data and offers potential for applications in real-time emotion-aware systems, assistive technology, and human-computer interaction [12].

Chapter 2

Literature Review

Voice emotion recognition has emerged as a significant area of research, bridging the domains of speech processing, affective computing, and artificial intelligence. Over the years, the approaches to voice emotion recognition have evolved significantly, ranging from traditional machine learning methods to more sophisticated deep learning and transformer-based architectures. More recently, cross-domain model reprogramming and multimodal adaptation techniques have opened new frontiers, particularly involving large language models (LLMs).

Traditional Approaches

Early systems primarily relied on handcrafted features such as Mel-Frequency Cepstral Coefficients (MFCCs), pitch, energy, and formants to capture speech's prosodic and spectral characteristics. These features were then fed into classical classifiers, including Support Vector Machines (SVM) [13], Hidden Markov Models (HMM) [14], and k-Nearest Neighbors (k-NN) [15]. While computationally efficient, these methods often struggled with speaker variability, noise sensitivity, and limited generalizability across datasets.

Deep Learning-Based Methods

With the rise of deep learning, models such as Convolutional Neural Networks (CNNs) [16] and Recurrent Neural Networks (RNNs) [17] became dominant. CNNs proved effective in learning spatial features from spectrograms and MFCC representations, while RNNs and their gated variants like LSTMs (Long Short-Term Memory) [18] excelled in modeling temporal dependencies in sequential audio data. Hybrid architectures, combining CNNs for spatial encoding and RNNs for sequence modeling, further improved performance. However, these models often required large, well-annotated datasets and struggled to generalize to unseen domains.

Transformer Models in Speech

The success of transformers in natural language processing (e.g, GPT-2)[1] prompted researchers to explore their applicability in speech tasks. Self-attention mechanisms [19] allow transformers to model long-range dependencies more effectively than RNNs. Approaches such as wav2vec, HuBERT, and Speech-BERT have shown promise in speech recognition and emotion detection. However, these models are typically pre-trained on large-scale audio corpora, making them resource-intensive.

Cross-Domain Reprogramming

An emerging paradigm involves reusing language models like GPT-2 for non-text domains through input reprogramming. This technique involves adapting the input space, such as vision patches or audio features, so that the model can process and learn from modalities it wasn't originally trained on. Research has shown that reprogramming can unlock powerful transfer capabilities without modifying the model weights, allowing existing NLP models to generalize to vision and speech tasks.

Research Gap and Our Contribution

Research Gap:

- While transformer architectures have achieved significant success in natural language processing (NLP) [20], their adaptation for voice emotion recognition (VER) tasks remains underexplored.
- There is a lack of research on reprogramming large-scale pre-trained language models, such as GPT-2, for processing and classifying non-textual audio data.
- Existing emotion classification approaches often require task-specific architectures and extensive training, which can be resource-intensive and less flexible.

Our Contribution:

- We propose a novel methodology that repurposes GPT-2 for voice emotion classification using a reprogramming-based pipeline.
- The raw audio signal is first converted into its Mel-Frequency Cepstral Coefficient (MFCC) representation. This creates a 2D feature matrix, which is then divided into a sequence of fixed-size spatial patches. Finally, each patch is flattened and linearly projected to create a token embedding, forming a sequence that serves as the input for the GPT-2 model.
- A reprogramming layer is introduced to map patch embeddings into GPT-2's token space, enabling compatibility without modifying GPT-2's core architecture.
- Our approach leverages the strengths of pre-trained transformer models while reducing training costs and data dependency.
- This work demonstrates the feasibility and effectiveness of using LLMs [21] in non-textual domains, bridging the gap between NLP and audio signal understanding.

Chapter 3

Dataset Description

This project utilizes the Toronto Emotional Speech Set (TESS) [2], a well-known dataset for emotion recognition in speech.

Dataset Overview

The TESS dataset contains recordings of two professional actresses (aged 26 and 64) who read a fixed set of 200 target words in the carrier phrase "Say the word" across seven different emotional categories. The dataset is modeled after the Northwestern University Auditory Test No. 6 (NU-6).

Emotion Classes

The dataset includes the following 7 emotions: Angry, Disgust, Fear, Happy, Neutral, Pleasant Surprise, and Sad.

