Automatic Multi-Modal Emotion Recognition for AI Interaction

Final Report

Student: Li Shiyi 3035844405

Department of Computer Science University of Hong Kong

Supervisor: Wu Chuan

2024/04/21

Abstract

Emotion recognition is a critical part of human-computer interaction, with applications across various fields such as marketing, security, and mental healthcare. However, existing ER models often require high computational resources due to their complex structures and large parameter sets, limiting their practicality for real-time applications on smart devices. To address these issues, this project develops a lightweight, multi-modal model for efficient real-time emotion recognition. The model utilizes data inputs including whole images, cropped facial images, and audio data. End-to-end training is conducted using shallow Convolutional Neural Networks and Long Short-Term Memory networks for spatial and temporal feature extraction, with the simple concatenation-based fusion approach further reducing complexity. The model is trained end-to-end and evaluated on the MELD dataset, with data augmentation techniques applied to improve robustness. Preliminary results show promising performance, particularly for dominant emotion classes. Future work will involve expanding to more diverse datasets and exploring advanced architectures for improved generalization and fine-grained emotion recognition.

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

As a crucial part of human-computer interaction, emotion recognition (ER) is drawing increasing attention because of its vital importance in several application fields, such as marketing, security, and mental healthcare [1]. Especially, in the development of interactive chatbots, the integration of human emotion status into AI models can help generate more natural and intuitive responses, thereby enhancing user experience and improving response efficiency.

Although some models have shown promising accuracy in emotion recognition tasks, most of them feature complex structures and large parameter sets, which hinder their ability to produce results in real time. For example, the M2FNet (Multi-modal Fusion Network) model achieves high accuracy by employing a hierarchical framework that processes features at both the utterance and dialog levels. It first uses the RoBERTa-based text feature extractor and custom visual and audio feature extractors to obtain utterance-level information, then combines these features through a multi-head attention fusion mechanism [2]. All these feature extractors and the attention mechanism significantly contribute to the model's high parameter count, making it computationally intensive and challenging for daily usage.

Significant demand exists for developing smaller models that can run effectively on smart devices (e.g., mobile phones or embedded systems) while maintaining a reasonable level of accuracy for emotion recognition. Adjusting the balance between model complexity and efficiency is essential for enabling real-time emotion detection across a wide range of smart devices.

This paper makes two contributions. First, it designs a lightweight, multi-modal model capable of efficient automatic emotion recognition and easy integration into existing systems. The model will be trained and validated using up-to-date datasets to ensure robustness and accuracy. Second, this paper develops a web interface that provides real-time emotion recognition for demonstration purposes. The interface leverages audio and visual data collected from the smart

devices to generate emotion labels. A server will also be established to host this web page, ensuring efficient handling and processing of requests.

The remainder of this paper proceeds as follows. Sections 2.1 to 2.3 outline the three main components of our emotion recognition model. Section 2.1 introduces the preprocessing procedures implemented. Section 2.2 discusses the feature extraction methods used to efficiently capture temporal and spatial information. Section 2.3 describes the feature fusion approach to fuse embeddings derived from different modalities. The front-end and server-side design of the website is shown in Section 2.4. Finally, Section 3 discusses the progress being made so far, and Section 4 presents the project schedules and milestones.

