Universiteit Leiden

Master Computer Science

Classification of Vocal Intensity Categories using ResNet

Name: Lin He
Student ID: 3567060
Date: 27/01/2025

Specialisation: Artificial Intelligence

1st supervisor: Dr. Erwin Bakker
2nd supervisor: Prof. dr. M.S. Lew

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Classification of Vocal Intensity Categories using ResNet

Lin He

February 2, 2025

Abstract

This paper proposes a simplified machine learning method for vocal intensity categorization. The vocal intensity in a vocal recording may contain hidden physiological signals involving someone's health and emotions. Vocal health reveals the condition of the vocal ligaments and any potential or existing problems in articulation. Unlike Vocal Pressure which can be measured directly by a sound level meter, the measurement of the vocal intensity is more complicated. In vocal intensity measurements, a standard calibration signal is required that can be used together with the vocal signal to quantify the measured sound pressure level(SPL) signals into vocal intensity levels. Consequently, an audio measurement without SPL cannot assure the actual vocal intensity. On the other hand, machine learning methods have shown to provide opportunities to classify vocal intensity in recordings without calibration information. In this paper, research is conducted on the application of different feature extraction methods (Spectrogram, Mel-spectrogram, MFCC) and classification models (ResNet) for classifying the vocal intensity categories (soft, normal, loud, very loud), and study the influence of calibration methods. It is shown that the proposed simplified model obtains a 71% accuracy on the Aalto Vocal Intensity Dataset.

1 Introduction

The speech signed contains a wide range of different types of information. Not only does it contain the speech contents, i.e., the human language information, but also hidden information in the audio waves if carries information on the the speakers' status. As such vocal intensity can give clues on the distance between two speakers, environmental noise gives information on the environment of the speaker, e.g., indoors or outdoors, the pitch and loudness carry information on the emotions of the speaker, etc. To explore the hidden information in the speech signal, it is crucial to have insight in the theory of sound production and sound transformation.

1.1 Vocal Pressure, Vocal Power and Vocal Intensity

A vocal source can be seen a radiative power, which ineffect causes the vocal pressure. The vocal pressure is a scalar that directly reflects on the effect that adds to the environment. But when measurement at a distance it does not describe the state of the power source. To evaluate the vocal source certain special methods are required to evaluate the vocal intensity, the sound waves are measured as power per unit area in a direction perpendicular to the area.

In the equation form, the intensity I is defined as:

$$I = \frac{P}{A} = \frac{(\Delta p)^2}{2\rho v_w} \tag{1}$$

where P is the power through an area A. Δp is the pressure amplitude in units of pascals(Pa or N/m^2), ρ is the density(kg/m^3) of material of the sound wave travels, and v_w is the speed(m/s) of sound in the medium. With the relationship in the equation, we acknowledge that the sound wave is produced by vibration, and the more air is locally compressed when conducting the sound, the higher the pressure amplitude.

1.2 Applications of Vocal Intensity

The vocal intensity is a direct characteristic of vocal source, which is difficult to measure in isolation. The vocal intensity in general is determined by measuring the vocal pressure at a certain distance from the speaker. The vocal pressure can only measure the vector scalar value of a certain point on a convergence. This scalar value includes also all other vocal sources in the current sound field. Le., it can be located in an open space or a room containing hundreds of other sound sources. Consequently, the measurement of vocal pressure requires specific environments like a silent recording room to collect the desired sound levels, which can be impractical. In general, the sound intensity measurement will take place anywhere. In those cases the sound intensity measurement can be adopted for a specific or individual sound source. For example,

using noise cancellation such that the noise in the background will have no influence when estimating the intensity.

In a conversation, the speaker's vocal intensity will vary during a sentence. The changing vocal intensity can be affected by a great number of reasons, such as communication goals, emotions expressed, environmental influences, and health conditions as mentioned above. Emphasizing the important content, softening the voice for intimacy, competing with environmental noise, and reflecting the speakers' health, age, etc., are all reflected in the vocal intensity of the speaker.

Vocal intensity and collect effective data in complicated environments, which can is used in many scenarios. One common usage of vocal intensity is in manufacturing, such as finding the noise sources in the vehicle design to give a better driving experience for passengers. Recently, vocal intensity has become an important topic in health. Vocal health can be an important sign of human health. When a vocal is produced, the muscles and tissues compress the air passing through the larynx. Vocal source can be influenced by human health conditions. Vocal health level can reflect aspects of someone's health. This ranges from the potential health damage of imbalanced system hydration[1] to more severe diseases like Parkinson's disease[2]. Thus, vocal intensity is a significant sign of health marker, and measuring and evaluating the vocal intensity can play a major role in disease prevention. Another significant usage of vocal intensity is emotion analysis. There is evidence that vocal intensity modification quantitatively affects the emotionality in vocal emotion[3].

However, measuring vocal intensity requires professional devices that record vocal data with calibration. Most recording devices are not calibrated. Most of the recordings from daily recording devices like phones lack calibration data, nor do they take place in a silent room at a fixed distance. This results in difficulties in collecting calibrated data and a correct analysis afterwards. Due to most of the data not being recorded with calibration but with non-standard amplitude scales, the measurement of vocal intensity can be used to describe speakers' biomarking more precisely. Without calibration also means that the measurement of the intensity is computationally impossible. In recent years, researchers have introduced machine learning (ML) based methods for vocal intensity classification using calibrated recordings.

In this work we propose a novel more straightforward method using deep neural networks (DNN) for vocal. The contribution of this research are:

- A simplified method using traditional spectrograms, mel-spectrograms and MFCCs as input combined with a more fine-grained FFT window-step-size to obtain ResNet features for vocal intensity classification.
- An evaluation of our simplified method for vocal intensity classification against SOTA methods based on transformers like Wav2Vec2, HuBERT, AST to extract

features and SVM as a classifier.

- Our simplified method shows an improved vocal intensity classification accuracy of 71% while having a reduced computational complexity.

The rest of the paper is organized as follows: In Section 2, we discuss the past research on many vocal intensity categorize methods. In Section 3, we introduce the fundamentals used in the baseline methods and our methods, and in addition, we explain the evaluations. The used benchmark dataset for vocal intensity category classification is described in Section 4. In Section 5, the baseline methods are described and in Section 6 our method is described in detail. In Section 7, the experimental setup and results are given in Section 8. Finally, in Section 9, conclusions, discussions, and future research directions are discussed.

2 Related Research

The studies on vocal intensity classification have focused on detecting single or binary intensity classes where the targets of vocal intensity classification are certain specific emotions or speech expressions. For example, single vocal intensity classification for whisper detection[4] and shouted speech detection [5], or the binary classification [6] of cries and whispers. These papers utilized acoustic engineering features and signal and image processing technologies to process the data and determine the vocal intensity feature. The author of the paper [6] used the source characteristics like Mel-frequency Cepstral Coefficients(MFCC), Teager Energy Operator(TEO), Voiced/Unvoiced Frequency(VUF), Voice Quality (VQ) etc to classify cries and whispers. In paper [7], the author brought up five speech modes that categorized the speech to classes of whispered, soft, natural, loud and shouted respectively. The proposed method uses Gaussian mixture models(GMM) and the Supported Vector Machines(SVM) to classify speech into the five models. The dataset used in the paper, the data was inadequate for not only limited in size and only contained male vocal data. Consequently, more recent researchers in the field of automatic speech intensity classification foucused on incorporating multiple intensity categories with more advanced models and experiment on larger datasets including data with various physiological property regarding genders, ages, etc.

Recent advancements in deep learning have popularized the use of pre-trained models across various areas of speech technology [8], [9]. These models are particularly valuable in domains like paralinguistics, where speech datasets are often limited. Pre-trained models allow for the application of deep neural networks initially trained on tasks requiring large datasets (e.g., automatic speech recognition) to areas with smaller training datasets, such as paralinguistics. Different methods for leveraging pre-trained models have been explored, including feature extraction, fine-tuning, and autoencoder implementations [10], [11], [12]. Notable examples of their application include emotion recognition [13], stuttering detection [9], and analysis of pathological speech [14], demonstrating their potential in paralinguistics.

Specifically, the methods introduced by the paper [15] and [16] are the baseline methods used in our evaluation. The first method used traditional acoustic methods (spectrogram, mel-spectrogram, MFCC) and is similar to our proposed but uses an SVM for classification while several ResNet DNNs. The second method used state-of-the-art transformer-based models (Wave2Vec2, HuBert, AST) to extract the acoustic information and classify it with SVM. In this paper, the methods we use are clearly inspired by both of these approaches. We make the acoustic information obtained by the traditional acoustic method more effective by changing the parameters and employing ResNet further for vocal intensity classification to enhance the features.

3 Fundamentals

In this section, we will introduce the fundamental concepts, models and algorithms involving both the baseline methods and the improved method to have a better understanding of the principles of the methods.

3.1 Speech Feature Extraction

Audio features can be extracted and represented vectors as multidimensional. The baseline methods both use the spectrogram of the audio as input. The horizontal axis shows the time and the vertical axis shows the frequencies of the input audio.

In many real-world applications, signals are non-stationary, meaning their frequency content changes over time. Examples include speech, music, and biological signals. A global Fourier Transform does not capture these time-varying characteristics because it assumes that the signal's frequency content does not change over time.

To address this limitation, we use the Short Time Fourier Transform(STFT) which processes the input audio signal in short, overlapping time segments, making it possible to examine localized frequency content. This localized analysis is achieved by applying the Fourier Transform to successive segments of the signal.

The STFT of a continuous-time signal $x(t)$ is defined as:

$$\mathcal{X}(t,\omega) = \int_{-\infty}^{\infty} x(\tau)w(t-\tau)e^{-j\omega\tau}\,d\tau \tag{2}$$

Here:

- $\mathcal{X}(t,\omega)$ is the STFT of $x(t)$.
- τ represents time.
- ω represents angular frequency.
- $w(t)$ is the window function that localizes the signal in time.

In practice, the STFT is applied to discrete-time signals, where the discrete STFT is defined as:

$$\mathcal{X}(m,k) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j2\pi kn/N} \tag{3}$$

Here:

- $\mathcal{X}(m,k)$ is the STFT of the discrete signal $x[n]$.

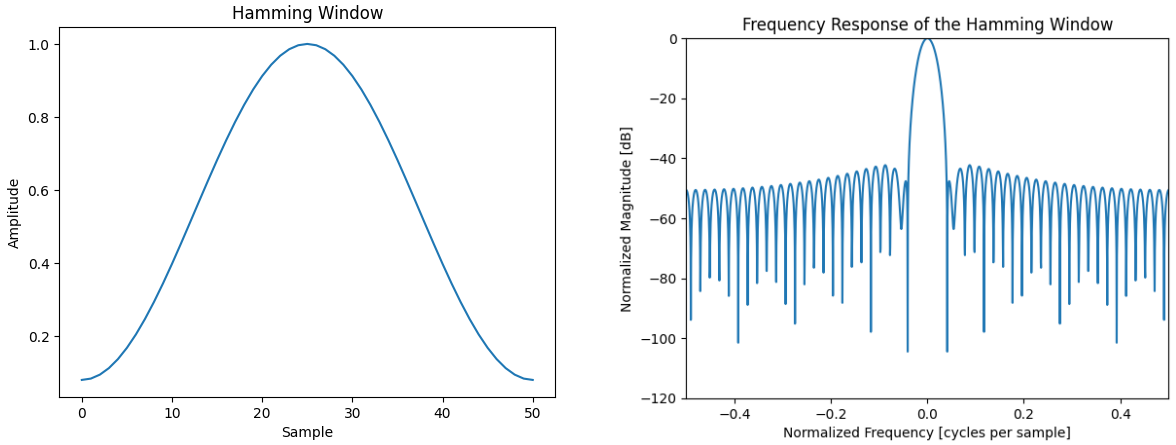
- n represents discrete time.
- m is the time-shifting index.
- k is the frequency bin index.
- N is the length of the FFT (Fast Fourier Transform).

