

ML model to recognize prosodic or emotional cues in audio data

Group Members: Dana Myktybek, Aigerim Dairanbek, Dariga Borasheva, Dilnaz Nauryzbayeva, Yeldana Barakatova.

Adviser(s): Prof. Dimitrios Zorbas

Date: April 20, 2026

1. Executive Summary

1.1 Problem Statement

The problem statement is to create a real-time speech emotion recognition model that recognizes prosodic characteristics such as rhythm, intonation, pausing, stress, and loudness from an audio stream on a device with limited computational power, such as a Raspberry Pi. Prosodic aspects account for a considerable share of the emotions conveyed through human speech; nevertheless, existing systems for extracting prosodic features are too large and require excessive computational power to operate within a limited space. In addition, one of the main problems related to implementations of SER, is the fact that this process is very speaker-dependent, meaning that for example, different language and gender features, all affect the performance of the system [3]. Therefore, the problem is to develop a model that will be able to classify emotions without exceeding the limits of a small-sized device in terms of computational power.

1.2 Objectives

The main objective of the project is to build a high-performing, efficient, real-time emotion detection system based on spoken languages using an algorithm that is computationally inexpensive and capable of running even on devices like the Raspberry Pi. The algorithm will efficiently detect six basic types of emotions expressed in the form of speech: happiness, sadness, anger, fear, surprise, and disgust.

To be considered successful, the proposed solution must find a proper trade-off between efficiency and accuracy to perform optimally on several benchmark datasets like RAVDESS, CREMA-D, and TESS, but not being too specific to one particular dataset and having sufficient memory requirements (2 GB of RAM or less).

The second goal of this project is to create and implement a suitable feature extraction pipeline. Here, the pipeline must extract both spectral and prosodic features. It means that it should not

only extract frequency features, but also look for variations in speaking styles, which are influenced by, for example, pitch and energy.

1.3 Methodology

The project utilized labeled speech emotion data files acquired from the RAVDESS, TESS and CREMA-D corpora. Custom data recordings were created with the voices of teammates to address issues such as background noise and lack of emotions in the audio to ensure that the dataset is various enough. All files were normalized at a sample rate of 16 kHz, also then turned into mono channels and then segmented accordingly. The segments were then preprocessed through different feature extraction methods, for example, MFCCs, Log-Mel Spectrograms, pitch and energy. There were different models created to find the best performing model that shows highest accuracy. They include the standard SVM using averaged MFCCs, an optimized SVM with statistical measures as well as the average features, CNN-BiLSTM with attention mechanism, and a multilayer perceptron (MLP). Various experiments were carried out in terms of architecture design and optimization, after finding the best-performing model, it was then optimized further to be run on Raspberry Pi using TensorFlow Lite or ONNX Runtime.

1.4 Key results

As the first experiment, we used a very simple linear SVM classifier with mean MFCC features as inputs. The result obtained, 43% accuracy, indicated that the task is solvable but that a too simplistic approach could miss important information contained in prosodic and temporal features.

A much higher accuracy (70-72%) was achieved using a richer set of features and a non-linear SVM classifier. The additional features included MFCC statistics, prosody, energy, etc. The result obtained confirmed that better feature extraction leads to higher precision.

Then, more sophisticated neural network models were considered. The CNN-BiLSTM with attention mechanism achieved an accuracy of 72.69%. It was shown that it can capture time-dependency in speech; nevertheless, there were no benefits compared to baseline model and difficulties of implementation because of the high computational complexity.

The most effective model is MLP, which is trained using the combined dataset that includes RAVDESS, CREMA-D, TESS, and recorded audio clips without any background noise. The highest level of accuracy was 78.8%, while the classification of predictions shows more symmetry for all six emotions. Errors in classifications are avoided, although some mistakes still occur, for example, some emotions such as fear and sad, which have similar energy and tone are still confused.

In terms of the system, the most optimal solution was to use the highest-performing MLP. The model could process the audio signal fast enough (less than 0.3-0.4 seconds per three seconds of the audio clip) and met the demands on memory (the size of the model was 50 MB) and the computing power of the Raspberry Pi.

1.5 Computing Alignment

This particular project follows a certain design and implementation process which includes defining the problem and implementing it. In the context of emotion recognition using constrained resources, computing is used to solve the defined problem, starting with the definition of functional and non-functional requirements. The following steps include data collection, feature generation, model training, and implementation on a Raspberry Pi. Evaluation is then conducted for both functional (model accuracy) and non-functional (computing speed) requirements.

2. Introduction

2.1 Motivation & Significance

The expression of emotions plays an integral role in human-computer interaction, providing more intuitive insights into how machines work. The prosody of speech comprises aspects such as rhythm, intonation, timing, and volume, serving as the music of speech and is important in identifying different emotions. Some of the uses include assisting applications, conversation, and monitoring mental well-being since people affected by certain mental illnesses, such as autism, can utilize the information in this study [1]. Despite its importance, state-of-the-art emotion classifiers are large and computationally expensive, making them incompatible with embedded devices with limited capacity.

2.2 Problem Description

The problem statement is about building an algorithm to recognize prosodic and emotional cues in the speech signal within a resource-constrained system such as a Raspberry Pi. There are several difficulties that need to be addressed first: unification of emotional classes from different datasets; recording new data since the available sets may contain noises and are not expressive enough; training a classifier that is able to identify six emotions, including happiness, sadness, anger, fear, surprise, and disgust, in real-time; and ensuring that some non-functional criteria are satisfied, namely, the minimum 80-85% accuracy level, model size no more than 50 MB, maximum RAM consumption of 2 GB, resilience against noises, fast booting up, and real-time performance on an embedded platform.

2.3 Proposed solution

Our proposed solution is an emotion recognition system for real-time speech, running on a Raspberry Pi. The model works when the person speaks in real time through the microphone, and then the model extracts necessary features from the voice, for example, pitch and energy of the audio. These features are the inputs to the MLP model, which was identified as the best performing model of its high accuracy of 78.8%. Previous works also identified the MLP model as the architecture that can select enough data, especially with the MFCCs features [13]. MLP model was trained on open-source datasets and our own dataset. In addition, model pruning and quantization were carried out in order to fulfill the platform requirements related to small memory consumption and low latency.

2.4 Report & Organization

The rest of this report will cover a detailed overview of the project and its methodology. The Identification of Constraints section highlights the most important constraints relevant to the project, such as hardware-related constraints and ethical constraints. This section further elaborates on how these constraints impacted the selected solution for the project. Background and Related Work will discuss the background of the project, the field of speech emotion recognition in general, the different approaches to solving this problem, and why one of the particular approaches was selected above others. Project Approach will provide an explanation of the overall architecture of the system, for example, the chosen models and explanations of why they were chosen, and the whole data flow within the project. Deviations from the initial project vision, faced challenges, and decisions made to overcome them will be highlighted in the Project Execution section, that will cover two semesters of the course. Evaluation will show how the developed system was evaluated and the performance of our project, showing whether the project goals were reached and showing its results. Conclusion and Future Work will summarize the main results obtained in the project and the possible ways of improving it in the future.

3. Constraints

3.1 Assessment of Relevant Constraints

Economic, Environmental, and Social Constraints

Real-time deployment of Raspberry Pi was adopted as an economical process; however, the limited CPU, limited RAM (2GB), and lack of a dedicated GPU led to the restriction for model complexity. This involved needing to invest in compression methods such as quantization, pruning, and ONNX/TFLite conversion to not depend on cloud infrastructure [6]. Free datasets and open-source tools ensured costs remained within the academic constraints. Pitch-shifting, time-stretching, and SpecAugment were used to ensure robustness ($\text{SNR} > 10\text{dB}$) in the training

case to stabilize the performance of the model with real acoustic environment noise injections [4]. In the initial custom dataset recording there was uncontrolled background noise (from which 5–20% accuracy can be extracted), that had to be completely reworked under controlled conditions. Balanced classification and speaker-independent performance was focused on the eight emotion classes with the aim of making them accessible to assistive technology users, such as users with Autism Spectrum Disorder [1].