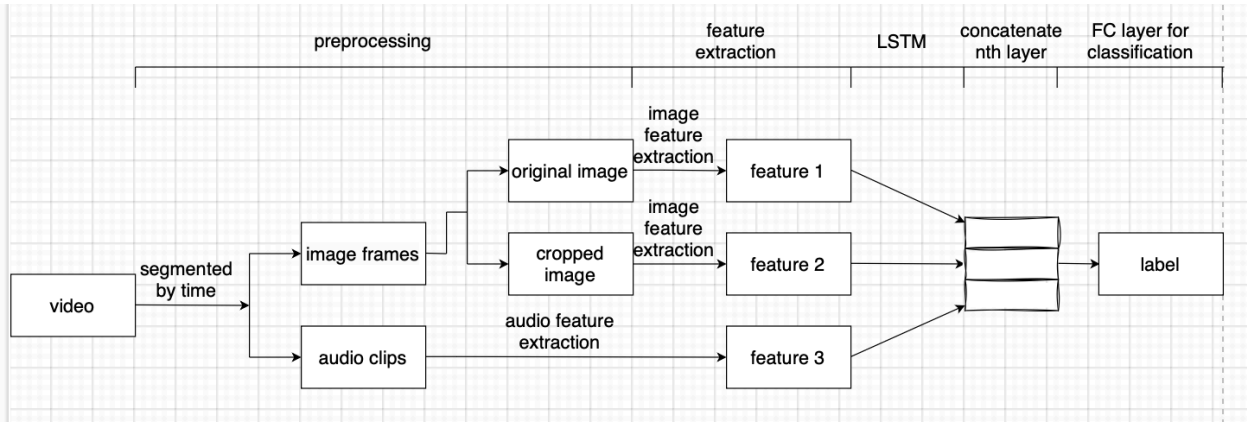

Figure 1. Model Structure

2. METHODOLOGY

Multi-modal emotion recognition models, compared to uni-modal models, utilize multiple data sources rather than a single source to detect human emotions. They have been proven to outperform uni-modal classifiers in detection accuracy[3]. By utilizing complementary data from multiple sources, multi-modal systems offer higher accuracy and robustness, especially in complex, real-world scenarios where noise or ambiguity can degrade single-modality

performance. Therefore, we will employ a multi-modal model for emotion recognition to leverage these advantages.

As shown in Figure 1, the model structure can be divided into three main sections: data preprocessing, feature extraction, and feature fusion across modalities. The process is facilitated using the End2You toolkit, which supports the integration of these components efficiently.

2.1 Data Preprocessing

Three types of inputs are utilized: the original video frame, the cropped facial region, and the associated audio segment. Each utterance is first segmented into synchronized audio clips and image frames. A face detection module is then applied to extract facial regions from each frame, while the full-frame context is retained to preserve background and gesture information. This design is inspired by prior work [2][4], which suggests that combining facial expressions with contextual visual cues can improve emotion recognition accuracy.

Given the variable-length nature of conversational data, we implemented a custom collate function to dynamically align temporal dimensions across samples within a batch. This avoids excessive zero-padding that may degrade model performance.

Data augmentation strategies were applied during training to improve model robustness. For audio, pitch shifting and noise injection were used to increase variability. For visual data, a series of augmentations were applied, including random horizontal flipping, random rotation, and color jittering (adjusting brightness, contrast, saturation, and hue). During evaluation, only resizing and normalization were applied to ensure consistency of inputs.

2.2 Feature Extraction

Recent studies [5] have demonstrated that shallow CNN + RNN architectures are highly effective for end-to-end multimodal emotion recognition, often outperforming attention-based methods in low-resource settings while requiring fewer parameters. Based on these findings and

the goal of real-time performance, lightweight models were selected for both visual and audio modalities.

For the visual stream, features were extracted using MobileFaceNet, applied to both the original video frames and the cropped facial regions. These spatial features were then passed through two-layer, unidirectional LSTM modules to model temporal dynamics across frames.

In parallel, audio segments were converted into Mel-spectrograms, which were treated as 2D grayscale images. These were processed using a shallow convolutional neural network (AudioCNN) followed by LSTM layers, enabling the model to learn both spatial (time-frequency) and sequential patterns within speech signals.

2.3 Feature Fusion

To maintain a lightweight model architecture suitable for real-time applications, we avoided complex fusion techniques such as cross-modal attention [6]. Instead, we adopted a simple yet effective strategy: feature embeddings from the visual and audio modalities are concatenated at the feature level, and the combined vector is passed through a series of fully connected layers to produce the final emotion label.

This straightforward fusion method ensures low latency and reduces computational overhead while still capturing complementary information across modalities.

3. EXPERIMENTS AND RESULTS

This section presents the experimental setup, design iterations, and evaluation results of our proposed multimodal emotion recognition model. We first describe the dataset and implementation details, followed by the architectural evolution informed by practical constraints and performance considerations. The final model is then evaluated on the MELD test set using a range of standard classification metrics. The results highlight both the strengths and limitations of the current approach, particularly its ability to handle class imbalance and represent subtle emotional expressions.