The window function $w(t)$ plays a crucial role in the STFT. It defines the segment of the signal to be analyzed and shapes it to minimize edge effects. Common window functions include the Hann, Hamming, and Gaussian windows, each with specific properties that influence the trade-off between time and frequency resolution. In our research we will use Hamming window with a window size of 25ms period length.

A fundamental concept in the STFT is the trade-off between time and frequency resolution, governed by the Heisenberg Uncertainty Principle. A narrow window provides better time resolution but poorer frequency resolution, as it captures shorter segments of the signal. Conversely, a wider window improves frequency resolution but reduces time resolution, as it captures longer segments.

This trade-off is a critical consideration when selecting the window length for a particular application. For instance, in speech processing, a balance must be struck between capturing rapid changes in speech sounds and providing sufficient frequency resolution to distinguish different phonetic elements.

3.1.1 Windows



(a) Hamming Window

(b) Frequency Response of Hamming Window

Figure 1: Hamming window and its frequency response

The Hamming window is defined as

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{M-1}\right), \quad 0 \leq n \leq M-1 \tag{4}$$

The Hamming was named for R. W. Hamming and is described in Blackman and Tukey[17]. It is recommended for smoothing the truncated autocovariance function in the time domain. The frequency response of the Hamming window demonstrates its ability to reduce spectral leakage in signal processing. The central peak, known as the main lobe, contains most of the energy and determines the frequency resolution, while the smaller side lobes represent unexpected energy leakage into other frequency bands. Compared to the rectangular window, the Hamming window achieves a side lobe rejection of about -42dB, which significantly reduces spectral leakage. This trade-off results in a slightly wider main band with moderate frequency resolution but excellent artefact suppression. Due to the uniform symmetry of the Hamming window in the time domain, its response is symmetrical around the zero point. These characteristics make it ideal for applications like spectral analysis and digital filter design, where balancing resolution and leakage reduction is critical.

3.1.2 Spectrogram

A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies over time. In this paper, the spectrogram is generated from pre-processed audio clips in the dataset.

Figure 2 illustrates the process of how the audio signals are processed into spectrograms. The audio signal will first be framed by the windows and segmented into smaller segmentation pieces. Then for each segmentation, there will be an FFT process upon the segmentation signal, in the spectrograms' case, the FFT process is STFT. The spectrogram is made of a group of STFT series that are converted from segmentations. As the figure shows, the spectrogram can be demonstrated in a 3d spectrogram, which vividly explores the data relationship between Time, Frequency and Power. The Spectrogram is the 3d spectrogram's overlook perspective, which is the project of Power on Time and Frequency. In the baseline methods, the window length is 25ms and the overlap length is 5ms. In the improved method, we keep the window length the same but adjust the overlap length to 20ms.

3.1.3 Mel-spectrogram

The Mel scale is a perceptual scale of which the listeners judge pitches that are at equal distances from each other. It is based on how humans perceive sound frequencies, which are not linear. Instead, humans perceive pitch logarithmically, which means that the difference in pitch between low frequencies is greater than the difference in pitch

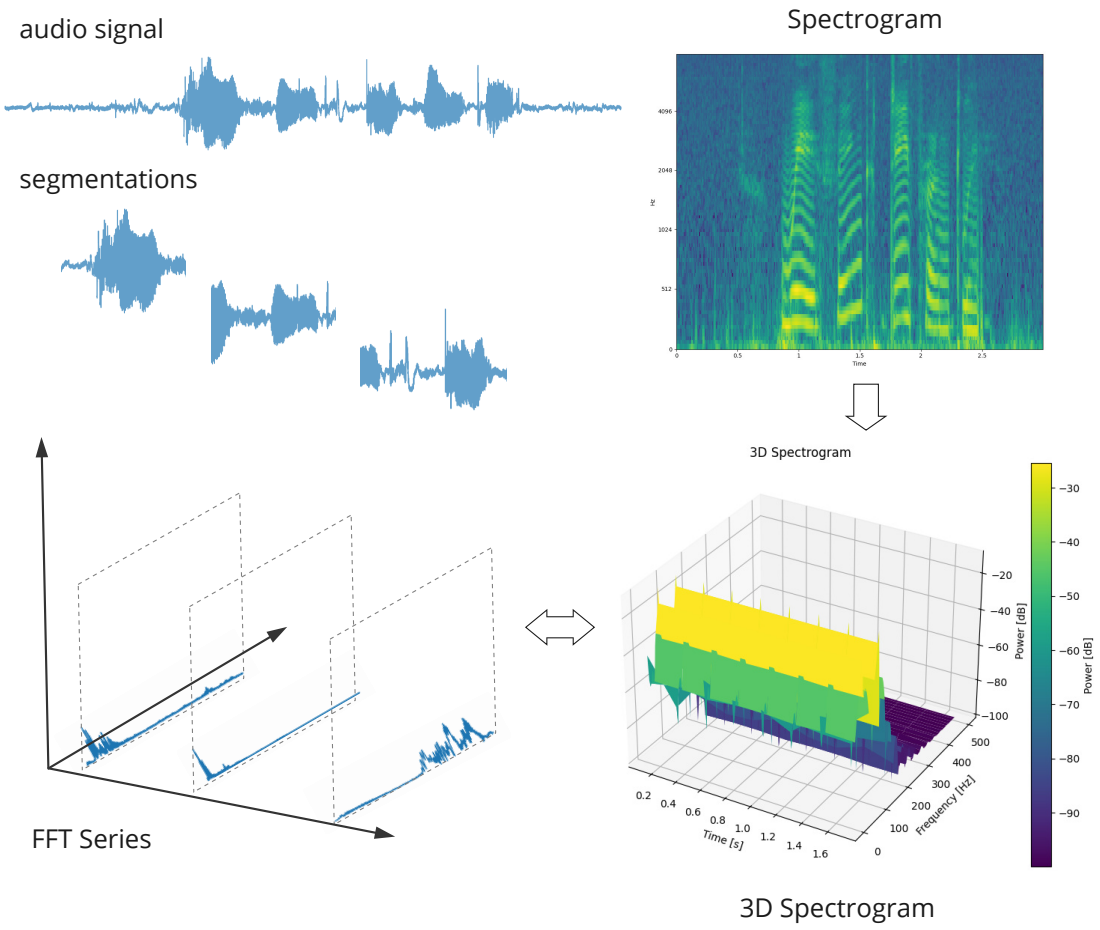


Figure 2: The transform from audio signal to spectrogram

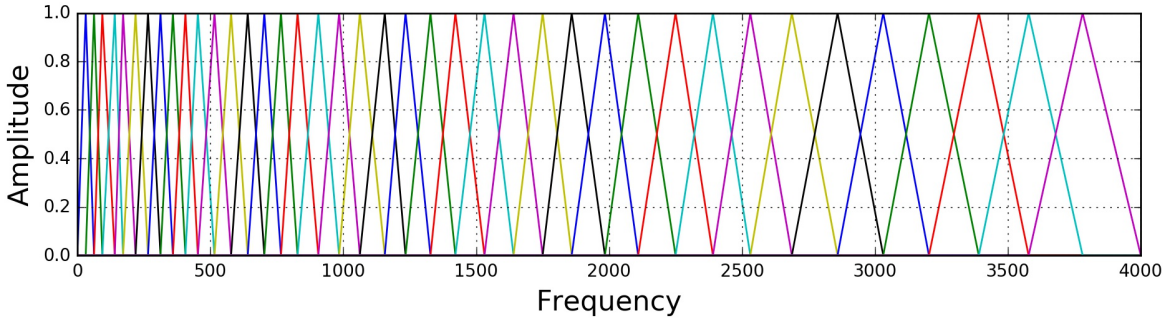


Figure 3: Mel-filters

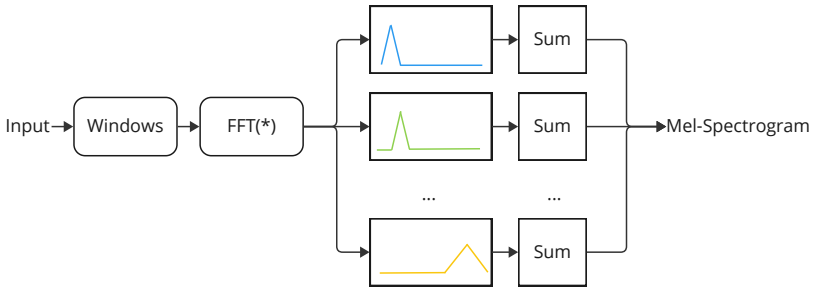


Figure 4: The Mel-spectrogram Algorithm

between higher frequencies.

The formula to convert a frequency f in Hertz to Mel scale is:

$$M(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \tag{5}$$

Conversely, the formula to convert the Mel scale back to a frequency in Hertz is:

$$f(M) = 700 \cdot \left(10^{\frac{M}{2595}} - 1\right) \tag{6}$$

To transform the mel-frequencies to mel-spectrogram, apply the Short-Time Fourier Transform (STFT) to the audio signal to obtain its frequency representation over time. Transform the linear frequency spectrogram into mel-scale with the mel filter bank as depicted in Figure 3. The Mel-filters consist of a group of triangle filters that map to the FFT vectors. Each vector in FFT will only be filtered by the corresponding as Figure 4 demonstrated. The output of the mel-filter will be summed up to create a mel-spectrogram.

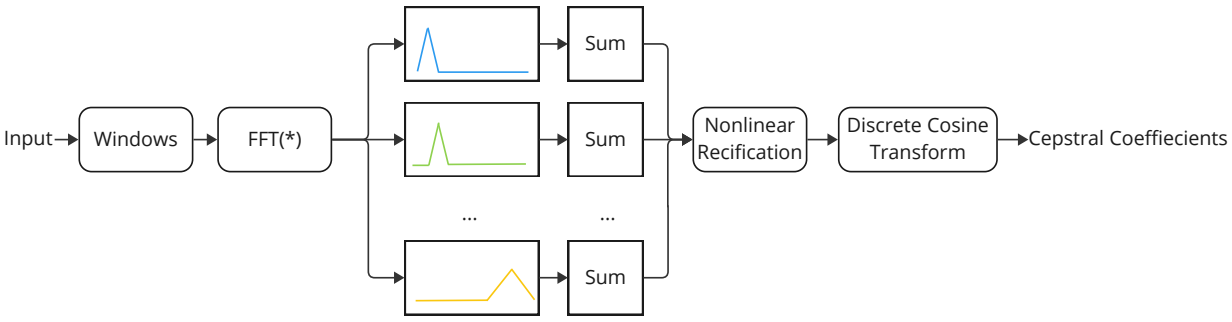


Figure 5: The MFCC Algorithm

3.1.4 MFCC

Mel-Frequency Cepstral Coefficients (MFCCs) [18] are an important feature extraction technique in audio signal processing, especially in automatic speech recognition (ASR). MFCCs aim to capture the short-term power spectrum of a sound signal in a way that mimics human auditory perception. This is achieved through a series of steps involving the transformation of the signal from the time domain to the frequency domain, mapping frequencies to the Mel scale, and applying cepstral analysis.