Political, Ethical, and Health & Safety Constraints

Institutional policies and licensing terms constrained the use of datasets - only those datasets having clear and permissive licenses were considered in our analyses, with no sources having ambiguous terms [10][11]. The risks of voice data re-identification and misuse are linked, and our team decided to process voice data on-device, store it in an encrypted local drive, and extract features only (MFCCs and spectrograms). Only clearly licensed datasets of ethical origin were used at the cost of reduced cultural diversity. To the potential applications of mental health monitoring, softmax-based confidence scoring was used to indicate a doubted prediction and prevent overtrust in model results.

Manufacturability and Sustainability Constraints

In other words, being deployed on Raspberry Pi was technically bound by a set of limits: the size of the model be less than 50MB, the CPU less than 80%, and inference latency lower than 100ms in one segment. Out of all these variables, MLP architecture was selected, which is the most optimal trade-off between the precision (78.8%) and compression capacity [24]. The system was in the meantime intended to be modular and extensible with the aim of being expanded upon in the future as required without major rewrites. This trend of low power IoT hardware over cloud infrastructure does not only make it cost less in power over the long-term, but also makes it cheaper to operate within the system, enabling it to exist in the real world in general without exhausting its resources in the long term.

3.2 Impact Evaluation

The strongest factor in general was the hardware constraint of Raspberry Pi - it was the one that predetermined the selection of handcrafted features over end-to-end models, the three-level model complexity approach, and even the whole spring optimization pipeline. Ethical and legal constraints determined dataset selection in a way that directly impacted accuracy. The choice to train on only three datasets RAVDESS, CREMA-D, and TESS kept the project legally sound (though this brought overfitting issues because of the very small number of training sets and cultural homogeneity) resulting in a high maintenance process during training. Environmental constraints were such that the initial custom dataset approach was unsuccessful, and a new recording protocol was developed – a typical instance of a real-world constraint necessitating a design update halfway through. Finally, the social constraint of speaker independence influenced

our evaluation approach: all splits were speaker-independent (80/10/10), and performance was monitored on a per-class basis to ensure that the program did not mask failures on underrepresented speakers or on emotion categories [1].

4. Literature Review

4.1 Overview of Speech Emotion Recognition (SER)

Speech Emotion Recognition (SER) is a branch of affective computing which is focused on identifying the emotions of a speaker based on the acoustic characteristics of the speech which include tone, pitch, intensity, etc. Applications of SER have a variety of direct uses, for instance, in human-computer interaction, mental health monitoring, IoT-based assistive technology and to be specific, in assistive technology to aid individuals with Autism Spectrum Disorder (ASD) [1]. In contrast to textual sentiment analysis, SER is performed on unprocessed audio signals, in which the information about the emotion is encoded in minor deviations features such as pitch, energy, speaking rate, spectral texture and so on, depending on the model, also known as prosodic and paralinguistic cues [2]. Paralinguistic cues, in our case, attribute to non-verbal cues such as laughter or sighing.

The main challenge with SER is that there is a strong inter-speaker difference between the expression of emotion in genders, languages, conditions of recording (abundance of background noise, the quality of the microphone, etc) and cultural backgrounds [3]. Another structural challenge is the situation of label vagueness: it can be challenging for people to agree with how emotion is acted on, because the expression of emotions can vary depending on the annotator [4]. That is why in a couple of databases discussed further in this review, there have been 2 or more levels of intensity of one emotion. Such difficulties are exacerbated in deployment situations, with the aim of a resource-constrained embedded hardware, where model complexity should be carefully regulated and cropped without undermining emotion recognition accuracy.

Initial SER systems were based on the hand-engineered acoustic characteristics coupled with traditional classifiers including Support Vector Machines (SVMs), Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs), some of which were explored during the first half of the year [3]. The resulting deep learning (DL) architectures that were adopted led to large improvements in accuracy on standard benchmarks at the cost of new computational and memory requirements that could not fit on embedded systems without much compression [5][6].

4.2 Acoustic Feature Extraction: MFCCs, Log-Mel Spectrograms and Prosodic Features

The input representation of a SER system greatly affects its overall performance. The literature we analyzed during these two semesters is dominated by two different types of features:

Mel-Frequency Cepstral Coefficients (MFCCs) and log-Mel spectrograms, which are based on the perceptual characteristics of human auditory processing.

The MFCCs were proposed by Davis and Mermelstein [7] and convert raw audio into a compressed cepstral representation of frequencies by averaging the output of a log-scaled Mel filterbank using the Discrete Cosine Transform. This is done because the perceptions of changes in low and high pitches by humans are different, while this method allows for identification of which pitches are present. The Mel scale is an approximation to the non-linear frequency resolution of the human cochlea which is primary organ for human hearing, condensing the high-frequency energy and enhancing the detail of low-frequency information in a way that is directly related to perceived pitch and timbre [7]. Historically, MFCCs were the most common characteristic of speech recognition and SER due to their efficiency in computation and low dimensionality, as well as, high discrimination ability in speaker-independent tasks.

Log-Mel spectrograms do not discard the dimensionality reduction that the cepstral step enforces on the time-frequency energy distribution as Mel filterbank processing does which is why they are considered to be richer. This gives them a higher spectral energy content and more amenable to input to architectures that capitalize on spatial or temporal variations in the frequency-time plane [8]. Trigeorgis et al. [8] showed that log-Mel representations with a deep convolutional-recurrent network can either equal or outperform hand-crafted features with a large enough model capacity.

4.3 Datasets

To achieve reproducibility and comparability of SER research, the corpora must be standardized. There are four datasets which are relevant to this work: CREMA-D, TESS, RAVDESS and SAVEE.

CREMA-D, which was introduced by Cao et al. [10], consists of more than 7,400 clips of 91 actors of a wide age, ethnicity and gender selection. The gender ratio is 48 males to 43 females. The methodological contribution of CREMA-D is its demographic diversity and almost equal gender distribution: sampling a heterogeneous speaker population, it provides the possibility to assess the deterioration of performance in recognition of underrepresented speaker groups [10].

According to Livingstone and Russo [11], the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) offers recordings of 24 professional actors (12 men and 12 women) who were engaged to recreate eight categories of emotion under controlled studio conditions. This dataset introduced 2 levels of intensity for each phrase. Its uniform quality of recordings and even distribution of classes make it a favorite corpus in ablation studies and in experiments of feature comparison where the confounding variation due to recording environment should be minimized.

Surrey Audio-Visual Expressed Emotion (SAVEE) database, which was recorded in the University of Surrey by Haq and Jackson [12], is an expression of seven categories of emotions using 480 British English utterances by four male actors. The TIMIT corpus was used to draw sentences, the main advantage of which was that those sentences were phonetically balanced to each emotion. Recognition rates of speaker-independent classification with standard features and classifiers were 61% (audio only), 65% (visual only), 84% (audio-visual combined) and audio-only SER with SAVEE set up audio-only unimodal classification systems as a valuable benchmark. In this project, SAVEE is used as a supplementary test, while the first three were used for training.

4.4 Classical Machine Learning (ML) Approaches

Before deep learning, classical pipelines that combine hand-crafted features with probabilistic or discriminative classifiers were predominant in SER. The Hidden Markov Models [2] was quite successful in recognizing emotional speech, which makes use of its inherent ability to model speech dynamics over time. GMMs were used to estimate the distribution of acoustic features per-class and classify the emotions through likelihood comparison [3].

Then the success of SVMs as the general-purpose classical classifier of SER to the end of the 2000s and the early 2010s was due to their stability to high-dimensional feature representations and effective generalization with minimal training data. In this time, Multilayer Perceptrons (MLPs) were also considered as feedforward neural alternatives. A systematic comparison of MLP, GMM and SVM classifiers was carried out by Ververidis and Kotropoulos [13] in emotional speech recognition, revealing that MLPs were able to show competitiveness in selecting enough data, especially using MFCC input. Importantly, in their analysis, they recognized the MLP as a viable option in limited deployment settings because it has a simple inference network and does not incur any overhead on kernel or density estimation at test time.