3.1 Dataset and Setup

We evaluated our model on the Multimodal EmotionLines Dataset (MELD) [7], a benchmark dataset for multimodal emotion recognition in conversations. MELD contains over 13,000 utterances annotated with seven emotion categories: *neutral, joy, sadness, anger, surprise, fear,* and *disgust*. Each utterance is accompanied by aligned audio and video, enabling synchronized multimodal analysis.

In this work, we focused on audio and visual modalities, processing each utterance into a sequence of video frames (10 FPS) and a corresponding audio clip. Facial regions were extracted from each frame, and Mel-spectrograms were computed from the raw audio signals.

The model was implemented using PyTorch and trained on NVIDIA RTX 3090 GPUs. To accelerate training and increase throughput, we adopted multi-GPU parallelism via nn.DataParallel. However, due to memory constraints arising from the use of both full-frame and cropped facial inputs, the effective batch size was reduced to 16 to ensure training stability and avoid out-of-memory (OOM) errors during multimodal processing.

The core training configuration is summarized in Table 1. All hyperparameters and system settings are managed through a centralized configuration file (config.py) to facilitate modular design and easy experimentation.

Parameter	**Value**
Batch size	16
Image resolution	112 × 112
Frame rate	10 FPS
Audio sampling rate	16 kHz
Optimizer	Adam
Initial learning rate	1e-4

Parameter	Value
Batch size	16
Image resolution	112 × 112
Frame rate	10 FPS
Loss function	CrossEntropyLoss (with class weights)
Gradient clipping	Max norm = 1.0
LSTM layers	2 (unidirectional)
Dropout rate	0.3
Evaluation metrics	Accuracy, F1-score (macro, weighted, micro), Confusion matrix

Table 1. Training Configuration

To address the class imbalance in MELD, class weights were computed from training label frequencies and incorporated into the loss function. During the early training phase, the MobileNetV2 backbone in RetinaFace was frozen to stabilize learning in the LSTM and classifier layers. After initial convergence, the backbone was unfrozen to enable full end-to-end fine-tuning.

To support flexibility and reproducibility, all hyperparameters, training settings, and data paths were managed via a centralized configuration file (config.py), facilitating efficient tuning, structured experimentation, and multi-stage training strategies (e.g., freezing and unfreezing backbone layers).

3.2 Design Evolution

The development of the final multimodal architecture involved several design iterations and practical trade-offs. This section outlines the progression of decisions that shaped the current system.

Initially, the project adopted the End2You toolkit [8] for rapid prototyping. Its modular design, with support for audio and visual streams and built-in training pipelines, allowed us to quickly validate the feasibility of our approach. However, substantial limitations soon emerged. First, End2You required all input data to be converted into the .hdf5 format, a process that was both time-consuming and storage-intensive—preprocessing the MELD dataset was projected to exceed 200 GB. More critically, its visual pipeline did not support feeding both full-frame and cropped face images, which was essential for capturing both contextual and fine-grained facial cues. Modifying the internal structure to support this multi-stream input was neither intuitive nor clean. As a result, we discontinued the use of End2You and migrated to a custom PyTorch-based implementation for greater flexibility and modular control.

For face detection, the initial solution was MTCNN [9], which has been widely adopted for accurate facial alignment. However, the implementation depended heavily on PIL (Python Imaging Library) image formats, which require image tensors to be moved from GPU to CPU for processing. This significantly limited the efficiency of our GPU-parallel training pipeline. Moreover, PIL-based I/O operations could not be easily serialized or batched, making MTCNN a major bottleneck when processing large batches of video frames during training.

We then explored using YOLO-Face [10], a real-time face detection model based on YOLOv5. Although promising in terms of speed, the available open-source implementations were outdated, poorly documented, and frequently failed on corner cases within MELD, particularly involving non-frontal or low-resolution faces. Moreover, as YOLO was originally designed for general object detection across multiple object categories, its architecture and anchor settings are not specifically optimized for fine-grained, single-class detection tasks like facial region extraction. Due to these limitations, YOLO-Face was ultimately abandoned in favor of a more specialized solution.