The first step is pre-emphasis, which applies a high-pass filter to the audio signal to amplify high-frequency components. This can be expressed as:

$$y(t) = x(t) - \alpha x(t-1) \tag{7}$$

where $x(t)$ is the input signal, $y(t)$ is the output signal, and α is a pre-emphasis coefficient (typically $\alpha = 0.97$). The signal is then divided into overlapping frames. Each frame is windowed using a Hamming window to reduce spectral leakage:

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \tag{8}$$

where N is the frame length. The Fast Fourier Transform (FFT) is applied to each windowed frame to obtain the frequency spectrum:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \tag{9}$$

The power spectrum is then computed as: $P(k) = |X(k)|^2$. The power spectrum is passed through a Mel filter bank to model human auditory perception.

The Mel scale is defined as:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{10}$$

The filters are triangular and spaced linearly on the Mel scale as figure 3. The logarithm of the filter bank energies is taken to compress the dynamic range:

$$\log E_m = \log \left(\sum_{k=0}^{K-1} P(k) H_m(k) \right) \tag{11}$$

where $H_m(k)$ is the m-th Mel filter.

The final step is applying the Discrete Cosine Transform (DCT) to obtain the MFCCs:

$$c_n = \sum_{m=0}^{M-1} \log E_m \cos \left[\frac{\pi n (2m+1)}{2M} \right] \tag{12}$$

MFCCs are widely used in ASR systems due to they are effective in capturing perceptually relevant features of speech. They are also applied in tasks like music genre classification, speaker identification, and environmental sound recognition. Although MFCCs are robust and computationally efficient, they are sensitive to noise, so noise reduction techniques should be used for preprocessing the audio data. In conclusion, MFCCs provide a powerful and widely-used method for audio feature extraction, which is closely aligned with human auditory perception. Their integration into various audio processing systems highlights their importance in the field.

3.2 Machine Learning Models

In this paper, several machine-learning models are adopted for feature extraction and classification. In this section, we will introduce all the models that are used in the baseline models and our proposed method.

3.2.1 SVM

Both baseline models use SVM with radial basis function as the radial basis kernel functions as a classifier. Support Vector Machines(SVM) [19] are machine learning method that separates the data points into different classes. SVM is a supervised machine learning algorithm that intends to separate data points with the optimal boundaries.

A linear SVM classifier is a 2-D classifier that separates the data points into two classes with a hyperplane. There are other kernel functions like Polynomial, Gaussian, and Radial Basic Function(RBF) for calculating more complex situations. The image6 illustrates the process of how to find the optimal classification. The red line is the hyperplane to separate the space, where the support vector w is the minimum distance of data points toward the hyperplane.

Multi-classification SVM is widely used in categorical tasks. The multi-classification SVM include the OneVsAll(OvA) classification and the OneVsOne(OvO) classification.

The OneVsAll approach requires training binary SVM classifiers for each class, where a single training separates one class as positive and the rest classes as negative. The OneVsOne approach will compare each class with the rest of the classes one class by one class until all the classes have trained once.

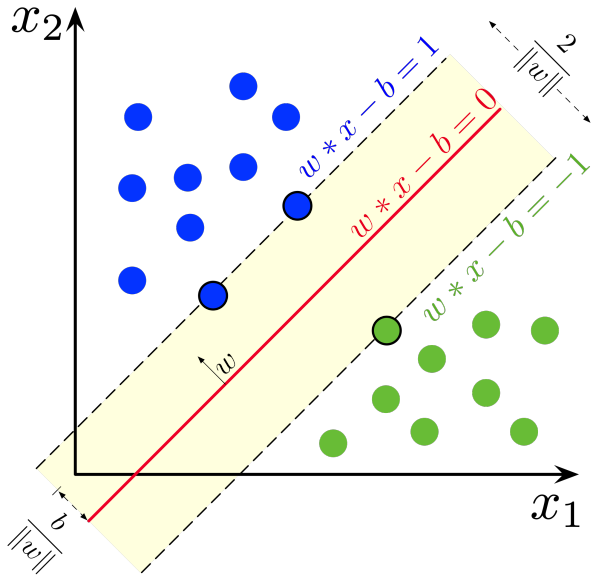


Figure 6: A simple linear SVM example [20]

The SVM functions as a classifier in the classification task with the extracted vocal intensity data.

3.2.2 Wav2vec2

In the second baseline method[16], Wave2vec2 and HuBERT are used as feature extractors. Wav2Vec2 [21] is a model developed by Facebook AI (Meta AI) for self-supervised learning on speech processing. The main purpose of Wav2Vec2 is to transform raw audio waveforms into meaningful representations, making it particularly useful in tasks like automatic speech recognition, audio classification, and other audio processing tasks.

Wav2Vec2's feature extraction is very effective for audio classification because it learns a nuanced hierarchical representation directly from the raw waveform. Using convolutional layers to capture short-term and long-term dependencies generates contextual embeddings that can be valuable for distinguishing between different classes. Wav2Vec2 uses masked prediction targets in self-supervised pre-training to develop robust context-aware features for a wide range of audio domains, from speech to ambient sound, without the need for manual feature engineering. This end-to-end learning approach enables Wav2Vec to capture subtle nuances in audio to provide a finer and more

efficient representation for classification.

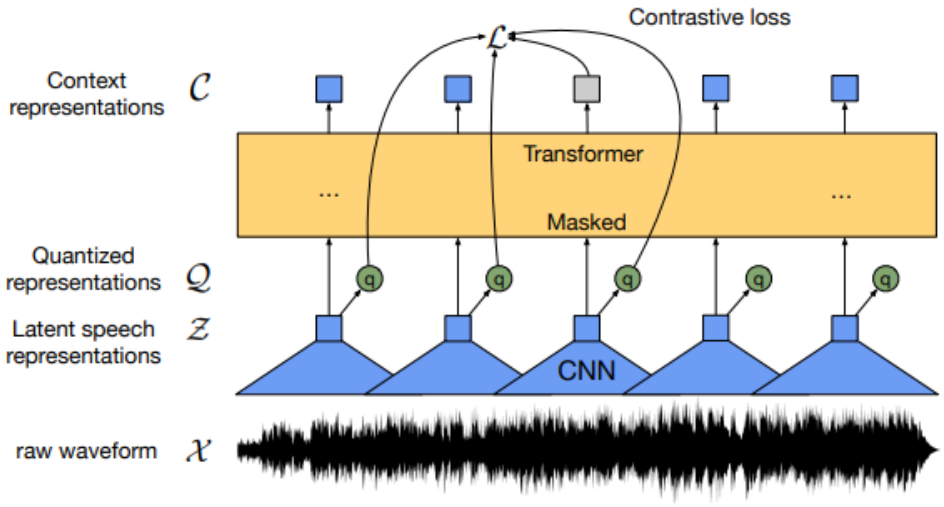


Figure 7: The joint learning of contextualized speech process on Wav2vec2 [22]

As the figure 7 shows the Wav2Vec2 model structure. The model consists of numerous speech units that are shorter than a phoneme, which eliminates the interference from the recording environment. The encoder for each unit is a Convolutional Neural Network (CNN) which converts the raw waveform into latent space, with the audio representations of 25ms each. This structure motivates the model to focus on the main feature of the raw data. The data from the latent space will be sent to the quantizer and transformer. The transformer will take the half-masked data and append information about the audio sequence data. The output of the transformer can be used in comparison tasks. In this paper, Wav2vac is used as a feature extractor for extracting the vocal intensity feature. After the extraction and encoding process, there will be a classifier(SVM) waiting for class prediction instead of a decoder structure like Connectionist Temporal Classification(CTC) as in the original paper for audio content recogition.

3.2.3 HuBERT

Hidden-Unit BERT(HuBERT)[22] is a self-supervising speech model inspired by the Wav2vec model above. HuBERT follows a similar structure as Wav2vec2 as figure 8. Both of Wav2vec2 and HuBERT have CNN encoder extracting the feature from the raw audio data followed by a Transformer.

HuBERT's clustering will add on most influence on the classification tasks in this paper's experiment. The first step in training HuBERT, the goal is to extract the hidden units from raw audio waves. The k-means clustering is utilized to separate the unit

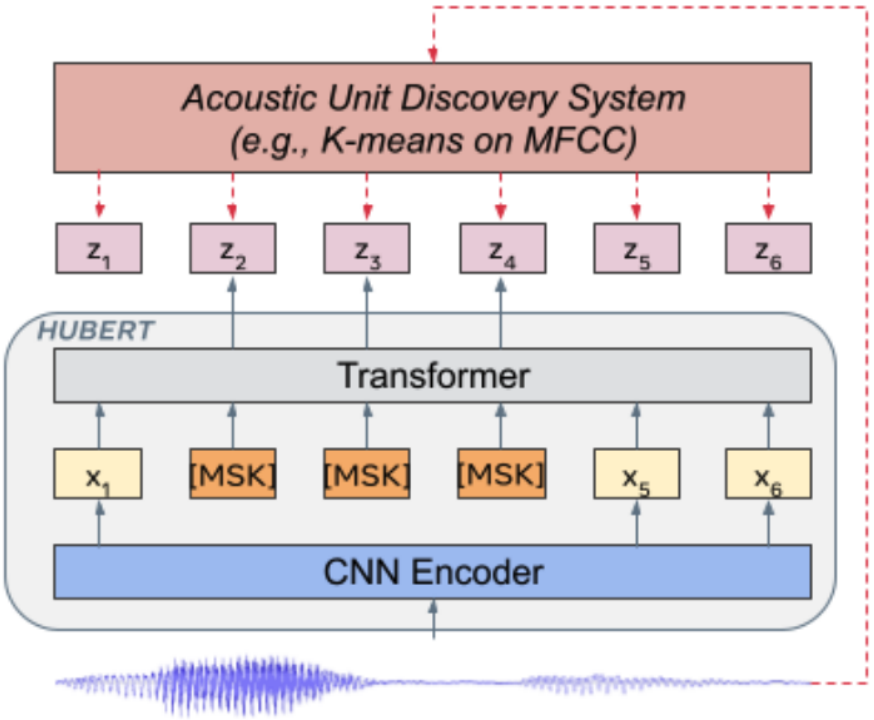


Figure 8: The model structure of HuBERT [**?**]

segmentation of audio into K-clusters. All the clustered segments will be signed to a unit label. Then the hidden units will be mapped to an embedding vector that will be used for prediction in the next step. CNN encoder isn't the only choice for feature extraction. MFCC features can also be used as the extractor before clustering.

However, there are several differences: The encoder in HuBERT framing is in 20ms, and encoded features are randomly masked. HuBERT adopts the cross-entropy loss just as BERT, while the Wav2vec2 uses combined loss that sums contrastive loss and diversity loss, which simplifies the training process and robustness.

Similar to Wav2vec2's usage in the experiment. The HuBERT model in the experiment was also followed by an SVM for classification.

3.2.4 AST

The AST is used as the feature extractor in the baseline2 method. The Audio Spectrogram Transformer (AST)[23] is a pure attention-based model for audio classification. The figure 9 shows the progress of how the AST model predicts attention mechanism resembling Vision Transformer(ViT)[24].