4.5 Deep Learning (DL) Architectures for SER

Deep learning applied to SER resulted in a significant improvement in accuracy as architectures learned task-informed representations from data as opposed to using hand-engineered features.

Convolutional Neural Networks (CNNs) were one of the earliest deep architectures to be used with SER, inspired by their naturally available use with spectrogram inputs as two-dimensional time-frequency images. Zhao et al. [5] established that implementation of CNNs on log-Mel spectrograms are capable of capturing local spectro-temporal patterns that are useful in the expression of emotion and that the use of attention mechanisms enhances the performance of CNNs employed on log-Mel spectrograms by selectively weighting emotionally salient temporal regions. According to a survey of CNN-based SER systems, the average accuracy is around 70-84% on standard datasets [14]. A 1D CNN architecture improved with channel and spatial attention, applied to MFCC features reached 97.49% on SAVEE, 99.23% on RAVDESS and

89.31% on CREMA-D [15], showing the maximum performance of CNN-based approaches in positive training conditions.

Recurrent architectures (especially Long Short-Term Memory (LSTM) networks) solve the sequential character of speech by storing a hidden state that combines information through the duration of an utterance [16]. Mirsamadi et al. [16] suggested an attention-based recurrent architecture conditioned on a weighted sum of past hidden states which showed better results than frame-level pooling strategies and proved that emotional content was not evenly spread over utterance length. On the EMO-DB dataset LSTM got 85.5% accuracy as compared to 78.75% on a four-class emotion task with a standard CNN, which demonstrates the benefit that time modeling can be useful. Hybrid CNN-LSTM models were then used to combine local feature extraction with sequential integration, where a four-layer CNN was used with three layers of LSTM with the result of 61.07 on CREMA-D [18].

The latest generation of SER architectures is transformer-based and self-supervised models. HuBERT models are self-supervised models that are trained on large, unlabeled collections of speech to extract rich contextual response as representations that are effective on emotion classification. The distilled version DistilHuBERT (18) scored 70.64% on CREMA-D with a F1 score of 70.36% using only 0.02MB of stored weights. In spite of good accuracy, transformer architectures are computationally expensive at inference time (with serial attention operations on long inputs) and are therefore not viable to be implemented on a CPU-only embedded system without heavy compression which affects the accuracy [19].

4.6 Comparative Analysis of Existing Architectures

Trade-offs (accuracy of recognition, complexity of models, inference latency, memory consumption and trainability in the face of constrained data) must be all considered when deciding upon a classification architecture of an embedded SER system. The MLP was chosen after a number of architectures had been implemented during the first semester of this project. Table 2.1 is a summary of the major nature of each of the approaches in the dimensions which are most relevant to our project since we are using a resource-constrained environment.

Property	SVM	MLP	CNN	CNN-LSTM	Transformer
Input type	Feature vector	Feature vector	Feature vector	Spectrogram/M FCC	Raw waveform (.wav)
Typical accuracy	77-83%	81-85%	78-97%	61-90%	70-87%

Inference speed (CPU)	Fast	Fast	Moderate	Slow	Very slow
Model size	SV-dependent	Small to Medium	Medium	Medium to Large	Very large (90M+ parameters)
RAM at inference	Low	Low	Moderate	High	Very high
Raspberry Pi feasibility	Yes	Yes	Partial	Unlikely	No
Data efficiency	High	High	Moderate	Low	Very low
Training complexity	Low	Low	Moderate	High	Very high
Selected for project	No (Tried)	YES	No (Tried)	No (Tried)	No (Not tried)

Table 2.1: Comparison of existing architectures used for SER

4.6.1 Convolutional Neural Networks (CNNs)

CNNs had a good SER performance (up to 72%) in the first-semester tests but had deployment costs that were slightly higher than could be afforded. Convolutional layers are known to have large intermediate feature maps which demand large amounts of memory and the number of parameters grows as the number of filters and as the depth of the convolutional layer, but in ways they are challenging to constrain without compromising representational power at the end. Large and self-supervised CNN-based models often have over 90 million parameters, which are infeasible to infer on embedded hardware with only a CPU without significant pruning and distillation [19]. The CNNs tested during the first semester demonstrated competitive accuracy on clean data owned by benchmarks but were over the target memory and inference time budgets to run on Raspberry Pi.

4.6.2 Recurrent Architectures

Models based on LSTMs have an inference overhead because of the sequentiality of recurrent computation which is not parallelizable, and increases latency linearly with utterance length. The recurring concealed condition should persist through all the timesteps which cannot be accommodated by the 2 GB RAM limit of Raspberry Pi. Moreover, the low accuracy by background noise was up to 18 percentage points lower in LSTM-based systems, which were tested under controlled conditions [20], which is a significant issue in the context of real-world implementation of the presented project. CNN-LSTM hybrids are more expressive but add to the memory demands of both architectures and were equally impractical on the target platform.

4.6.3 Transformer-Based Models

The cost of inference of transformer architectures made them ineligible in the candidate set. Although models like wav2vec 2.0 or its distilled versions can reach the competitive accuracy state, they typically need to be run on a GPU to perform real-time inference and cannot be run on a Raspberry Pi at their native accuracy without extensive accuracy loss when compressed [18][19].

4.6.4 Support Vector Machines (SVMs)

The inference of SVMs is computationally efficient and no calculation of the gradient is required, thus making them feasible to implement on an embedded platform. Their formulation, however, in kernel form needs to store all support vectors and the memory cost increases as the size of the training set increases. SVMs do not (also) support joint optimization of feature extraction and classification and thus are not as adaptable to augmented or combined features. A comparative analysis revealed that Gradient Boosting scored 84.96 percent on a mixed RAVDESS-TESS dataset, whereas MLP classifier scored better than any of the other classifiers tested, SVM and Random Forest, on all 3 separate datasets [21].

4.6.5 Multilayer Perceptron (MLP)

The MLP was selected as the main structure of this project, due to the overlap between its deploymentability and its empirical accuracy on the standard SER benchmarks and scalability to data constraints as it was mentioned in section 4.4. Due to the model size and RAM usage, and data inference of 1-2 seconds, we decided to go with MLP that can also work with the resource-constrained environment. Deploymentwise, the MLP presents the most simplistic graph of inferences of a neural architecture: a fixed sequence of matrix multiplications and non-linear activations with no recurrent state, no convolutional feature maps, no attention calculation. This results in limited, deterministic inference latency as well as low memory usage, both of which are crucial with the 2 GB RAM and real-time inference demands of the Raspberry Pi.

MPLs are a proven empirical selection of SER. A MLP trained with MFCC features resulted in 81% accuracy on RAVDESS [22] and another study observed MLP to be the highest performing single classifier with 85% accuracy on a four-emotion RAVDESS task when using MFCC and Mel features [23]. An ensemble model consisting of MLP and CNN components assessed on RAVDESS, EmoDB, SAVEE and TESS, affirmed that the MLP component significantly contributes to overall accuracy independent of the model [24], affirming the fact that MLP is a principled and not a fallback model choice.

4.7 Knowledge Gap

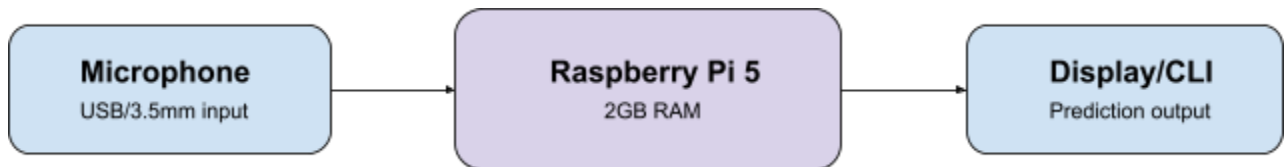
The above literature review indicates that there are a number of gaps, which drive the targeted contributions of the given project. Although with careful architecture design, deep learning systems have demonstrated good SER performance on benchmark data sets, their computational complexity makes deploying them on embedded hardware systems with resource constraints unfeasible without major compression. Lightweight substitutes, such as MLPs trained on hand-engineered acoustic features, have been relatively little systematically analysed in the past few years, although they have been demonstrated to have deployment benefits and competitive performance on standard corpora [13][21][23]. The performance of compressed MLP-based SER systems in comparison with less efficient architecture under embedded hardware constraints (in particular, on Raspberry Pi) has not been fully described in the literature.