Ultimately, we adopted RetinaFace [11] with a MobileNetV2 backbone as our face detection module. Although the official implementation was originally available only in TensorFlow, we referenced a reliable PyTorch reimplementation from GitHub, and further handcrafted a simplified and modular version tailored to our pipeline. This customized version maintained key components such as multi-scale feature maps and single-stage anchor-based detection, while reducing unnecessary dependencies and adapting input/output formats for seamless integration.

In addition to architectural iterations, we also revised the temporal input formatting strategy. Initially, each utterance was processed using a fixed-length format (e.g., 2 seconds of audio and a fixed number of frames). This approach introduced inefficiencies: shorter utterances required heavy padding, while longer ones were truncated, leading to context loss. To improve flexibility while maintaining batching efficiency, we adopted a batch-wise dynamic padding strategy, where each mini-batch is padded according to the maximum sequence length within the batch, and overly long sequences are truncated to an average upper bound (e.g., 4 seconds). This balances memory usage and contextual coverage, reducing unnecessary padding while preserving important sequential information.

An overview of design evolution across core components is shown in Table 2.

Component	**Initial Approach**	**Issue Encountered**	**Solution**
Model framework	End2You toolkit	High storage demand, no dual-image input support	Custom PyTorch pipeline
Face detection	MTCNN	PIL-based, slow and non-serializable	Replaced with RetinaFace
Face detection	YOLO-Face	Unstable, outdated repo, poor detection on MELD faces	Replaced with RetinaFace
Input formatting	Fixed 2s / fixed frames	Excessive padding and context loss	Dynamic batch-wise alignment

Table 2. Design Iteration Summary for Major Components

3.3 Results

We report the evaluation results of the model on the MELD dataset. While the model performs reasonably on dominant classes, such as neutral, it fails to generalize to minority emotions.

3.3.1 Evaluation Metrics

To evaluate the model's performance, we used standard classification metrics including accuracy, precision, recall, and F1-score. Given the imbalanced nature of the MELD dataset, we report both macro-averaged and weighted-averagedF1-scores. Accuracy reflects the overall proportion of correct predictions, while macro-F1 gives equal weight to each class, highlighting performance on minority emotions. Weighted-F1 accounts for class imbalance by considering the relative frequency of each class.

In addition to these metrics, a confusion matrix and per-class classification report were computed to provide a more detailed understanding of prediction patterns and class-wise performance.

3.3.2 Experimental Results

On the MELD test set, the model achieved an overall accuracy of 48.16%, with a macro-F1 score of 0.096 and a weighted-F1 score of 0.315. The per-class results are summarized in Table 3.1. The model demonstrated acceptable performance on the dominant class neutral (F1 = 0.65), but failed to generalize to minority classes such as fear, disgust, and sadness, with near-zero F1-scores.

Notably, joy and anger exhibit unusually high precision but extremely low recall, suggesting that the model rarely predicts these classes, but when it does, the predictions are occasionally correct. Conversely, certain emotions such as fear and disgust are never predicted at all, resulting in zero recall and F1-score for these categories.

This pattern of performance is likely due to several factors: the severe class imbalance present in the MELD dataset, the limited expressive capacity of the lightweight CNN+LSTM architecture,

and the lack of high-quality training examples for underrepresented emotions. In addition, due to time and computational resource constraints, we were unable to perform large-scale hyperparameter tuning or experiment with deeper or pretrained modules, which may have further impacted model generalization.

Emotion	Precision	Recall	F1-score	Total
Neutral	0.482	0.998	0.650	1254
Joy	1.000	0.002	0.005	402
Sadness	0.000	0.000	0.000	208
Anger	0.750	0.009	0.017	345
Surprise	0.000	0.000	0.000	281
Fear	0.000	0.000	0.000	50
Disgust	0.000	0.000	0.000	68
Avg.	0.485	0.482	0.315	2608

Table 3. Evaluation Results

4. CONCLUSION AND FUTURE WORKS

This project explored a lightweight multimodal approach for real-time emotion recognition, combining facial and audio features using CNN+LSTM architectures. The model was trained and evaluated on the MELD dataset, with carefully designed pipelines for data preprocessing, feature extraction, and modality fusion. Throughout the project, several implementation challenges were encountered, such as data imbalance, limited model capacity, and computational constraints. Solutions including class-weighted loss, sample-level data augmentation, and staged training were applied to mitigate these issues.