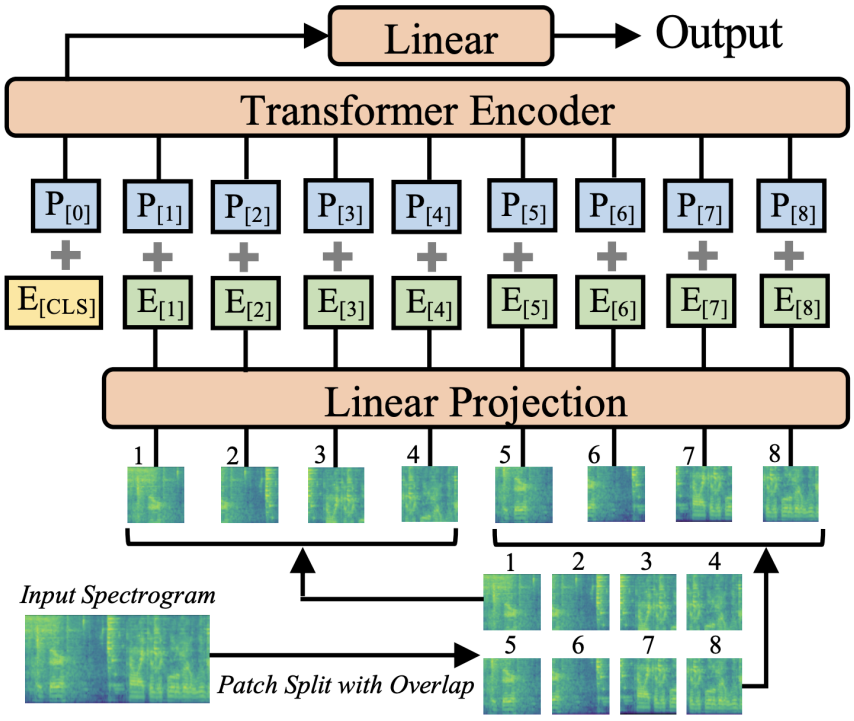


Figure 9: The AST model architecture [23]

As the input for the model is an image, the mel-spectrogram of the raw audio is used as the input. Just like ViT, AST will split the spectrogram into small patches of the size of 16×16 with an overlap of 6 in time(x-axis) and frequency(y-axis) dimensions. Each small patch will be firstly sequential and flattened to a 1-D patch embedding of size 768 by linear projection. After the linear projection, a trainable positional embedding is added on each patch embedding to ensure the sequence can be read by the Transformer.

3.2.5 ResNet

In our proposed simplified method for Vocal Intensity classification, we use several versions of ResNet (ResNet34 and ResNet101) for feature extraction and improvement.

Residual Neural Network(ResNet)[25] is a very classic deep learning CNN model that arises from ImageNet[26]. Figure 10 depicts the residual building block in ResNet. The residual building blocks are the major components of ResNet. The ResNet facilitates the training of deep neural networks by using skip connections among layers to mitigate the vanishing gradient problem and ensure the effective learning of identity mappings. The stacked non-linear layers are denoted as $F(x)$, where identity x is the input of the residual block. The desired underlying mapping $H(x) = F(x) + x$, which the shortcut connections feedforward the identity mapping, and the output of stacked non-

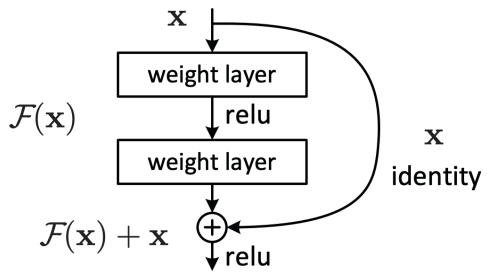


Figure 10: A residual building block [25]

linear layers's output and the identity are added together to the next block. ResNet offers great improvements by addressing the vanishing gradient problem so enabling scalability to deep-level networks, and achieving excellent classification performance and high computational efficiency.

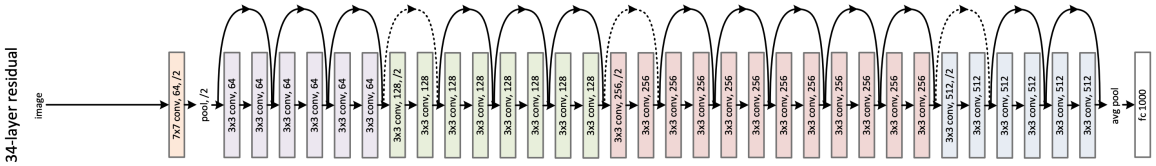


Figure 11: The architecture of ResNet34 [25]

The ResNet is built by multiple of such residual blocks. Figure 11 illustrates the ResNet34 architecture. The residual blocks connect one by one when the output size of the previous block and the input size of the next block is the same. There are 2 options for increasing the dimensions, one is using an extra zero entries padded, another is mapping projection shortcut, and both of the options are performed with a stride of 2.

Owing to the flexibility of the residual blocks, the ResNet can be shaped into many layers. The common ResNet are 18-layer, 34-layer, 50-layer, 101-layer, 152-layer, etc. Just as we discussed above, the ResNet consists of residual blocks that can be added to any layer as wished. However, in this paper, we will use the common ResNet layers to simplify the experiment and demonstrate the generalization of the model for academic purposes.

4 Dataset

In our experiments, we use the dataset Aalto Vocal Intensity Database(AVID)[27] (http://research.spa.aalto.fi/projects/intensity_category_db/) dataset, a spe-

cial classification dataset often used for vocal intensity research. The dataset collection recorded the vocal intensity with a DPA 4065-BL headset condenser microphone and an EG2-PCX2 electroglottograph(EGG), a calibrator, sound card, and a laptop with the Audacity program. During the recording sessions, the microphone was positioned 5 cm from the centre of the speaker's lips, and the electrodes were adjusted according to the placement indicator on the EGG device. Both signals were transmitted through an RME Babyface sound card and captured using the Audacity software at a sampling frequency of 44.1 kHz. Following setup and calibration, the speakers were instructed to sequentially produce utterances at four designated vocal intensity levels: soft, normal, loud, and very loud.

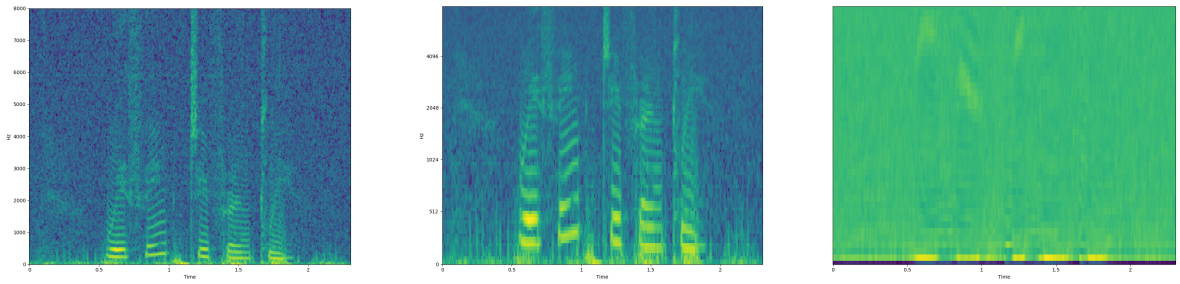
The dataset consists of two speaker major tasks. Task 1 requires each speaker to record 25 given sentences from the TIMIT[28] database in 4 different vocal intensity categories(i.e. soft, normal, loud, very loud). During the recording process, the speakers deliver the sentences in their natural speaking style with a pause between words. Task 2 requires each speaker to record 2 given paragraphs of the novel "The Call of the Wild" by Jack London in 4 different vocal intensity categories as Task 1.

The texts used in Task 1 are listed in table 8 which can be found in the appendix. Each speaker was asked to recite the 25 given sentences in the four intensity categories. Overall there are $25 \times 4 \times 50 = 5000$ audio data. For each category, there are 1250 audio recordings respectively. In the experiments, the sentences will be repeated once by the speakers, therefore the total data in Task 1 is $5000 * 2$ repetitions$= 10000$. In this paper, we only adopted Task 1 as the experiment dataset because of dataset size of Task 2 is only 800 files(2 paragraphs 50 speakers 4 intensity categories 2 repetitions). In addition, to conduct a fair comparison of the experiment results in paper[15] and paper[16], which all used Task 1 as the experimental dataset but there is no Task 2 data used in paper[16].

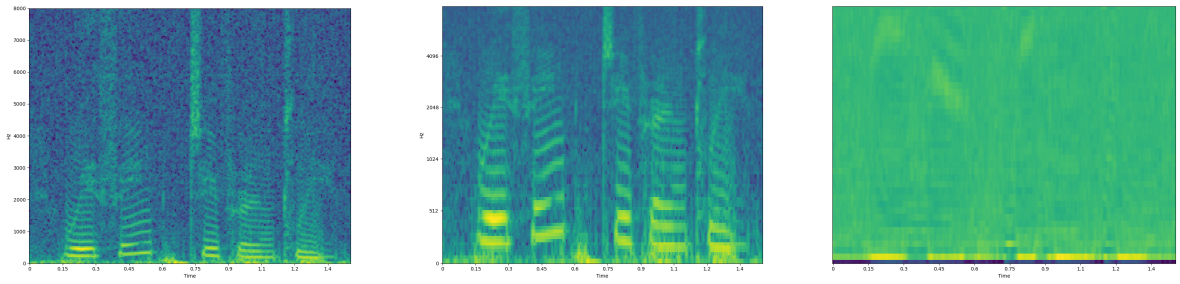
4.1 Data Exploration

Initially, we followed the dataset processing method from baseline method paper [15], where cut all the silence in each slice of recording in the dataset. But considering that CNNs are known to be very effective in learning features from spectrograms for various classification tasks, we proposed the potential to use the original data from vocal intensity classification.