Moreover, most published SER evaluations report accuracy with only clean benchmark data sets without reporting its performance with moderate background noise or across a combination of multiple evaluation corpora. The two gaps that are also addressed in this project include training and evaluation of an MLP based SER model that is optimized to run on Raspberry Pi and benchmarked on CREMA-D, RAVDESS and SAVEE on both controlled and noisy environments.

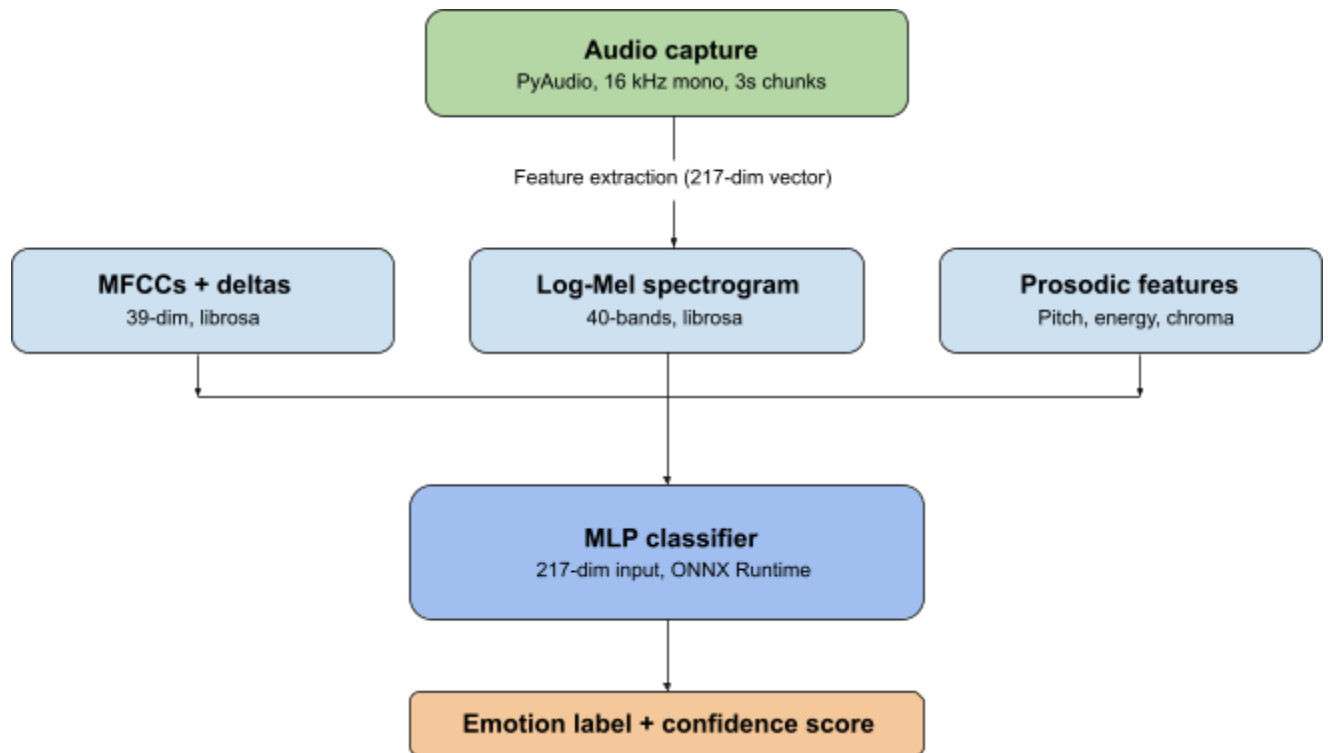
5. Project Approach

5.1. System Architecture

1. Hardware layer

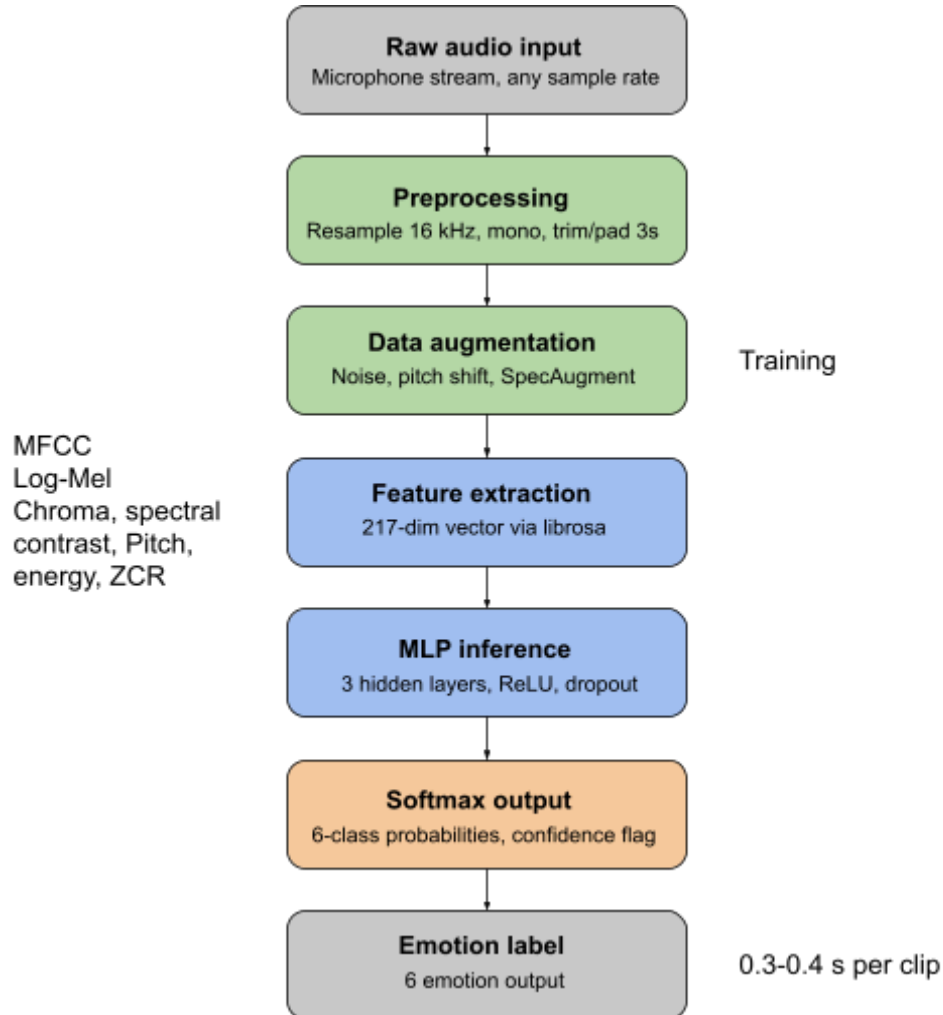


2. Software stack (Raspberry Pi 5)

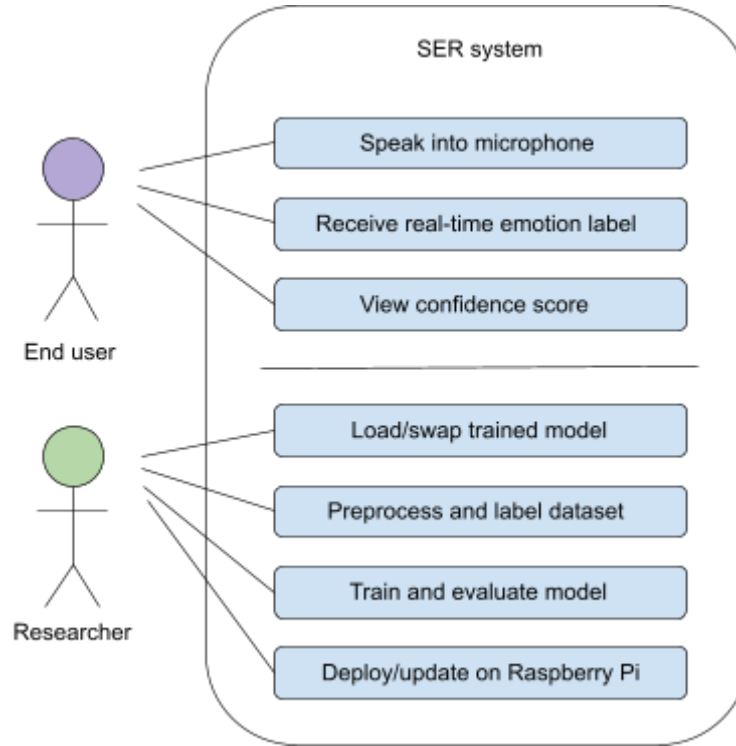


5.2 Technical Details

The given figure represents the end-to-end data pipeline, starting from raw audio through preprocessing, feature extraction, MLP inference, and final emotion label output. During training data augmentation is applied.



5.3 Use Case Diagrams



5.4 Third-Party Integration

Component	Library/Tool
Feature extraction	librosa
Audio capture	PyAudio
Model training	PyTorch
Model export & inference	ONNX/ONNX Runtime
Data processing	Numpy, pandas
Preprocessing	scikit-learn
Augmentation	audiomentations

5.5 Team Functionality

There were no rigid allocated roles, it was more flexible. Initially, each member independently analyzed the datasets which were used in different research papers (RAVDESS, IEMOCAP, CREMA-D) and created a unified dataset containing 6 emotions. Throughout the project's progress, members took responsibility for the tasks based on project needs, such as optimization, model training, preprocessing were collaborative efforts.

6. Project Execution

6.1 Timeline & Development

Fall semester:

In the fall semester, we have employed the iterative approach to developing a pipeline for emotion recognition through prosody. The main goal was to move beyond simple classifiers and develop deep learning models capable of recognizing prosodic patterns. Furthermore, we laid the foundation for future in-person tests, with a hardware installation planned for next semester.

Firstly, each teammate focused on analyzing different datasets: RAVDESS, IEMOCAP, and CREMA-D. For each dataset, we gathered the relevant information on the number of participants, the total number of utterances and samples, and the list of annotated emotions and the agreed-upon eight main emotions, combining similar categories (for example, neutral with calm).

An attempt was made to construct a very basic classifier based exclusively on mean MFCC values. To be precise, each audio sample was converted to a thirteen-dimensional vector consisting of mean values of thirteen coefficients. Using a linear support vector machine (SVM) yielded a classifier with only 43% accuracy. While it indicated the presence of learnable patterns in the provided data set, the poor accuracy was obvious in a number of factors. First of all, by reducing the signal to its mean value, we lost information about prosody, including features such as pitch variance, rhythm, and many others. As a result, our classifier showed extremely high confusion between calm, neutral, and sad emotions due to their similarity.

Next, it became necessary to enrich the feature representation used by the algorithm. For this purpose, an extended vector with 37 components was constructed. It included the mean and variance of MFCCs, duration, statistical features of RMS energy and zero-crossing rate, and several pitch-related parameters. The experiments conducted with RandomForest and a grid-searched SVM achieved accuracies of about 70-72%

Finally, despite a decent result, these algorithms could not account for temporal dependencies in audio signals. This fact motivated the construction of a deep learning framework. According to the previous related works, deep learning models better identify necessary voice features, for

example, spectral and temporal patterns [5]. Therefore, instead of summarizing the signal, the model processed feature sequences, including MFCCs, delta and delta-delta coefficients, chroma features, and spectral contrast, thereby preserving both spectral and temporal dynamics. Our model included a convolutional neural network (CNN) followed by a bidirectional LSTM to extract features and account for dependencies. The attention layer identified the most emotionally significant segments, while fully connected layers with dropout produced the final predictions. Training such a model was accompanied by stabilization methods such as learning-rate warm-up, gradient clipping, mixed-precision training, and SpecAugment data augmentation. All mentioned changes and implementations led to the increase of the accuracy to 72.69%. Apart from creating different neural network models, we made our dataset, which was in line with the RAVDESS framework, comprising around 100 sound clips for different emotions. The previously trained models performed rather poorly on this data set, achieving only about 5% accuracy with medium-complexity models and up to 20% with an LSTM, while predicting a single emotion most of the time.