3. Data Format

The dataset consists of WAV files sampled at 24kHz, with each sample lasting approximately 1-2 seconds. There are around 2,800 samples in total.

Chapter 4

Methodology

This project proposes a novel framework for voice emotion classification using the Generative Pre-trained Transformer 2 (GPT-2), a large-scale language model. Unlike traditional approaches that rely on domain-specific architectures, our method leverages the powerful representation capabilities of GPT-2 through a reprogramming strategy and patch-based audio embeddings. The pipeline transforms raw audio signals into a format compatible with GPT-2 while maintaining the model's contextual reasoning ability.

System Overview

The methodology consists of four main components:

1. **Audio Preprocessing and Feature Extraction**
2. **MFCC Feature Extraction**
3. **Patch Embedding Layer**
4. **Reprogramming Layer**
5. **GPT-2-based Classification Head**

Each component plays a crucial role in adapting speech signals into a sequence of embeddings interpretable by a language model. The following sections detail each stage of the pipeline.

1 Audio Preprocessing and Feature Extraction

Before feeding audio data into the model, several preprocessing steps were applied to ensure consistency and enhance the quality of extracted features:

- **Resampling:** All audio clips were resampled to a standard sampling rate of 16 kHz to maintain uniformity across the dataset.
- **Mono Conversion:** TESS audio is already mono.
- **Noise Reduction:** TESS is clean studio-quality speech.

2 MFCC Feature Extraction

MFCCs (Mel Frequency Cepstral Coefficients) were extracted to represent the emotional content of speech signals. We extracted **Mel-Frequency Cepstral Coefficients (MFCCs)**, a widely used feature in speech processing tasks.

- MFCCs capture the power spectrum of the audio signal using the Mel scale, which aligns closely with the human auditory system. The process includes windowing, Fourier transformation, Mel-scale filterbanks, logarithmic compression, and Discrete Cosine Transform (DCT).
- Typically, 40 MFCCs were extracted per frame to represent the speech signal. These coefficients capture the spectral envelope of the audio and are widely used in emotion recognition due to their perceptual relevance. These coefficients were then aggregated across time steps to form fixed-size feature vectors for training the emotion classification model.

This robust pre-processing and MFCC-based feature extraction combination ensures that the model receives emotionally relevant and acoustically rich representations of speech signals.

3 Patch Embedding Layer

To enable transformer-based architectures like GPT-2 to process MFCC features, we introduce a Patch Embedding layer that tokenises the MFCC input sequence into patch-wise representations.

The MFCC feature matrix, typically of shape $T \times F$, where T is the number of time frames and F is the number of MFCC coefficients (e.g., 40), is segmented along the temporal dimension into fixed-size patches. Each patch comprises a block of consecutive frames and is flattened into a 1D vector.

Each of these flattened patch vectors $x_i \in \mathbb{R}^{F \cdot p}$ (where p is the number of frames per patch) is then projected into a lower-dimensional embedding space through a learnable linear layer:

$$z_i = W_p \cdot \text{Flatten}(x_i) + b_p$$

Here, x_i denotes the i^{th} MFCC patch, which is a matrix of shape $p \times F$. $W_p \in \mathbb{R}^{d \times (F \cdot p)}$ represents the learnable weight matrix, and $b_p \in \mathbb{R}^d$ is the learnable bias vector. The variable d refers to the desired embedding dimension compatible with GPT-2's input layer.

This process converts the MFCC feature matrix into a sequential embedding:

$$\{z_1, z_2, \ldots, z_n\}$$

Then it passes through the reprogramming and transformer layers. This design bridges the gap between audio signal representations and language models, treating temporal acoustic segments as pseudo-text tokens.

4. Reprogramming Layer

Since GPT-2 was initially trained on natural language tokens, we introduce a reprogramming layer that maps the audio patch embeddings into GPT-2's expected input space.

Projection: A trainable adapter projects the audio embeddings to the same dimensionality as the GPT-2 token embeddings (e.g. 768 for 'gpt2'). The final transformed sequence is passed to GPT-2 as a regular language input.

5. GPT-2 Transformer Model

GPT-2 processes the reprogrammed input sequence using its original transformer architecture, with all weights left unfrozen to allow end-to-end fine-tuning for emotion classification. In our implementation, we employ the pre-trained GPT-2 base model ('gpt2'), which comprises 12 transformer layers, 12 attention heads, and a hidden size 768. After the model processes the input, the output embeddings from all tokens are aggregated, typically via averaging or pooling, to form a fixed-size representation. This pooled vector is then passed through a classification head, consisting of a linear layer followed by a softmax activation, to predict the corresponding emotional label.