Figure 12 demonstrates the original and cut spectrograms, mel-spectrograms and MFCC. Compared to the original spectrograms, before the raw data was translated into the cut spectrograms, the silence was removed from the raw data. The silence in the raw data includes the before/after of the speech and the silence between the words. From the images, we noticed that the cut spectrogram still shows a similar spectral



(a) Original Spec (b) Original Mel (c) Original MFCC



(d) Cut Spec (e) Cut Mel (f) Cut MFCC

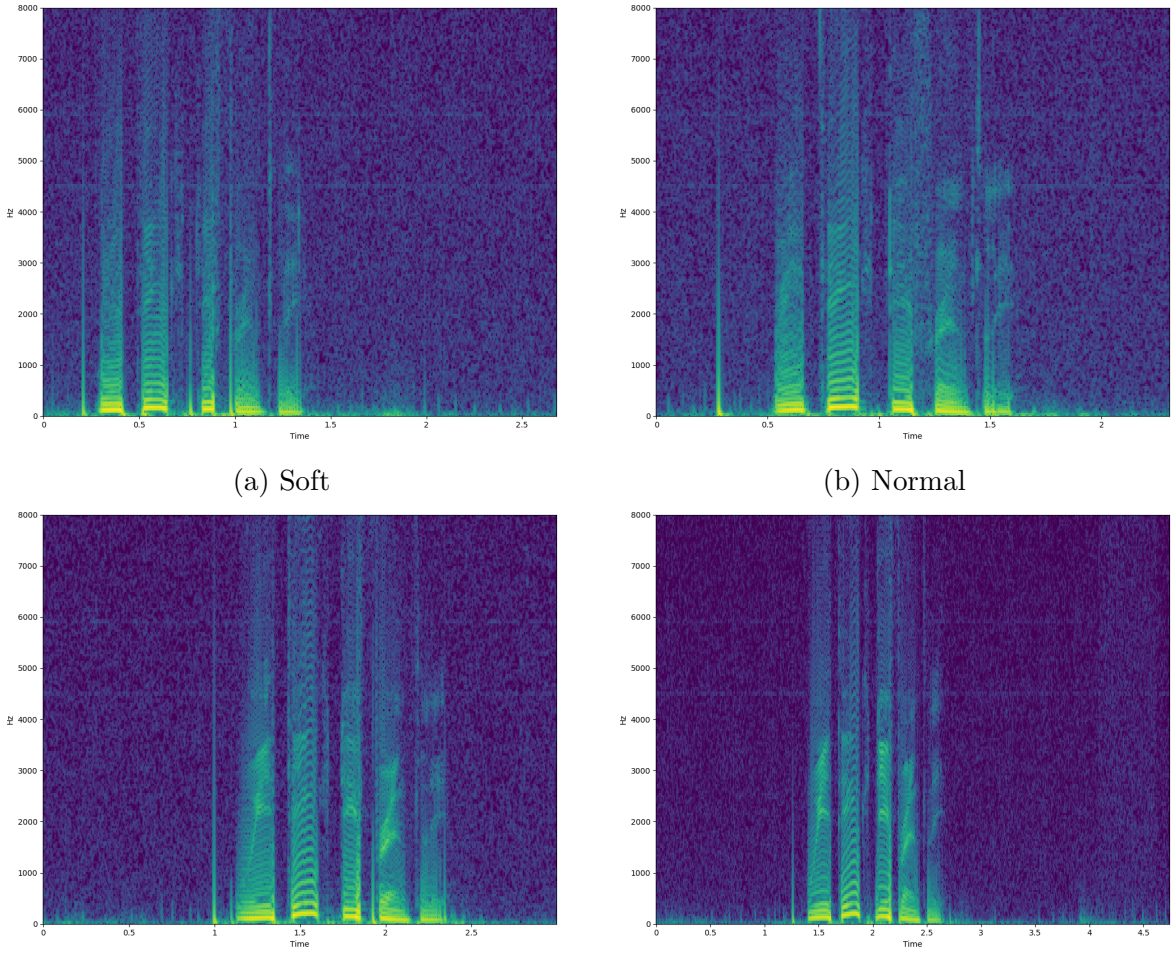
Figure 12: Comparison of the different spectrogram for Original and Cut Features

distribution and features. When the size of the spectrogram stays the same, the bandwidth in the cut spectrogram is also increased, and so is the potential noise. The cut spectrograms do not bring a thorough change to the features and, thus will not greatly influence the feature learning process for classification. Accordingly, it is unnecessary to cut the silence before feature extraction.

Figure 13 shows the spectrograms of 4 different categories, respectively. The spectrograms for the same uttered text but in different categories show a marked difference. There is no clear visible feature or pattern in the spectrogram that indicates the corresponding vocal intensity. It is design methods that are able to determine the vocal intensity spectrogram.

4.2 SPL label

Sound pressure level (SPL) is the measurement of vocal intensity, expressed on a decibel(dB) scale, as the logarithm of the ratio between the sound pressure and a standard reference pressure of $20\mu Pa$ [29]. The original label of the dataset, i.e., soft, normal, loud, very loud, are the subjective labels of each recording that depend on the speaker's understanding of the vocal intensity. The actual vocal intensity might differ from the label, so the SPL expressed is introduced to represent the objective measurement in-



(a) Soft

(b) Normal

(c) Loud

(d) Very Loud

Figure 13: The spectrograms for 4 vocal intensity classes from the same speaker that speaking the same sentence

tensity level label.

In the AVID dataset [27], the author calculated 18 different SPL labels for each audio data covering the most prevalent SPL parameters in frequency weightings (A-, C-, and Z(zero) weightings), time weightings(slow(S), medium(M), fast(F) with the time constant τ of 1 s, 0.125 s and 0.03 s), and time averaging(the mean SPL and the equivalent SPL) according to paper[30] and [27]. The following equation shows how to calculate the SPL value.

$$SPL_{\text{speech}} = 94 + 10\log_{10}\frac{\text{Energy(speech)}}{\text{Energy(calibration)}}. \tag{13}$$

The SPL label used in the baseline methods is the calculation result L_{meanZF}, i.e. with mean average, zero frequency weighting and fast time weighting. All the SPL data are stored in a metadata sheet in the dataset. However, when we try to relabel the SPL label as the baseline method, we find out that there is one-row lacking data of the audio file *sp*35_*s*1_*sen*4_*loud*. Consequently, we can not follow exactly the same procedure for determining the SPL label as used in the original SPL experiments of the baseline methods. In this paper, we use the provided audio and calibration data to calculate the Energy due to the limited data. The Energy is calculated with the equation 14, where the $x(n)$ represents the amplitude of the audio wave at the nth sample, and N is the total sample.

$$Energy = \sum_{n=0}^{N}(x(n))^2 \tag{14}$$

The SPL labels can be categorized as the SPL values in the table1 based on the calculation with the equation 13 and 14.

Intensity Category	**SPL Classification Range**	**SPL Categories File Numbers**
Soft	$SPL < 79\,\text{dB}$	1930
Normal	$79\,\text{dB} \leq SPL < 86\,\text{dB}$	2506
Loud	$86\,\text{dB} \leq SPL < 93\,\text{dB}$	3406
Very loud	$SPL \geq 93\,\text{dB}$	2158
Total		10000

Table 1: SPL Intensity Categories

5 Baseline Methods

In this section, we introduce the two baseline methods. The first method [15] is based on acoustic feature engineering plus the SVM as the vocal intensity classifier. The second method [16] is based on Transformers models for feature engineering and the SVM for vocal intensity classification.

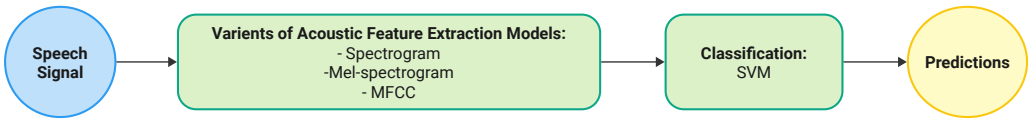


Figure 14: Baseline 1, using acoustic characteristics as features either spectrograms, mel-spectrograms or MFCCs. SVM with radial basis kernels is used as the classifier.

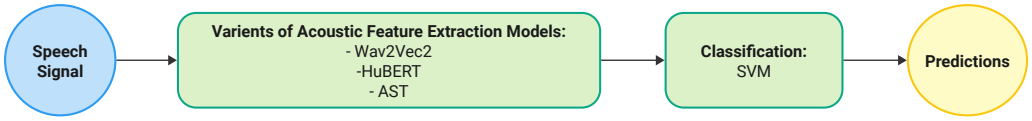


Figure 15: Baseline 2, using Transformer based models i.e., Wav2vec2, HuBERT and AST for feature extraction. SVM with radial basis kernels is used as the classifier.

The first baseline is introduced by [15] and the second baseline is studied paper[16] and shown to be the current SOTA on Vocal Intensity Category classification. Figure 14 and 15 depict Baseline 1 [15] and Baseline 2 [16] respectively. In the baseline model, the classification was done by SVM only. The main difference between these baselines is the acoustic feature extraction methods. Baseline 1 used spectrograms as acoustic features. 3 Types of spectrograms have been used, i.e., spectrograms, mel-spectrograms, and MFCCs. Baseline 2 used two speech models and one model used for audio classification for feature extraction, respectively. Wav2vec, HuBERT and AST are each time followed by an SVM classifier.

6 Spectrograms-ResNet Classification

The global architecture of our method is depicted in Figure 16. Our proposal was inspired by Baseline 1 [15], and during the research Baseline 2 [16] appeared, establishing the current state of the art on Vocal Intensity Category classification. We adopted the Acoustic Feature Extraction Model as Baseline 1, using Spectrograms, Mel-spectrograms, and MFCCs as the feature extractors. Meanwhile, in Baseline 1, the classifier is replaced by a further feature enhancement and subsequent FCNN classifier

in the form of ResNet 34, and ResNet 101, respectively. Note that, these could be replaced by other ImageNet-based DNN classifiers.

CNNs and other image classifiers trained on the ImageNet dataset [31] have been widely effective in the field of audio classification. Therefore, we studied and proposed to use ResNet as an analyzer and a feature enhancement method in this application of Vocal Intensity Category Classification.

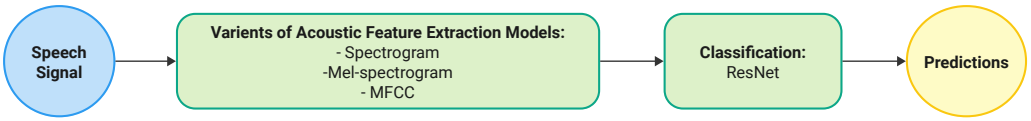


Figure 16: Spectrograms-ResNet Classification, using acoustic characteristics as features either spectrogram, mel-spectrograms or MFCCs. ResNet is used as the classifier.

Figure 17 is the outline of both base methods and improved methods that are involved in this paper. The left side depicts Baseline 1 with the traditional feature extraction process also used in our method, and the right side depicts Baseline 2.

6.1 Optimized STFT window-size

The frequency resolution is determined by window length and FFT size, the longer window length brings better frequency resolution but deficient the time resolution. The time resolution is determined by the hop size between successive frames, and the smaller hop size provides a better time resolution. Thus, when the hop size is relatively small, the time resolution will be greatly improved.

When the spectrogram size is fixed, the smaller hop size contributes a higher discrete time period density in time resolution. Therefore, the information density is easily increased in the spectrogram without increasing the spectrogram size itself. In this paper, the hop length/window length ratio is 0.2, where the hop length is a reduced number that is only 1/4 of the original setting. The spectrogram feature therefore is 4 times than the baseline method. Although the computational cost has increased during spectrogram transformation, the image size of spectrogram maintains the consistency, so no computational cost will be added on in the later machine learning process. With such data quality improvement, the training result will be improved correspondingly.

With our implementation, we conducted experiments to determine the optimized window length. See in section 3.1 for detailed theories.