Spring semester:

First of all, we decided to simplify the task by reducing the number of emotions to predict and standardizing their names. In particular, in spring, we worked with only six labels: happiness, sadness, anger, fear, surprise, and disgust. We believe this step greatly helped us reach our goals by reducing potential misinterpretations during sample labeling.

The second thing that we had to do was build a completely new dataset, as the recordings we used in the Fall semester were of low quality. In addition to using our previously created dataset, which includes recordings in a quiet room with a consistent microphone position and more expressive performance, we merged several other pre-existing databases, namely RAVDESS, CREMA-D, and TESS. All files were converted to 16 kHz mono, resampled to the same length, and normalized before being split into the three partitions, which should not contain recordings by the same speakers to provide an unbiased evaluation of our models. To improve our model, we again tried to use a convolutional neural network (CNN), which is aimed at finding subtle differences in sounds based on mel-spectrograms. Nonetheless, CNN performed poorly in testing the small-sized SAVEE database because it predicted all audio files as sad emotions.

We switched to training a multilayer perceptron (MLP) based on a 217-dimensional feature vector (log-Mel spectrogram, MFCC, chroma, spectral contrast, and pitch). The presented model outperformed all other models used in this study. For example, on the modified dataset, the accuracy exceeded 78.8%, and inference time was 0.3-0.4 seconds per sound sample. Besides, we calculated the sizes of the models and made sure that they fit the imposed storage restrictions.

Another issue related to robustness needs mentioning. We discovered that MLP demonstrates better performance than SVM and CNN-biLSTM while being tested on another dataset and our voice samples; however, the accuracy of class prediction stays lower than 80-85%. In particular, some classes such as fear or happiness are hard to classify correctly. Cleaning the input data and merging labels positively impacted model performance. Our decision also aligns with the results

of the prior research, where MLP was considered the best performing architecture in these types of problems, such as SER [23]. Moreover, adding the CREMA-D and TESS databases allowed us to diversify the training data set. MLP proved to be the best classifier with a compact size fitting our device, i.e., Raspberry Pi. At the end of the term, we managed to create the whole pipeline for audio data from recording to classification in real-time mode.

6.2 Design Decisions

Fall semester:

1. In order to accommodate for heterogeneity among the different databases, we standardized the emotion labels by merging “neutral” and “calm” categories and choosing eight emotions to be recognized.
2. We explored three different model complexities:
 - Basic (Linear SVM)
 - Improved classical model (SVM with RBF kernel and statistics of MFCC)
 - Prosody-aware models with advanced architecture (CNN+BiLSTM+Self-Attention)
3. Stratified split 70/15/15 was used as a procedure of training-validation-testing with clear separation of speakers to avoid potential data leakage. As a result, the first database (RAVDESS) was chosen as the main training dataset. In addition, other datasets (CREMA-D and IEMOCAP) were utilized as generalization test sets, allowing us to cover more diverse audios.
4. For all audio samples, resampling at 16 kHz and conversion to mono were conducted. Then, if necessary, they were either cropped or prolonged up to 3 seconds to have consistent sizes as inputs for our pipeline.
5. Initially, only mean values were considered; then MFCC statistics, Pitch, Chroma and Spectral Contrast were added to our model.
6. We used stratified splits and balanced class weights for classical models. Deep models were stabilised with gradient clipping, mixed-precision training, label smoothing, and SpecAugment data augmentation. These methods reduced overfitting and controlled exploding gradients.
7. Custom PyTorch Datasets and Collate functions were defined in order to make handling sequential inputs easier in deep learning models.