6. Classification Head and Loss Function

The reprogrammed audio embeddings are passed through GPT-2, which outputs a sequence of contextualized token representations. To perform classification, we apply mean pooling across the token dimension to obtain a fixed-size vector $h \in \mathbb{R}^{768}$, summarizing the emotional features of the entire audio input.

A fully connected classification head is applied to transform the pooled representation into a probability distribution over the 7 emotion categories:

$$\hat{y} = \text{softmax}(W_c \cdot h + b_c)$$

In this formulation, h denotes the pooled output vector from GPT-2. The matrix $W_c \in \mathbb{R}^{7\times768}$ represents the learnable weights of the classification layer, while $b_c \in \mathbb{R}^7$ is the corresponding bias vector. The output $\hat{y} \in \mathbb{R}^7$ represents the predicted probability distribution across the seven emotion classes.

To optimize the model, we employ the categorical cross-entropy loss function defined as:

$$\mathcal{L} = -\sum_{i=1}^{7} y_i \log(\hat{y}_i)$$

Here, y_i is the one-hot encoded ground truth label for class i, and $\hat{y}_i$ is the predicted probability for the same class.

7. Training Procedure

The model was trained using the AdamW optimizer with a learning rate of 1×10^{-4}, incorporating a linear warmup schedule. Training was conducted for 5, 15, 50 epochs, depending on early stopping criteria to prevent overfitting. A batch size of 16 was used throughout. To enhance generalization and stabilize training, regularization techniques such as dropout (with a dropout rate of 0.1) and gradient clipping were applied.

All layers, including GPT-2, are fine-tuned. Mixed precision training was optionally used to accelerate training on GPUs.

Chapter 5

Implementation Details

This section outlines the technical configuration, development environment, and key parameters used in implementing the proposed Voice Emotion Classification pipeline based on a reprogrammed GPT-2 transformer. The implementation was carried out using Python and popular deep learning libraries within a GPU-enabled environment for efficient training and experimentation.

Development Environment

The model was developed and trained using the following software stack:

- **Programming Language:** Python 3.9
- **Deep Learning Framework[22]:** PyTorch 2.0
- **Transformer Toolkit:** HuggingFace Transformers (v4.x)
- **Audio Processing:** Librosa, NumPy, SciPy
- **GPU Acceleration:** CUDA 11.7 with NVIDIA RTX 3060 (12GB)
- **Training Platform:** Google Colab P/ Local Workstation

Preprocessing Configuration

- **Sampling Rate:** 16,000 Hz (resampled from original 48 kHz)
- **Feature Type:** MFCC (Mel-Frequency Cepstral Coefficients)
- **Number of MFCCs:** 40 coefficients per frame
- **FFT Window Size:** 1024 samples
- **Hop Length:** 256 samples
- **Input Shape for Patch Embedding:** Sliced MFCC sequences into fixed-length 1D patches compatible with GPT-2 input

Model Configuration

- **Base Model:** GPT-2 (117M parameters)
- **Embedding Projection:** A linear layer maps MFCC-based patch embeddings to a 768-dimensional input space expected by GPT-2
- **Positional Encoding:** Learnable positional embeddings are added to retain the temporal ordering of the audio signal
- **Reprogramming:** An optional trainable prompt of length 5 tokens is prepended to guide the model adaptation
- **Classifier Head:** A fully connected output layer with softmax activation predicts the emotion class
- **Output Classes:** (7 classes) *Angry, Disgust, Fear, Happy, Neutral, Pleasant Surprise, Sad*

Evaluation Metrics

To assess performance, the model was evaluated on the test set using the following metrics:

- **Accuracy**
- **Precision, Recall, F1-score**

Training Procedure

The training process involved the following steps:

1. Audio signals were converted into MFCCs and embedded into patches.
2. Patches were projected to GPT-2's embedding size and reprogrammed.
3. The full sequence was passed through GPT-2, followed by the classification head.
4. Gradients were backpropagated through all layers, including GPT-2, enabling full fine-tuning.
5. Best model weights were saved based on validation loss.