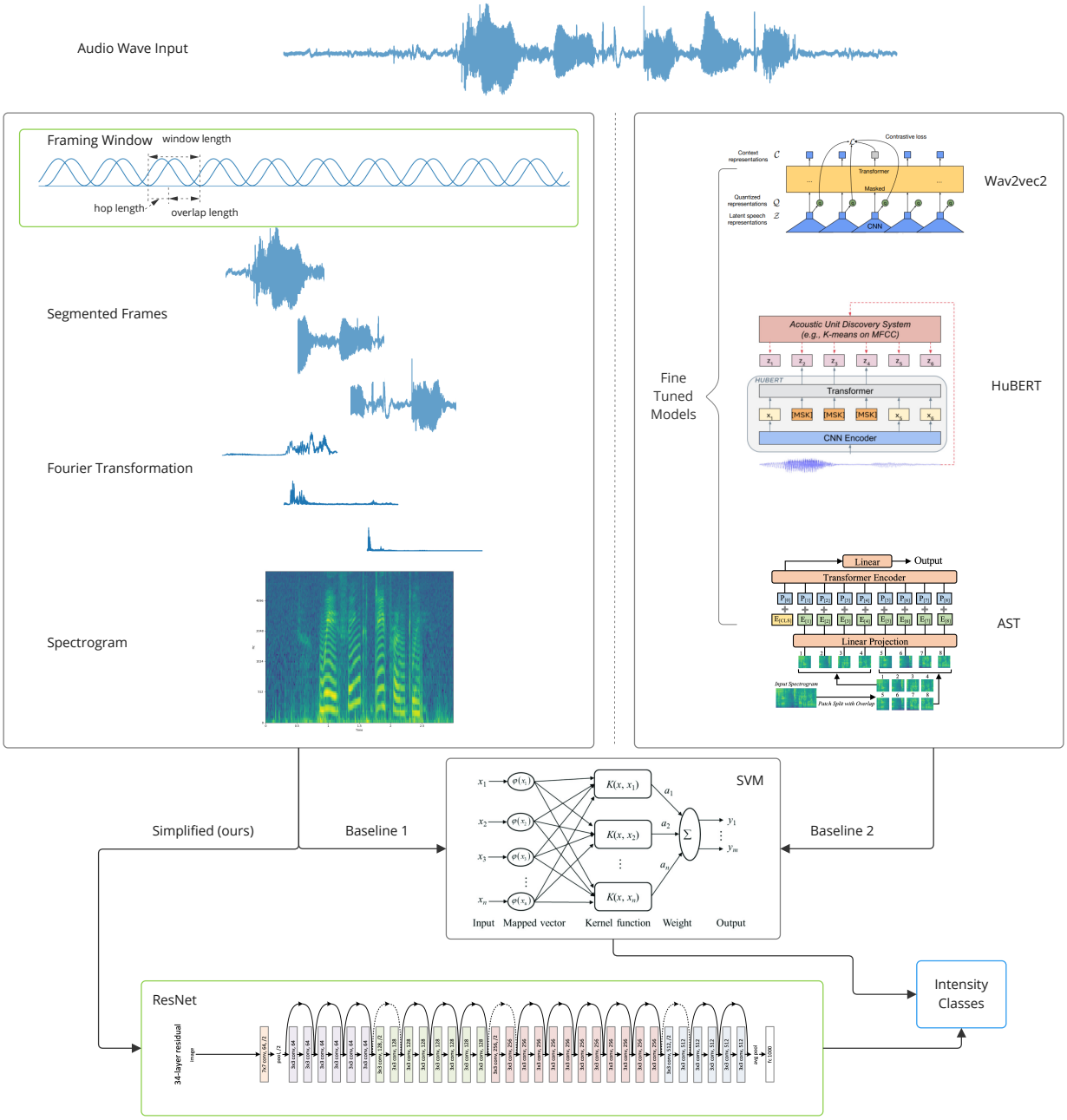


Figure 17: A detailed summary of the implementation of the evaluation of the methods used in this paper. The green rectangles are the major improvement compared to the baseline methods.

7 Experiment Setup

For the experiments, we used the original dataset as described in section 4 with the subjective labelling, i.e., recordings of the categories based on the speakers' target intensity category. In our proposed method we used Pytorch implementations of both ResNet34 and ResNet101. This will be indicated for the different experiments. The training and evaluation of the model was conducted on a PC with an NVIDIA RTX4090 with 24GB DRAM running on Windows 11 and Anaconda environment.

7.1 Feature Extraction

The original recordings are processed using the feature extraction methods mentioned in Section 3.1. The key parameter settings are listed below:

1. Spectrogram: the signals are sampled by the 25 ms Hamming window with a 5 ms hop length. The audio signals are processed to a 1024-point FFT represented as a 513-D vector.

2. Mel-spectrogram: using 1024-point FFT and 128 mel-filters giving a 128-D vector representing the mel-spectrogram.

3. MFCC: using a 39-D vector that included the delta and delta-delta coefficients, following the procedure as described in Section 3.1.4.

Besides evaluating these 3 feature extraction methods, we also conducted experiments to evaluate the influence of the labelling of the feature representations and classification results. All the audio data in the experiment was processed using the `librosa` library from Python.

7.2 GroupKFold Cross Validation

The training dataset used in our experiment is relatively small, whereas the ResNet models have a large parameter. Therefore in our experiments, we adopted group 5-fold cross-validation. The data set is separated into 5 portions of data with personal ID. As shown in Figure 18. For cross-validation training, 1 portion is chosen as the test dataset and the rest of the dataset is used as training data. This will be replicated 5 times until all the portions are considered as the test dataset once. When organizing the result of the experiment, we will gather the overall 5 training-testing results.

7.3 Training and Hyper Tuning

The training includes 2 dataset tasks, the original label dataset task and the SPL label dataset task. In the experiment, to optimize the model performance, there are a group of hyperparameters to be tuned in Table 2 on ResNet34 upon the original label data.

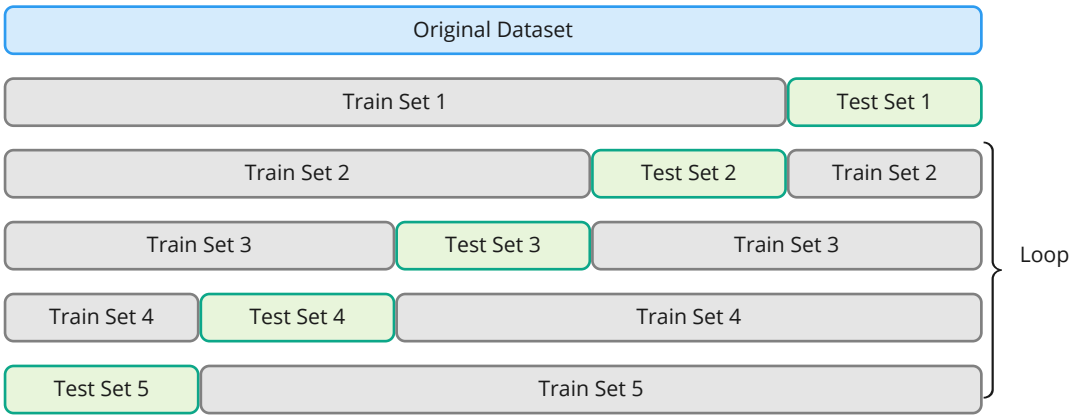


Figure 18: Cross Validation in the experimental dataset

Hyperparameters	**Notation in Code**	**Search Space**
Optimizer	opt	(Adam, SGD)
Learning rate	LR	(0.001,0.01, 0.02)
Momentum	M	0.9
Weight decay	WD	5e-4
Epochs	Epochs	[1,30]
Batch size	Batch size	32

Table 2: Hyperparameters and range of values that have been considered in our Hyper Parameter optimization Grid Search

In Table 2 the range of the hyperparameters considered for our experiment is listed. We employed a Grid Search to find the optimal value within the search space defined by these ranges.

8 Experiemental Results

For the hyperparameter optimization, we used ResNet34 model. Table 3 summarizes the performance of 3 types of spectrograms sets (mel, mfcc, spec) using the two considered optimizers (Adam, SGD) across different learning rates (0.001, 0.01, 0.02) in terms of accuracy, precision, recall, and F1-score.

Table 3: ResNet34 Results for Accuracy, Precision, and Recall using 30 epochs

Feature	**Optimizer**	**Learning Rate**	**Accuracy**	**Precision**	**Recall**	**F1-score**
mel	Adam	0.001	0.6936	0.6886	0.6936	0.6863
mel	Adam	0.01	0.6529	0.6827	0.6529	0.6548
mel	Adam	0.02	0.5687	0.6117	0.5687	0.5629
mel	SGD	0.001	**0.7117**	**0.7170**	**0.7117**	**0.7101**
mel	SGD	0.01	0.7061	0.7165	0.7061	0.7074
mel	SGD	0.02	0.7007	0.7119	0.7007	0.7001
mfcc	Adam	0.001	0.6773	0.6989	0.6773	0.6808
mfcc	Adam	0.01	0.5939	0.6253	0.5939	0.5870
mfcc	Adam	0.02	0.5845	0.6367	0.5845	0.5830
mfcc	SGD	0.001	0.6981	0.7017	0.6981	0.6980
mfcc	SGD	0.01	**0.7036**	0.7103	**0.7036**	**0.7038**
mfcc	SGD	0.02	0.7003	**0.7128**	0.7003	0.7017
spec	Adam	0.001	0.7007	0.7072	0.7007	0.6988
spec	Adam	0.01	0.6417	0.6548	0.6417	0.6356
spec	Adam	0.02	0.6054	0.6296	0.6054	0.6001
spec	SGD	0.001	0.7046	0.7134	0.7046	0.7045
spec	SGD	0.01	**0.7145**	**0.7317**	**0.7145**	**0.7171**
spec	SGD	0.02	0.6981	0.7200	0.6981	0.7023

Among the features, spectrogram consistently achieves the highest performance, with the best accuracy (0.7145) and F1-score (0.7171) observed when using SGD with a learning rate of 0.01. For mel-spectrogram, the best performance is achieved with SGD and a learning rate of 0.001 (accuracy: 0.7117, F1-score: 0.7101), while for MFCC, SGD with a learning rate of 0.01 yields the highest accuracy (0.7036) and F1-score (0.7038).

SGD generally outperforms Adam in terms of stability and performance, particularly with the lower learning rates (0.001 and 0.01). In contrast, Adam performs poorly at higher learning rates, with a noticeable degradation at 0.02. These results suggest that spectrogram or mel-spectrogram paired with SGD at a learning rate of 0.001 or 0.01 might be the optimal configuration for achieving robust and consistent performance across metrics. The optimized number of epochs is studied in Table 4.

Image Type	Optimizer	Learning Rate	Epoch	Accuracy
spec	SGD	0.01	28	0.7145
spec	SGD	0.01	26	0.7126
mel	SGD	0.01	24	0.7117
mel	SGD	0.001	23	0.7102
mel	SGD	0.001	28	0.7090
spec	SGD	0.01	15	0.7090
spec	SGD	0.001	21	0.7088
spec	SGD	0.001	25	0.7064
mel	SGD	0.01	18	0.7061
spec	SGD	0.001	27	0.7046

Table 4: Top 10 Performance metrics for different configurations of image type, optimizer, learning rate, epoch and accuracy

In Table 4 the respective number of epochs with highest accuracy are listed. The data corroborate the hypothesis that the model works great on the spectrogram or mel-spectrogram with a learning rate of 0.01 or 0.001. Figure 19 also demonstrates the accuracy over epochs upon the 4 potential optimal parameters. The 4 lines all show a similar trend in the training process, where the mel-spectrogram with a learning rate of 0.01 is relatively low performance.

Table 5 calculated the average last 10 accuracies of the 4 potential parameter settings and the standard deviation of the last 10 accuracies to check the stability and robustness of the parameter. The learning rate of 0.001 outperforms the learning rate of 0.01 in standard deviation, but the mel-spectrogram and spectrogram yield highly similar outcomes.

Due to time constraints, we did not conduct any further hyperparameters optimization experiments and selected the parameter setting **mel-spectrogram, learning rate 0.001, SGD optimizer** as the optimal setting because of therefore the highest accuracy.