Spring semester:

1. To make the process more efficient, some classes were eliminated, and similar classes were merged together according to the six core emotions outlined by the revised functional requirements.
2. New recordings have been added to a dataset containing samples from RAVDESS, TESS, and CREMA-D after cleaning and normalization.

3. The data has been split 80/10/10 train/validation/test using the speaker ID to avoid leakage and ensure the model generalizes better. All samples were trimmed to exactly 3 seconds, resampled to 16 kHz, and processed to mono.
4. A unified feature vector including log-Mel spectrogram, MFCCs (as well as their delta), chroma, spectral contrast, pitch statistics, and energy was created for compatibility with traditional and deep learning methods.
5. Among such approaches, the most suitable for our task was the multilayer perceptron trained on top of the 217-dimensional features. The MLP demonstrated the highest trade-off between performance on the test set (78.8% accuracy) and inference latency (0.3-0.4 seconds per clip) while keeping the relatively compact model size. The CNN-BiLSTM could not generalize to the test sample and required more computational power.
6. We used early stopping, gradient clipping, mixed precision, and SpecAugment during training to reduce the risk of overfitting and speed up the process. Non-functional targets required us to limit the target model size to 50 MB and keep RAM consumption below 2 GB.
7. We integrated the trained MLP with a feature extractor and label encoder into a Python script able to read and process live audio streams coming from a Raspberry Pi.

6.3 Challenges & Solutions

Fall semester:

1. Challenge: Since the initial datasets used different labelings and recording settings, it was not possible to build a unified model.
Solution: We normalized the emotions across the datasets by combining the similar categories of emotions, and the best compromise was represented by the RAVDESS dataset.
2. Challenge: An initial test with basic features provided only 43% accuracy, suggesting that the chosen features contain insufficient information.
Solution: We increased the number of features by adding the MFCC standard deviation and switched to nonlinear SVMs, which improved accuracy to about 72%.
3. Challenge: Initial LSTM models were unstable due to exploding gradients and overfitting.
Solution: Gradient clipping, warm-up learning-rate schedules, dropout regularization, SpecAugment, and early stopping helped to increase the LSTM generalization capacity.
4. Challenge: Rare emotions such as calm and neutral were often confused because of their similarity and unbalance between classes.
Solution: We used stratified splitting of the data and label smoothing technique, while eventually merging both classes to solve the problem of imbalance.

5. Challenge: Training with custom recordings was inefficient, as models achieved up to 5–20% accuracy on this dataset.

Solution: We concluded that the custom dataset was too noisy, and decided not to use it in future experiments.

6. Challenge: The hardware restrictions on Raspberry Pi required us to use models with reduced CPU usage.

Solution: For this, we preferred classical lightweight models and investigated the use of various compression techniques, such as quantization, pruning, TFLite format, and ONNX conversion.

Spring semester:

1. Challenge: Spring objectives required developing a single model with 80-85% accuracy across several corpora and running it in real time on a Raspberry Pi with only 2 GB of RAM, while maintaining a 50 MB model size limit.

Solution: Emotion classes were decreased to six. The dataset has been remade to include only high-quality recordings, and an MLP model was developed, achieving 78.8% accuracy while meeting the requirements for memory and real-time operation.

2. Challenge: The self-recording dataset collected during the Fall semester was of poor quality and monotonous, resulting in unreliable training.

Solution: All customized clips were recorded again with special attention being paid to quality and diversity. Then came RAVDESS, TESS, and CREMA-D preprocessing and integration. The labels were applied in all cases and the speakers were distinguished.

3. Challenge: Deep CNN-BiLSTM achieved good accuracy but was too large and failed to generalize to other datasets (overpredicted emotions, especially on the SAVEE dataset).

Solution: After trying classical models and several neural architectures, an MLP with log-Mel, MFCC, chroma, spectral contrast, prosody, and other features proved to be a decent model in terms of accuracy and resource consumption, delivering good results and being trainable.

4. Challenge: At the very beginning of the spring, the MLP model showed only slightly higher accuracy than the SVM baseline when trained solely on the RAVDESS dataset.

Solution: More corpora (CREMA-D and TESS), along with additional feature engineering, helped increase the MLP's accuracy to nearly 78.8%.

5. Challenge: Training and testing MLP on a Raspberry Pi with 2GB of memory puts strict restrictions on the accuracy and runtime of our model.

Solution: Mixed precision training was conducted. MLP parameters have been trimmed, and some expensive computations are performed beforehand, cached, and loaded directly

at runtime. The final pipeline takes around 3-second clips as input, calculates features, and makes predictions in 0.3-0.4 seconds per clip on average.

6.4 Teamwork & Roles

In our team's collaboration process, flexibility outweighed predetermined task allocation. Initially, the assignment of tasks followed the structure of the datasets used. Each member independently reviewed individual datasets, estimating their sizes, numbers of voices, emotional classes, and recording quality, and proposed a unified labeling strategy for all datasets. Some other members of the team were involved in modeling and creating the initial baseline model. As the project progressed, flexibility in teamwork was seen, because, rather than having fixed responsibilities among team members, the tasks were undertaken by all members depending on the needs at that particular stage. For example, while creating the baseline model, the team members worked together in enhancing features in the model.

Collaboration did not stop once tasks started being assigned. Members discussed the results and collectively decided on further steps. Collaboration tools allowed us to keep track of our progress and facilitate communication. There was no clear hierarchy in the project; leadership was determined by the project phase and could be exercised by different members at any time.

7. Evaluation

7.1 Evaluation Methodology

The solution was validated by checking against the functional and non-functional requirements which were outlined initially. Functional requirements such as classification accuracy was computed on a separate test partition using a speaker-independent 80/10/10 split across all six emotion classes. For non-functional aspects, model size under 50 MB, limiting RAM to 2 GB, and real-time inference, performance was measured on the Raspberry Pi 5. To further generalize, the model was evaluated on the SAVEE dataset, which was unseen during training.

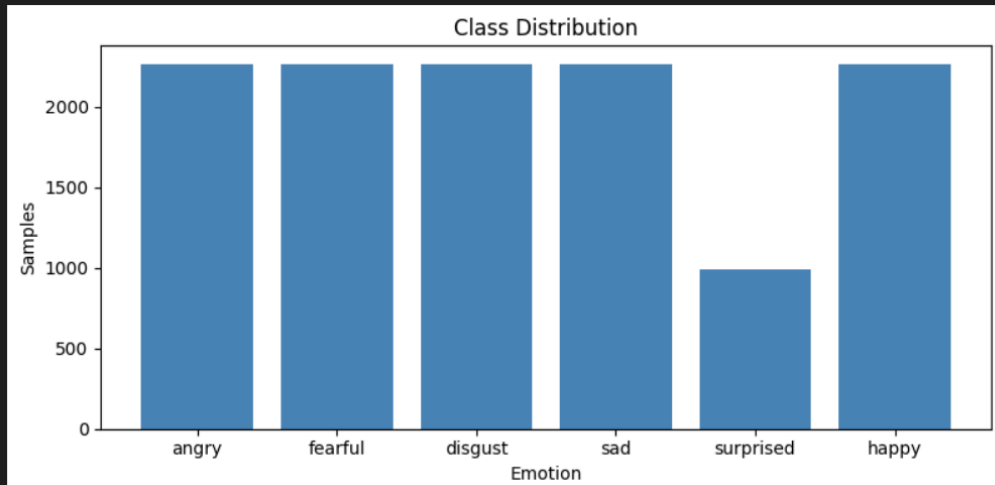
7.2 Data analysis

The final dataset was constructed from using 3 speech emotion recognition datasets: RAVDESS (1152 files), TESS (4800 files) and CREMA-D (6355 files). As shown in the figure, the dataset is balanced, each emotion totaling on average 2263 files, with the exception of "surprised", which has 992 samples. Thus, this imbalance dataset might affect the model output, however class weights were applied during training to partially compensate for this disparity.

```

[RAVDESS] Loaded 1152 files
[CREMA-D] Loaded 6355 files
[TESS]    Loaded 4800 files
Total: 12307 files
angry      : 2263
disgust    : 2263
fearful    : 2263
happy      : 2263
sad        : 2263
surprised  : 992