The final model demonstrated acceptable performance on high-frequency emotion classes such as neutral, but failed to generalize to underrepresented categories like fear, disgust, and sadness.

Despite various optimization efforts, the performance gap between majority and minority classes remained significant. This outcome suggests that, while lightweight architectures are attractive for real-time applications, they may lack the representational capacity required for fine-grained emotion classification under real-world data conditions.

Future work will focus on improving generalization and representation learning in several ways. First, we plan to evaluate the current model on the CREMA-D dataset [12], which contains more demographically diverse speakers and controlled emotional expression, in order to assess cross-domain robustness. Second, although class-balanced loss and data augmentation were already applied, their limited effectiveness suggests the need for stronger architectural strategies, such as deeper CNNs or attention-based fusion, to better capture emotional cues across modalities. Third, instead of relying on manually tuned or unstable detection outputs, we propose integrating a pretrained face detection module at the input stage to provide consistent and accurate facial crops. This design ensures that the model receives clean and well-aligned visual inputs, which are critical for downstream emotion classification

5. REFERENCES

[1] K. Kamble and J. Sengupta, “A comprehensive survey on emotion recognition based on electroencephalograph (EEG) signals,” *Multimedia Tools and Applications*, Feb. 2023, doi: https://doi.org/10.1007/s11042-023-14489-9.

[2] V. Chudasama, P. Kar, Ashish Gudmalwar, N. J. Shah, Pankaj Wasnik, and Naoyuki Onoe, “M2FNet: Multi-modal Fusion Network for Emotion Recognition in Conversation,” *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2022, doi: https://doi.org/10.1109/cvprw56347.2022.00511.

[3] S. Poria, E. Cambria, R. Bajpai, and A. Hussain, “A review of affective computing: From unimodal analysis to multimodal fusion,” *Information Fusion*, vol. 37, no. 37, pp. 98–125, Sep. 2017, doi: https://doi.org/10.1016/j.inffus.2017.02.003.

[4] R. Kosti, J. M. Alvarez, A. Recasens, and A. Lapedriza, “Emotion Recognition in Context,” *IEEE Xplore*, Jul. 01, 2017. https://ieeexplore.ieee.org/document/8099695

[5] V. Karas, Mani Kumar Tellamekala, A. Mallol-Ragolta, M. Valstar, and B. Schuller, "Time-Continuous Audiovisual Fusion with Recurrence vs Attention for In-The-Wild Affect Recognition," Jun. 2022, doi: https://doi.org/10.1109/cvprw56347.2022.00266.

[6] R, Gnana Praveen, E. Granger, and P. Cardinal, "Cross Attentional Audio-Visual Fusion for Dimensional Emotion Recognition," *arXiv (Cornell University)*, Nov. 2021, doi: https://doi.org/10.48550/arxiv.2111.05222.

[7]S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea, "MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations," *arXiv:1810.02508 [cs]*, Jun. 2019, Available: https://arxiv.org/abs/1810.02508

[8]P. Tzirakis, S. Zafeiriou, and B. W. Schuller, "End2You -- The Imperial Toolkit for Multimodal Profiling by End-to-End Learning," arXiv (Cornell University), Jan. 2018, doi: https://doi.org/10.48550/arxiv.1802.01115.

[9]K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499–1503, Oct. 2016, doi: https://doi.org/10.1109/lsp.2016.2603342.

[10]W. Chen, H. Huang, S. Peng, C. Zhou, and C. Zhang, "YOLO-face: a real-time face detector," The Visual Computer, vol. 37, no. 4, pp. 805–813, Mar. 2020, doi: https://doi.org/10.1007/s00371-020-01831-7.

[11]J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-stage Dense Face Localisation in the Wild," arXiv:1905.00641 [cs], May 2019, Available: https://arxiv.org/abs/1905.00641

[12]H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "CREMA-D: Crowd-Sourced Emotional Multimodal Actors Dataset," *IEEE Transactions on Affective Computing*, vol. 5, no. 4, pp. 377–390, Oct. 2014, doi: https://doi.org/10.1109/TAFFC.2014.2336244.

[13] OpenAI, ChatGPT, Oct. 2023. [Online]. Available: https://chat.openai.com/chat