Parameter Set	Average Accuracy	Standard Deviation
mel_0.001	0.70562	0.004359
mel_0.01	0.69100	0.008050
spec_0.001	0.70008	0.003677
spec_0.01	0.70060	0.007564

Table 5: Average Accuracy and Standard Deviation of Accuracy for Each Parameter Set of last 10 epochs

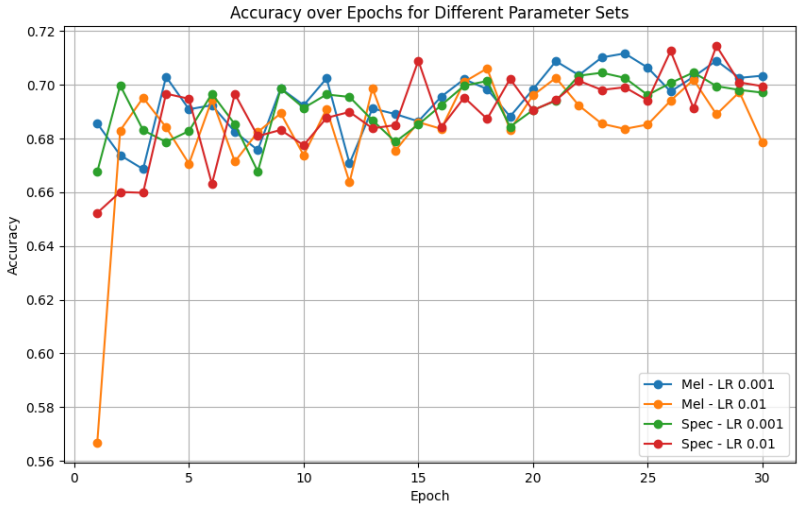


Figure 19: Accuracy of best parameter settings over epochs

Table 7 summarizes the fine-tuned baseline models as reported in [15] and [16]. Our proposed Spectrogram-ResNet method reaches a higher accuracy compared to the baseline methods. ResNet34 increases more than 5% accuracy than the Baseline 1 method and 2% than the Baseline 2 methods on the original subjective labelled vocal intensity dataset. ResNet101 has a higher accuracy than ResNet34, and the Wav2vec2-LARGE has a higher accuracy than Wav2vec2-BASE, indicating that the bigger networks are able to detect and extract additional features.

The results on the SPL labelled dataset surpass all the respective results on the target labels, which shows the effect of speakers' subjective understanding on the vocal intensity. In the following discussion in the confusion matrix and UMAP, we will discuss deeply the impact of original labels and SPL labels. Table 6 shows the highly unbalanced labelling in SPL labels. Both of the labels are reasonable for research. Original data shows the most realistic state of the data and provides hints for practical applications. SPL data shows a more realistic state of the data and effectively improves the accuracy of the data. Note that, the SPL labels used in our ResNet model experiments are different from the SPL labels in the baseline experiments [15] [16], due to the lack of metadata for the SPL label calculation process. The SPL labels methodology in the improved experiments are calculated as described in Section 4.2. It is evident that the ResNet still performs well on SPL labels.

Figure 20 depicts the classification results in the confusion matrix about different label methods on ResNet. As the complex matrix is shown in figure 20, the prediction results have a similar distribution trend with the same label method. Figure 20a and figure 20b are the objective label classification results, the accuracies are higher in cor-

Category	Original File Numbers	SPL File Numbers
Soft	2500	1930
Normal	2500	2306
Loud	2500	3406
Very Loud	2500	2158

Table 6: Comparison of original and SPL-labeled datasets across categories.

ner cases like soft or veryloud than normal or loud. The normal and loud categories are the intermediate transition zones and the vocal intensity in this zone is easily mixed up because of the objective induction of the speaker. After the SPL label correction, the SPL label results reinforced the soft category but averaged results among normal, loud and veryloud categories.

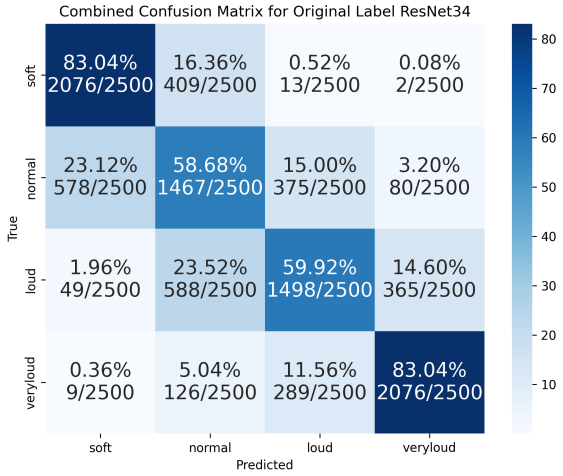
Figure 21 depicts the UMAP clusters of the features classification used in this paper. The presence of aggregated noise outside the clustering structure in all the images indicates that the quality of the dataset feature engineering leaves something to be desired. The figure of original label spectrogram 21a, original label mel-spectrogram 21b, and SPL label spectrogram 21c show a similar cluster structure, as we see a similar pattern in the previous dataset, where the spectrogram's and mel-spectrogram's feature are also relatively similar in naked eye observation. In figure 21a, the data show clustering in the center of each category but some of the data are also scattered to the side, confirming the problem of data mixing due to strong subjective judgments in the original data. The main difference between Figure 21d and 21b is the aggregation intensity among classes, where in SPL label mel-spectrogram the cluster layering is clearer the soft class is reinforced while veryloud class is diluted. Meanwhile, original label MFCC spectrogram 21c illustrated a very different cluster structure like crescent moon, and the clustering hierarchy is not obvious.

Combining the above analyses, we can conclude that SPL's distribution is more conducive to local intensity classification, because the clustering of each class in his distribution is stronger, and the similarity in features is higher, thus increasing the possibility of correct classification. The original dataset received the influence of the subjective judgement of vocalisation by the SPL, which resulted in the real vocal intensity deviating from the objective value, and the data hierarchy was not as distinct as that of the SPL. However, the significance of studying the raw data is to study the validity and robustness of the modelling approach, i.e., whether it can still be effective in real-life applications where there is a lot of noise.

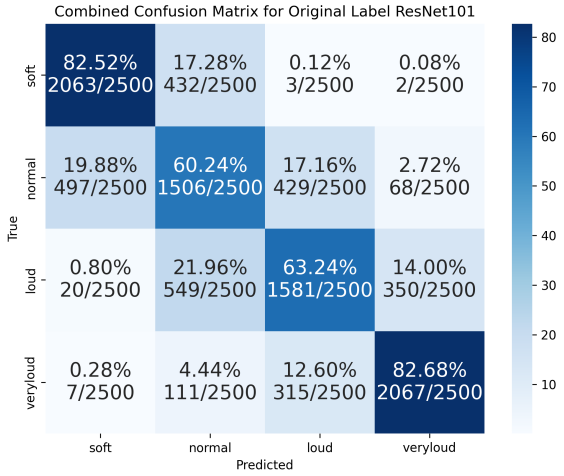
Table 7: Classification accuracy (in %) for both the baseline features and for the best-fine-tuned features of all the four models, evaluated for the target intensity category label and the SPL-based intensity category label.

Features	**Original, subjective intensity category label**	**SPL-based intensity category label**
Baseline1		
Spectrogram	66.08±2.77	81.00±2.36
Mel-spectrogram	65.41±2.11	68.65±4.00
MFCCs	63.19±2.63	66.62±4.8
Baseline2		
Wav2vec2-BASE	68.10±2.10	78.98±4.63
Wav2vec2-LARGE	69.90±2.79	79.71±4.76
HuBERT	69.7±3.40	**81.27±3.82**
AST	68.10±2.10	77.98±3.24
Improved		
ResNet34	71.17±5.72	* 78.20 ±2.40
ResNet101	**72.17 ±2.63**	* 78.13 ±3.75

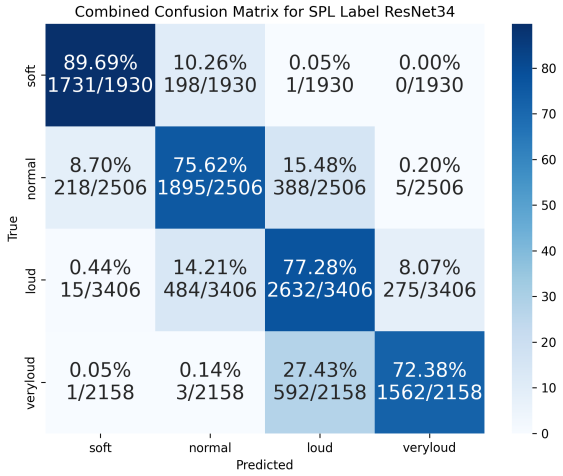
** the SPL label are calculated using our own scheme as introduced in Section 4.2, which differs from the original baseline papers*