```



7.3 Performance Metrics

Model	Accuracy	Notes
Linear SVM (baseline)	43%	Mean MFCCs only
SVM	70-72%	Extended 37-dim vector
CNN-BiLSTM	71-72.69%	Fall semester best
MLP	78.8%	217-dim, merged dataset

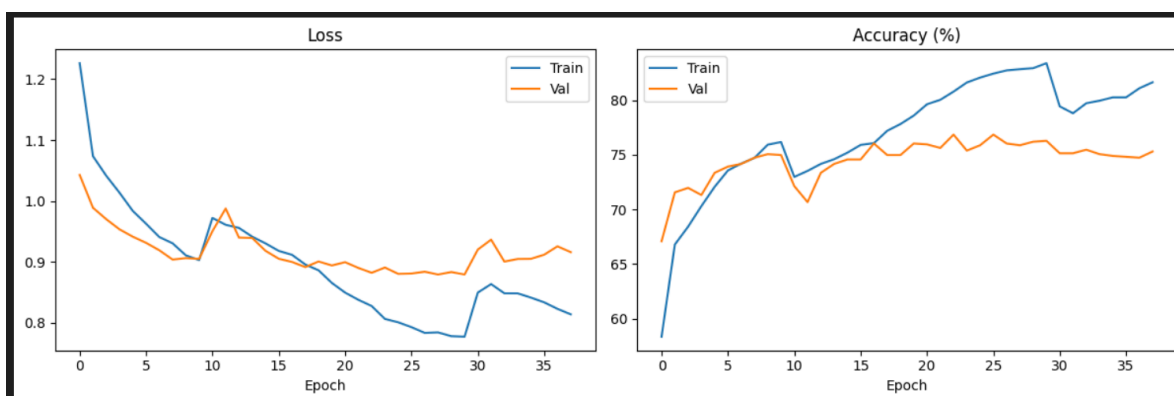
Non-functional results:

- RAM usage: 2 GB limit
- Model size: 50 MB limit

The final MLP reached 78.80% test accuracy. As shown in the confusion matrix, “surprised” and “angry” achieved the best performance, F1-score reaching 0.965 for “surprised” and 0.842 for “angry”, respectively. It is worth noting that “surprise” achieved a very high score, this result should be interpreted with caution considering the comparatively smaller sample size for that class. “Happy” and “fearful” showed the worst results, F1-score of around 0.71 and 0.75, “fearful” being confused with “happy” and “sad”, due to their similar pitch and energy characteristics. Finally, “disgust” showed an F1-score of 0.771, with the confusion toward “angry”.

MLP Test accuracy: 78.80%				
	precision	recall	f1-score	support
angry	0.826	0.859	0.842	227
disgust	0.807	0.739	0.771	226
fearful	0.711	0.717	0.714	226
happy	0.760	0.740	0.750	227
sad	0.762	0.805	0.783	226
surprised	0.960	0.970	0.965	99
accuracy			0.788	1231
macro avg	0.804	0.805	0.804	1231
weighted avg	0.788	0.788	0.788	1231

Figure below illustrates the training and validation loss and accuracy over almost 40 epochs. As seen in the graph both curves converge gradually, validation accuracy reaching stability around 70-75%, while training accuracy increasing up to approximately 85%, which indicates an overfitting. Considering the dataset diversity and speaker-independent split, the gap is acceptable.



```

=== MLP Results – SAVEE ===
Evaluated: 360 | Skipped: 0

```

	precision	recall	f1-score	support
angry	0.29	0.72	0.41	60
disgust	0.38	0.30	0.34	60
fearful	0.53	0.15	0.23	60
happy	0.29	0.17	0.21	60
sad	0.29	0.53	0.38	60
surprised	0.00	0.00	0.00	60
accuracy			0.31	360
macro avg	0.30	0.31	0.26	360
weighted avg	0.30	0.31	0.26	360

To see how well the model generalized, we used a completely unseen dataset of 360 files, titled as SAVEE, from four British English male speakers. As expected, accuracy dropped drastically to 31%, with F1-score being 0.26 on average. “Surprised” was not predicted for a single file, and other emotions performed poorly, respectively. Class “angry” was able to achieve a reasonable recall of 0.72, but, with low precision of 0.29, indicating the over-predicting.

The main reason for that outcome is the fundamental mismatch between the training datasets, which included only North American English speakers, and the testing dataset, SAVEE’s British English accent. The two differ in distinct prosodic and spectral patterns that the models were never exposed to during training. This is SER systems’ well-known limitation, which suggests that accent diversity in training data would be a meaningful part for improvement.

7.3 User Testing

The model was able to identify four emotions out of six defined and tested by all 5 team members speaking into the microphone connected to Raspberry Pi.

```

[OK] Model loaded | input_size=217 | classes=['angry', 'disgust', 'fearful', 'happy',
[MIC] Device #2: MacBook Pro Microphone | channels=1 | rate=48000
[MIC] Recording started (Ctrl-C to stop)

😞 SAD (80%) [disgust:80% | angry:13% | fearful:2%] 🇬🇧 | sad:11%

(.venv) (base) danamyktybek@Danas-MacBook-Pro senior_project_final % python3 inference.py
[OK] Model loaded | input_size=217 | classes=['angry', 'disgust', 'fearful', 'happy', 'sad', 'surprised'] | val_acc=73.43%
[MIC] Device #2: MacBook Pro Microphone | channels=1 | rate=48000
[MIC] Recording started (Ctrl-C to stop)

😨 FEARFUL (71%) [fearful:71% | disgust:11% | angry:8%] 🇬🇧 | angry:8%

```

```
(.venv) (base) danamyktybek@Danas-MacBook-Pro senior_project_final % python3 inference.py
[OK] Model loaded | input_size=217 | classes=['angry', 'disgust', 'fearful', 'happy', 'sad', 'surprised'] | val_acc=73.43%
[MIC] Device #2: MacBook Pro Microphone | channels=1 | rate=48000
[MIC] Recording started (Ctrl-C to stop)

🟡 DISGUST (89%) [disgust:89% | sad:4% | surprised:2%] █ angry:11% ]

(.venv) (base) danamyktybek@Danas-MacBook-Pro senior_project_final % python3 inference.py
[OK] Model loaded | input_size=217 | classes=['angry', 'disgust', 'fearful', 'happy', 'sad', 'surprised'] | val_acc=73.43%
[MIC] Device #2: MacBook Pro Microphone | channels=1 | rate=48000
[MIC] Recording started (Ctrl-C to stop)

🟡 ANGRY (82%) [fearful:82% | sad:10% | disgust:4%] █| disgust:8%]

(.venv) (base) danamyktybek@Danas-MacBook-Pro senior_project_final % python3 inference.py
[OK] Model loaded | input_size=217 | classes=['angry', 'disgust', 'fearful', 'happy', 'sad', 'surprised'] | val_acc=73.43%
[MIC] Device #2: MacBook Pro Microphone | channels=1 | rate=48000
[MIC] Recording started (Ctrl-C to stop)

😊 HAPPY (63%) [happy:63% | disgust:13% | angry:10%] █ | angry:15%]
```

As in the images above, the model was able to confidently produce predictions for 5 emotions, with sad, fearful, disgust, angry, and happy, except surprised. This emotion proved more difficult to detect in real time. Even though the model performed well on the samples of surprised emotion, live microphone input frequently resulted in misclassification. Due to the common key acoustic properties with both happy and fearful utterances, including elevated pitch, and increased energy. Thus, it is inherently challenging for models that rely on handcrafted acoustic descriptors to differentiate these emotion classes.