Chapter 6

Results and Evaluation

This section presents the evaluation of the proposed voice emotion classification model. The model was tested in a waiting portion of the TESS dataset and performance was assessed using standard classification metrics.

Classification Report

Table 6.1: Classification Performance of OAF

epoch	accuracy	Precision	Recall	F1-score
5	0.75	0.80	0.75	0.73
15	0.957	0.958	0.957	0.955
50	0.996	0.996	0.996	0.996

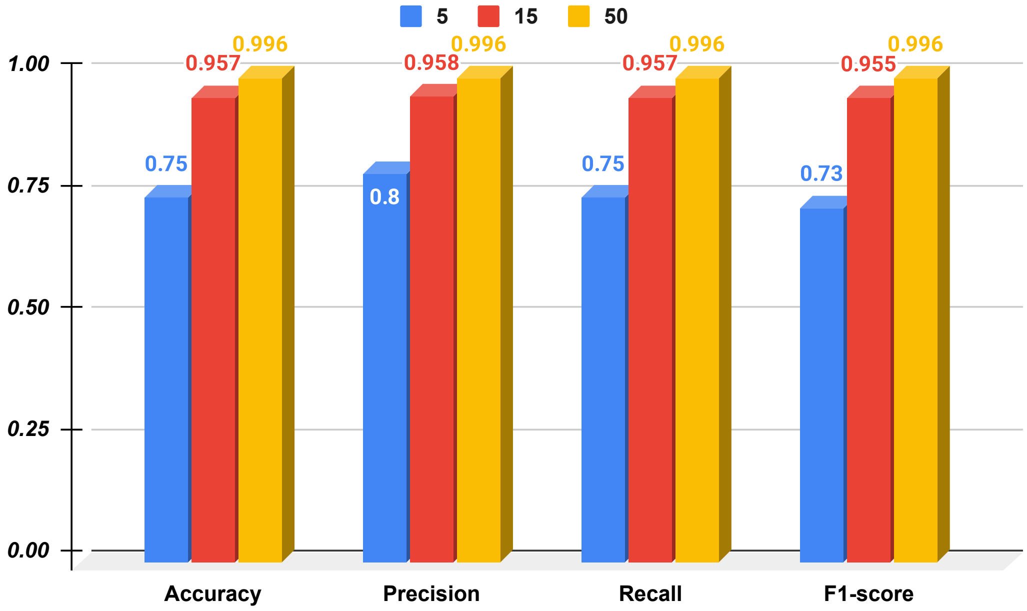

Figure 6.1: Classification performance of OAF model across 5, 15, and 50 epochs.

Table 6.2: Classification Performance of YAF

epoch	accuracy	Precision	Recall	F1-score
5	0.589	0.663	0.589	0.567
15	0.978	0.980	0.978	0.978
50	1.0	1.0	1.0	1.0