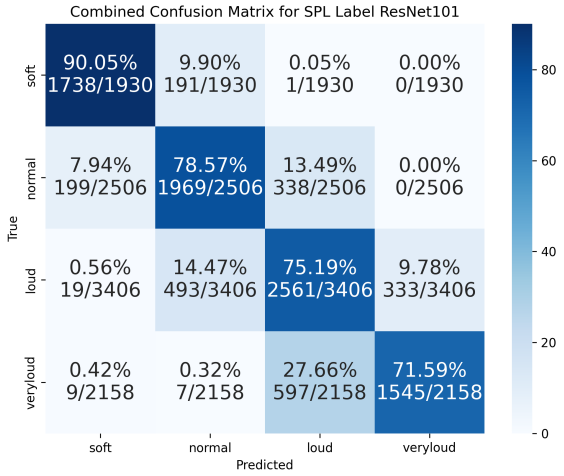
(a) Original Label ResNet34



(b) Original Label ResNet101

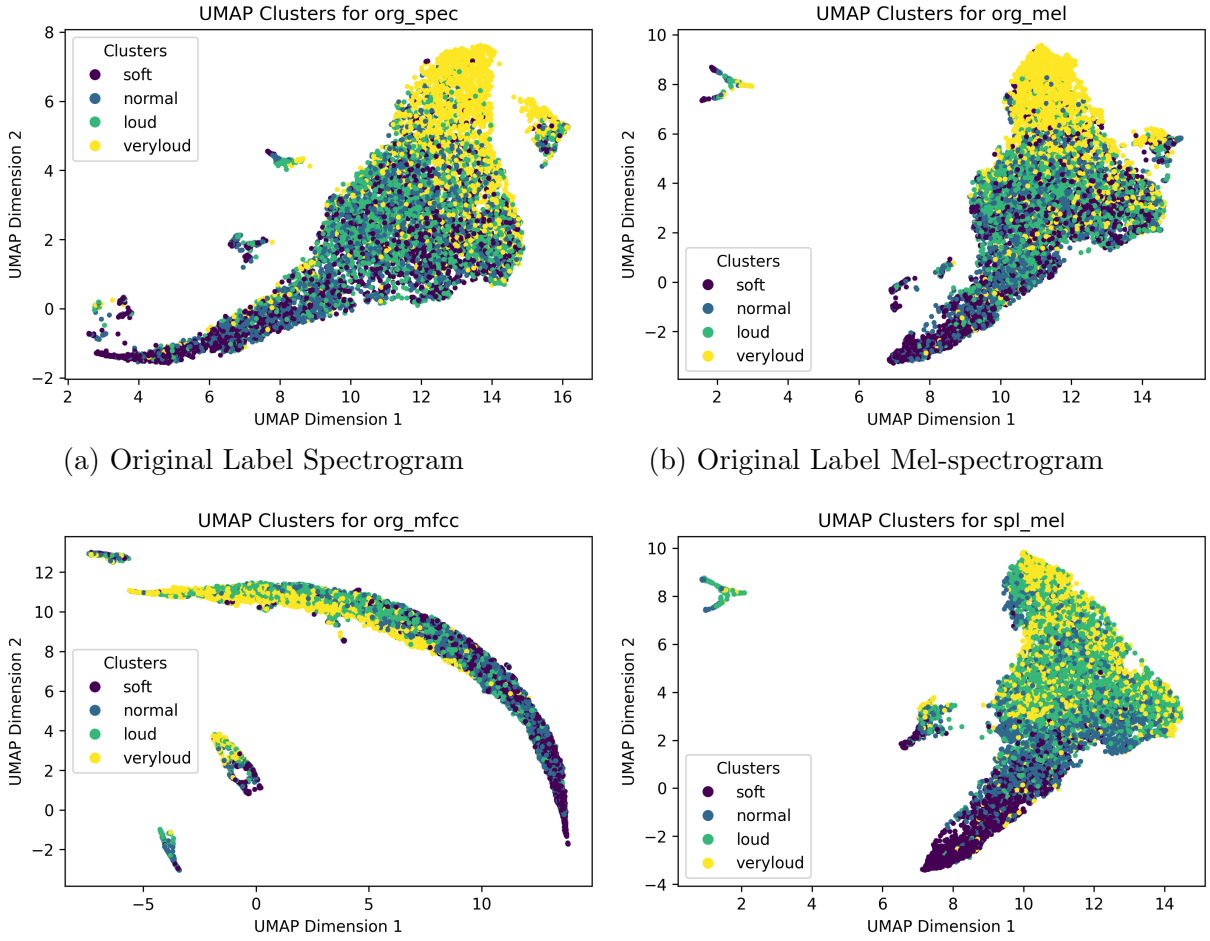


(c) SPL Label ResNet34



(d) SPL Label ResNet101

Figure 20: The spectrograms with 4 vocal intensity classification with ResNet



(a) Original Label Spectrogram

(b) Original Label Mel-spectrogram

(c) Original Label MFCC

(d) SPL Label Mel-spectrogram

Figure 21: The UMAP of spectrograms on the different acoustic features

9 Conclusion and Future

The paper investigates advancements in multi-class vocal intensity classification by exploring a group of models and analyzing vocal intensity data. It begins with an introduction to the concept of vocal intensity, its significance, and the rationale for its study. The research evaluates various models, including traditional machine learning approaches like SVM, deep learning architectures such as ResNet, and state-of-the-art frameworks like Wav2Vec2, HuBERT, and AST. Each model is described in terms of its functionality and application to vocal intensity classification.

Additionally, the paper delivers acoustic feature engineering focusing on spectrogram-based techniques. It outlines the processes for generating spectrograms, Mel-spectrograms, and MFCCs. And also discusses the methods for extracting features for classification. A detailed labelling system SPL for vocal intensity is also presented.

To address the challenge of improving classification methods, the paper proposes an enhanced approach leveraging all the spectrograms and ResNet. Experiments employing GroupKFold cross validation and hyperparameter optimization with GridSearch were conducted to achieve optimal performance. The results demonstrate that the proposed method improves classification accuracy by approximately 5% compared to the baseline on effectiveness.

In this paper, we delivered a simplified method for Vocal Intensity classification using classical acoustic feature engineering methods including spectrograms, mel-spectrograms and MFCC, respectively, combined with a more fine-grained FFT window-step-size, and subsequently adopting ResNet for image classification upon these spectrograms for vocal intensity. Our simplified method performs better against SOTA methods based on Transformers like Wav2Vec2, HuBERT, and AST with SVM as a classifier. As a result, our simplified method reaches an accuracy of 71% beating the SOTA, while simplifying the computing process and reducing computational complexity.

Although the final result is decent, there are many things that still can be improved.

1. **Bigger Dataset:** Contribute a bigger and more complete vocal intensity dataset and improve the data quality in the dataset that can satisfy the high-volume data requirements for SOTA and more complex models.

2. **Better Hyperparameter Optimization:** Incorporate advanced optimization strategies, such as adaptive optimizers and learning rate schedules, to ensure efficient training and prevent overfitting.

3. **Other Methods Instead of ResNet:** Explore innovative approaches like self-supervised models or alternative architectures tailored for vocal intensity analysis,

integrating advanced feature engineering techniques such as higher-dimensional features and time-frequency analysis.

4. **Multi-modal Models:** Leverage multimodality models like CLIP [32] to integrate audio, images (e.g., spectrograms), and text, enabling better feature representations and capturing complex patterns across modalities.

5. **Novel Visual Encoder-Decoder Models:** Experiment and adjust with advanced encoder-decoder frameworks like ViT [24] for enhanced feature learning in encoding and decoding.

6. **Data Augmentation:** Apply modality-specific augmentation techniques, such as adding noise and shifting pitch for audio or image transformations on spatial or pixel level, to enhance model robustness.

7. **Domain-specific Fine-tuning:** Fine-tuning on high-quality, task-relevant datasets like Parkinson's disease task [2] to tailor the model for specific problems, improving accuracy in the domain.

These future imrpovements aim to enhance the performance, robustness, and learning capabilities of the model, ensuring it is versatile and effective in analyzing vocal intensity across applications.

References

[1] Naomi A. Hartley and Susan L. Thibeault. Systemic hydration: Relating science to clinical practice in vocal health. *Journal of Voice*, 28(5):652.e1–652.e20, 2014.

[2] Manila Kodali, Sudarsana Reddy Kadiri, and Paavo Alku. Automatic classification of the severity level of parkinson's disease: A comparison of speaking tasks, features, and classifiers. *Computer Speech Language*, 83:101548, 2024.

[3] Xuhai Chen, Jianfeng Yang, Shuzhen Gan, and Yufang Yang. The contribution of sound intensity in vocal emotion perception: Behavioral and electrophysiological evidence. *PLOS ONE*, 7(1):1–11, 01 2012.

[4] Chi Zhang and John H. L. Hansen. Whisper-island detection based on unsupervised segmentation with entropy-based speech feature processing. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):883–894, 2011.

[5] Jouni Pohjalainen, Tuomo Raitio, Santeri Yrttiaho, and Paavo Alku. Detection of shouted speech in noise: Human and machine. *The Journal of the Acoustical Society of America*, 133(4):2377–2389, 04 2013.

[6] Nicolas Obin. Cries and Whispers - Classification of Vocal Effort in Expressive Speech. In *Interspeech*, pages 2234–2237, Portland, United States, September 2012.

[7] Chi Zhang and John H. L. Hansen. Analysis and classification of speech mode: whispered through shouted. In *Interspeech*, pages 2289–2292, 2007.

[8] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[9] Tamás Grósz, Dejan Porjazovski, Yaroslav Getman, Sudarsana Kadiri, and Mikko Kurimo. Wav2vec2-based paralinguistic systems to recognise vocalised emotions and stuttering. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 7026–7029, New York, NY, USA, 2022. Association for Computing Machinery.

[10] Shahin Amiriparian, Maurice Gerczuk, Sandra Ottl, Nicholas Cummins, Michael Freitag, Sergey Pugachevskiy, Alice Baird, and Björn Schuller. Snore sound classification using image-based deep spectrum features. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2017-August:3512–3516, 2017. Publisher Copyright: Copyright © 2017 ISCA.; 18th Annual Conference of the International Speech Communication Association, INTERSPEECH 2017 ; Conference date: 20-08-2017 Through 24-08-2017.

[11] Jeno Szep and Salim Hariri. Paralinguistic classification of mask wearing by image classifiers and fusion. In *Interspeech*, 2020.

[12] Albert Haque, Michelle Guo, Prateek Verma, and Li Fei-Fei. Audio-linguistic embeddings for spoken sentences. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7355–7359, 2019.

[13] Leonardo Pepino, Pablo Riera, and Luciana Ferrer. Emotion recognition from speech using wav2vec 2.0 embeddings. In *Interspeech*, pages 3400–3404, 08 2021.

[14] Saska Tirronen, Sudarsana Reddy Kadiri, and Paavo Alku. Hierarchical multi-class classification of voice disorders using self-supervised models and glottal features. *IEEE Open Journal of Signal Processing*, 4:80–88, 2023.

[15] Kodali Manila, Sudarsana Kadiri, and Paavo Alku. Classification of vocal intensity category from speech using the wav2vec2 and whisper embeddings. In *Interspeech*, pages 4134–4138, 08 2023.

[16] Kodali Manila, Sudarsana Kadiri, and Paavo Alku. Fine-tuning of pre-trained models for classification of vocal intensity category from speech signals. In *Interspeech*, pages 482–486, 09 2024.

[17] R. B. Blackman and J. W. Tukey. The measurement of power spectra from the point of view of communications engineering — part i. *The Bell System Technical Journal*, 37(1):185–282, 1958.

[18] S. Molau, M. Pitz, R. Schluter, and H. Ney. Computing mel-frequency cepstral coefficients on the power spectrum. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, volume 1, pages 73–76 vol.1, 2001.

[19] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.

[20] Wikipedia. Support vector machine — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Support%20vector%20machine&oldid=1271093106, 2025. [Online; accessed 27-January-2025].

[21] Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. Unsupervised speech recognition. *CoRR*, abs/2105.11084, 2021.

[22] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

[23] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. In *Interspeech*, pages 571–575, 08 2021.

[24] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[27] Paavo Alku, Manila Kodali, Laura Laaksonen, and Sudarsana Reddy Kadiri. Avid: A speech database for machine learning studies on vocal intensity. *Speech Communication*, 157:103039, 2024.

[28] John Garofolo, Lori Lamel, William Fisher, Jonathan Fiscus, David Pallett, Nancy Dahlgren, and Victor Zue. TIMIT Acoustic-Phonetic Continuous Speech Corpus, 1993.

[29] Ingo R. Titze and Daniel W. Martin. Principles of voice production. *The Journal of the Acoustical Society of America*, 104(3):1148–1148, 09 1998.

[30] Jan G Švec and Svante Granqvist. Tutorial and guidelines on measurement of sound pressure level in voice and speech. *Journal of Speech, Language, and Hearing Research*, 61(3):441–461, 2018.

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

A The Task1 in AVID dataset recording text

The following table includes all the text for recording in Task 1 in AVID.

Sentences list extracted from TIMIT for Task1
We think differently.
He spoke soothingly.
They despised foreigners.
That is your headache.
Nevertheless, it's true.
Leave me your address.
Come home right away.
Turn shaker upside down.
He makes me uncomfortable.
Did you eat yet?
Did anyone see my cab?
Push back up and repeat.
Hope to see you again.
This was easy for us.
Are you looking for employment?
Guess the question from the answer.
Orange juice tastes funny after toothpaste.
They all like long hot showers.
How do they turn out later?
Who is going to stop me?
All nut kernels are rich in protein.
Don't plan meals that are too complicated.
They often go out in the evening.
It was time to go up myself.
Birthday parties have cupcakes and ice cream.

Table 8: Sentences used for Task-1 data collection.