Overall, the live test showed the system functions on previously unseen voices in real time. However, accuracy on individual speakers was lower compared to testing, which is the expected result given the difference between studio recording and natural, real life speech.

8. Conclusion and Future Work

8.1 Key findings

This project demonstrated that a lightweight SER system can run in real time on a Raspberry Pi.

The key empirical finding is that classical and shallow neural architectures with sufficient handcrafted features, can match more complex models under real-world conditions. As discussed above, the final chosen model is MLP which achieved 78.8% accuracy outperforming all other architectures tested. It processes a 3 second audio clip in 0.3-0.4 seconds and fits within all hardware constraints. During the life testing real time operation of all six emotions were confirmed; however, due to the acoustic similarity to “happy” and “fearful” “surprised” emotion remains as the most difficult to classify.

Moreover, the importance of data quality over data quantity was established. The notable difference between the initial custom dataset that was recorded without controls and the latter custom dataset recorded with proper conditions can be observed. Great microphone positioning, quiet environment and expressive performance improved training stability.

However, testing on the SAVEE (unseen) dataset revealed that generalization across unseen speakers and recording conditions remains a key limitation.

8.3 Future Work

First priority for future work would be closing the gap of achieved and targeted accuracy level. It could be achieved using more training data across different demographics, speaker normalization, and finer prosodic features. Next, lightweight on device adaption should be also considered. If the model had the ability to finetune briefly on the target speaker’s voice, it could reduce misclassification in real world conditions. Also, extending the system’s functionality by replacing discrete emotion categories with dimensional representations would make the system more expressive and better suited for assistive technology.

8.4 Closing Statement

This project showed that speech recognition is achievable on constrained hardware. It is possible to obtain a system that processes live audio and correctly identifies the majority of emotional states with the right feature engineering and model selection. The pipeline and findings provide a practical foundation for future privacy preserving, on-device affective computing systems.

References

- [1] Schuller, B., Steidl, S., Batliner, A., Vinciarelli, A., Scherer, K., Ringeval, F., Chetouani, M., Weninger, F., Eyben, F., Marchi, E., Mortillaro, M., Salamin, H., Polychroniou, A., Valente, F., Kim, S. (2013) The INTERSPEECH 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism. Proc. Interspeech 2013, 148-152, doi: 10.21437/Interspeech.2013-56
- [2] Schuller, B., Rigoll, G., & Lang, M. (2003). Hidden Markov model-based speech emotion recognition. Proceedings of ICASSP 2003 (Vol. 2, pp. II-1–II-4). IEEE. <https://ieeexplore.ieee.org/document/1202279>
- [3] El Ayadi, M., Kamel, M. S., & Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3), 572–587. <https://doi.org/10.1016/j.patcog.2010.09.020>
- [4] Akçay, M. B., & Oğuz, K. (2020). Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. *Speech Communication*, 116, 56–76. <https://doi.org/10.1016/j.specom.2019.12.001>
- [5] Zhao, J., Mao, X., & Chen, L. (2019). Speech emotion recognition using deep 1D and 2D CNN LSTM networks. *Biomedical Signal Processing and Control*, 47, 312–323. <https://doi.org/10.1016/j.bspc.2018.08.035>
- [6] Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*.
- [7] Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), 357–366. <https://doi.org/10.1109/TASSP.1980.1163420>
- [8] Trigeorgis, G., Ringeval, F., Brueckner, R., Marchi, E., Nicolaou, M. A., Schuller, B., & Zafeiriou, S. (2016). Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network. Proceedings of ICASSP 2016 (pp. 5200–5204). IEEE. <https://ieeexplore.ieee.org/document/7472669>
- [9] Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., & Narayanan, S. S. (2008). IEMOCAP: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42(4), 335–359. <https://doi.org/10.1007/s10579-008-9076-6>

- [10] Cao, H., Cooper, D. G., Keutmann, M. K., Gur, R. C., Nenkova, A., & Verma, R. (2014). CREMA-D: Crowd-sourced emotional multimodal actors dataset. *IEEE Transactions on Affective Computing*, 5(4), 377–390. <https://doi.org/10.1109/TAFFC.2014.2336244>
- [11] Livingstone, S. R., & Russo, F. A. (2018). The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PLOS ONE*, 13(5), e0196391. <https://doi.org/10.1371/journal.pone.0196391>
- [12] Haq, S., & Jackson, P. J. B. (2011). Multimodal emotion recognition. In W. Wang (Ed.), *Machine Audition: Principles, Algorithms and Systems* (pp. 398–423). IGI Global. <http://kahlan.eps.surrey.ac.uk/savee/>
- [13] Ververidis, D., & Kotropoulos, C. (2006). Emotional speech recognition: Resources, features, and methods. *Speech Communication*, 48(9), 1162–1181. <https://doi.org/10.1016/j.specom.2006.04.003>
- [14] Lalitha, S., Geyasruti, D., Narayanan, R., & Shravani, M. (2020). A literature survey on speech emotion recognition. *Proceedings of ICTC 2020*. <https://doi.org/10.1109/ICTC49870.2020.9289227>
- [15] Lee, H., & Nadeem, M. (2025). Toward Efficient Speech Emotion Recognition via Spectral Learning and Attention. *arXiv preprint arXiv:2507.03251*.
- [16] Mirsamadi, S., Barsoum, E., & Zhang, C. (2017). Automatic speech emotion recognition using recurrent neural networks with local attention. *Proceedings of ICASSP 2017*, 2227–2231. <https://doi.org/10.1109/ICASSP.2017.7952552>
- [17] Dhavale, M., & Bhandari, P. S. (2022). Speech emotion recognition using CNN and LSTM. *Proceedings of ICCUBE 2022*, 1–3. <https://doi.org/10.1109/ICCUBE54992.2022.10010751>
- [18] Onyekwelu-Udoka, L., Islam, M. S., & Hasan, M. S. (2025). Emotion Detection in Speech Using Lightweight and Transformer-Based Models: A Comparative and Ablation Study. *arXiv preprint arXiv:2511.00402*.
- [19] Waleed, G. T., & Shaker, S. H. (2025). Speech Emotion Recognition on MELD and RAVDESS Datasets Using CNN. *Information*, 16(7), 518. <https://doi.org/10.3390/info16070518>
- [20] Kotkar, S. (2025). Real-Time Emotion Recognition with CNN and LSTM. Preprints. <https://doi.org/10.20944/preprints202505.0672.v1>

- [21] Bhanbhro, J., Memon, A., Lal, B., Talpur, S., & Memon, M. (2025). Speech emotion recognition: Comparative analysis of CNN-LSTM and attention-enhanced CNN-LSTM models. *Signals*, 6(2), 22. <https://doi.org/10.3390/signals6020022>
- [22] Aggarwal, A., Srivastava, A., Agarwal, A., Chahal, N., Singh, D., Alnuaim, A. A., Alhadlaq, A., & Lee, H.-N. (2022). Two-Way Feature Extraction for Speech Emotion Recognition Using Deep Learning. *Sensors*, 22(6), 2378. <https://doi.org/10.3390/s22062378>
- [23] Barhoumi, C., & BenAyed, Y. (2025). Real-time speech emotion recognition using deep learning and data augmentation. *Artificial Intelligence Review*, 58, 49. <https://doi.org/10.1007/s10462-024-11065-x>
- [24] Middya, A. I., Nag, B., & Roy, S. (2024). Effective MLP and CNN based ensemble learning for speech emotion recognition. *Multimedia Tools and Applications*, 83, 83963–83990. <https://doi.org/10.1007/s11042-024-19017-x>