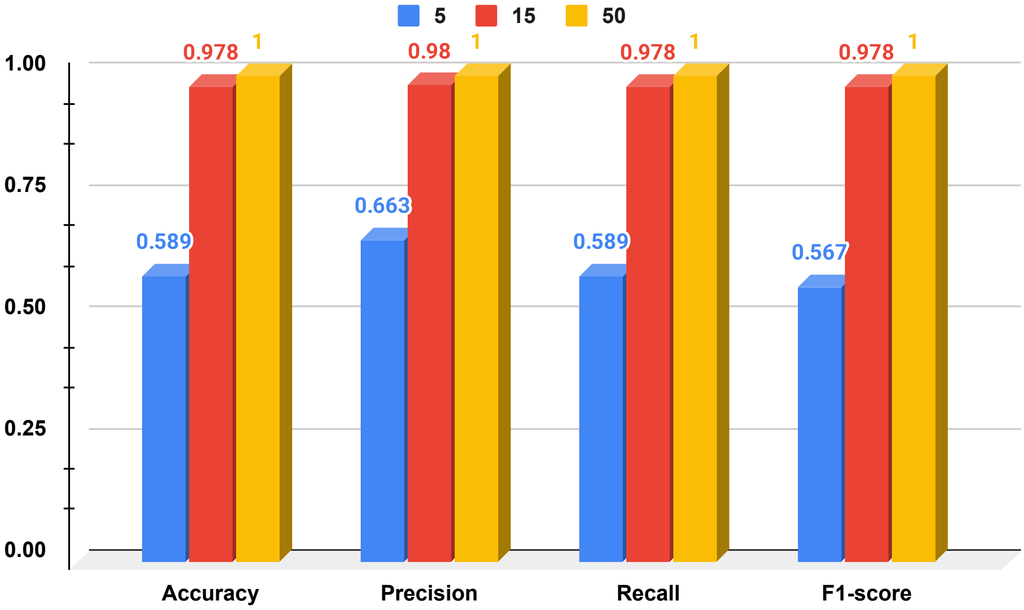

Figure 6.2: Classification performance of YAF model across 5, 15, and 50 epochs.

Table 6.3: Classification Performance of BOTH(OAF+YAF)

epoch	accuracy	Precision	Recall	F1-score
5	0.976	0.977	0.976	0.976
15	0.985	0.986	0.985	0.985
50	0.992	0.992	0.992	0.992

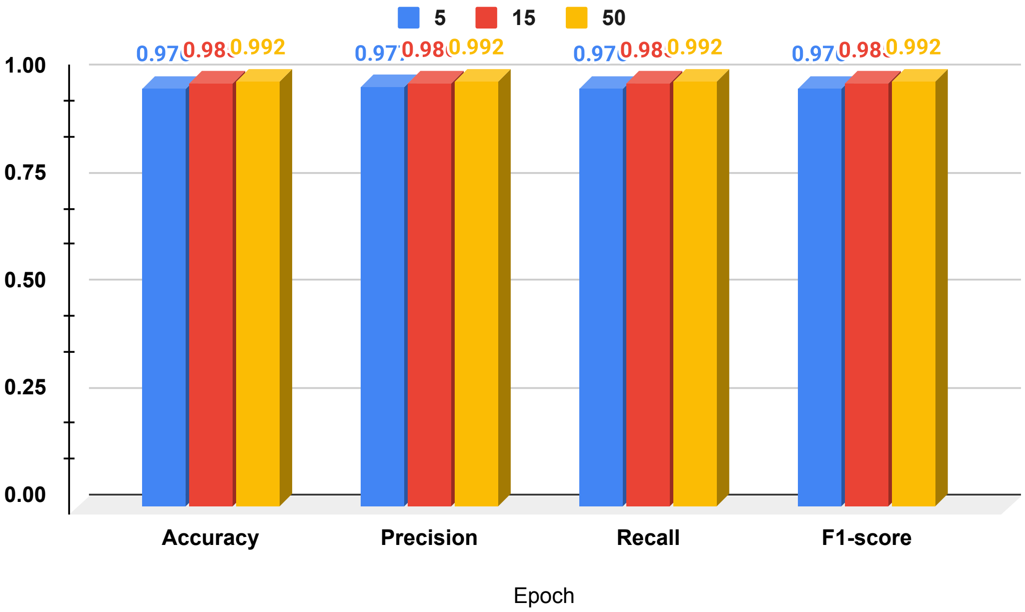

Figure 6.3: Classification performance of OAF+YAF model across 5, 15, and 50 epochs.

Chapter 7

Conclusion and Future Work

Conclusion

This project proposed a novel method for speech-based emotion recognition by reprogramming the GPT-2 transformer architecture for audio classification. Using MFCC representations of emotional speech, we introduced a patch embedding mechanism that effectively transformed audio features into GPT-2 compatible token sequences. The model was fine-tuned end-to-end on the TESS dataset, achieving strong generalization in emotion classification. Additionally, the framework is modular and adaptable, making it suitable for a wide range of sequence-based tasks beyond natural language processing.

The proposed model achieved strong classification performance in all seven emotional categories, as shown in the evaluation results. The ablation study further validated the significance of the proposed architectural components, particularly the reprogramming layer and GPT-2 fine-tuning.

Future Work

Although the results are promising, several directions exist for further exploration and improvement:

- **Multimodal Extension:** Integrate facial expressions or physiological signals (e.g., EEG, heart rate) to build a multimodal emotion recognition system.
- **Model Compression:** Explore lightweight alternatives to GPT-2 for deployment on edge devices or mobile platforms.
- **Cross-Language Generalisation :** Evaluate model robustness across different languages or accents to improve global applicability.
- **Real-Time Emotion Tracking:** Optimise the real-time audio emotion inference pipeline, functional in affective computing and human-computer interaction systems.

In summary, this work contributes a novel perspective to voice emotion classification by demonstrating the versatility of large language models when reprogrammed for new domains. It opens the door for future research into cross-domain transformer reuse and efficient speech understanding.

Bibliography

[1] O. Community, "openai-community/gpt2," *Hugging Face Model Hub*, vol. N/A, p. N/A, 2024, accessed: 2025-07-22. [Online]. Available: https://huggingface.co/openai-community/gpt2

[2] M. K. Pichora-Fuller and K. Dupuis, "Toronto emotional speech set (TESS)," 2020. [Online]. Available: https://doi.org/10.5683/SP2/E8H2MF

[3] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980. [Online]. Available: https://doi.org/10.1109/TASSP.1980.1163420

[4] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 15, no. 3, Mar. 2024. [Online]. Available: https://doi.org/10.1145/3641289

[5] Y. Wang, W. Zhong, L. Li, F. Mi, X. Zeng, W. Huang, L. Shang, X. Jiang, and Q. Liu, "Aligning large language models with humans: A survey," *arXiv preprint arXiv:2307.12966*, 2023. [Online]. Available: https://arxiv.org/abs/2307.12966

[6] Z. Liang, Y. Xu, Y. Hong, P. Shang, Q. Wang, Q. Fu, and K. Liu, "A survey of multimodal large language models," in *Proceedings of the 3rd International Conference on Computer, Artificial Intelligence and Control Engineering (CAICE '24)*. New York, NY, USA: Association for Computing Machinery, 2024, pp. 405–409. [Online]. Available: https://doi.org/10.1145/3672758.3672824

[7] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, and A. Ku, "Image transformer," *CoRR*, vol. abs/1802.05751, 2018. [Online]. Available: http://arxiv.org/abs/1802.05751

[8] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, L. Zhang, W. Han, M. Huang, Q. Jin, Y. Lan, Y. Liu, Z. Liu, Z. Lu, X. Qiu, R. Song, J. Tang, J.-R. Wen, J. Yuan, W. X. Zhao, and J. Zhu, "Pre-trained models: Past, present and future," *AI Open*, vol. 2, pp. 225–250, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666651021000231

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[10] J.-B. Alayrac *et al.*, “Flamingo: A visual language model for few-shot learning,” *arXiv preprint arXiv:2204.14198*, 2022. [Online]. Available: https://arxiv.org/abs/2204.14198

[11] H. Liu, C. Zhang, Z. Hu, Y. Wang, Z. Yang, J. Wang, W. Chen *et al.*, “Visual instruction tuning,” *arXiv preprint arXiv:2304.08485*, 2023. [Online]. Available: https://arxiv.org/abs/2304.08485

[12] N. Babu, J. Mathew, and A. P. Vinod, “Large language models for eeg: A comprehensive survey and taxonomy,” 2025. [Online]. Available: https://arxiv.org/abs/2506.06353

[13] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, “Support vector machines,” *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, 1998.

[14] L. Rabiner and B. Juang, “An introduction to hidden markov models,” *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[15] T. M. Cover and P. E. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967. [Online]. Available: https://doi.org/10.1109/TIT.1967.1053964

[16] K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” *CoRR*, vol. abs/1511.08458, 2015. [Online]. Available: http://arxiv.org/abs/1511.08458

[17] A. Sherstinsky, “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network,” *CoRR*, vol. abs/1808.03314, 2018. [Online]. Available: http://arxiv.org/abs/1808.03314

[18] R. C. Staudemeyer and E. R. Morris, “Understanding LSTM - a tutorial into long short-term memory recurrent neural networks,” *CoRR*, vol. abs/1909.09586, 2019. [Online]. Available: http://arxiv.org/abs/1909.09586

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[20] D. Khurana, A. Koli, K. Khatter, and S. Singh, “Natural language processing: State of the art, current trends and challenges,” *CoRR*, vol. abs/1708.05148, 2017. [Online]. Available: http://arxiv.org/abs/1708.05148

[21] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, “A comprehensive overview of large language models,” 2024. [Online]. Available: https://arxiv.org/abs/2307.06435

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” *CoRR*, vol. abs/1912.01703, 2019. [Online]. Available: http://arxiv.org/abs/1912.01703