WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

SPEECH EMOTION RECOGNITION

Poorvi. B[1], Narram. S[2]

Coimbatore,Tamil Nadu

A B S T R A C T :

Speech Emotion Recognition (SER) is a growing field in AI that allows computers to identify and analyze emotions in human voice. This project aims to develop a fully reliable SER system, which can classify speech into various emotional categories such as calm, happy, fearful, and disgust. The system analyzes important acoustic features like MFCCs, chroma, and mel spectrograms, and then utilizes these extracted features to train an MLP classifier for efficient prediction of emotions from audio data. The basic motive for the existence of this system is that emotional intelligence in human-computer interaction is very much in demand nowadays, enriching far-ranging improvement of user experiences in domains such as healthcare, customer services, virtual assistance, and class training.The methodology takes a more structured approach starting with acquiring speech datasets in WAV format, followed by an extensive feature extraction process to sufficiently capture the emotions embedded in these audio signals. Afterward, the features get put into the MLP classifier, which, through supervised learning, develops an association between speech patterns and particular emotions.

Keywords: Speech Emotion Recognition, SER, MLPClassifier, audio feature extraction, MFCC, librosa, Streamlit, machine learning, affective computing, RAVDESS dataset, Python, emotion classification, user interaction, AI deployment, scikit-learn.

1. Main text

Within human-computer interactions, providing emotional intelligence completes the experience of the user. Speech Emotion Recognition (SER) empowers machines with the ability to recognize human emotions from speech, fostering more natural and humane interactione. The incorporation of SER systems within virtual assistants, healthcare diagnostics, and customer service platforms has opened up a black canal for AI applications concerning empathy. This paper stands for the design and implementation of a Speech Emotion Recognition system based on Multi-Layer Perceptron (MLP) classification and trained on an extracted audio feature set. Significant stress is placed on ensuring accurate emotion classification with extensive audio feature analysis and machine learning methods.

The primary goal for this project is to develop high-performance, real-time emotion recognition models that can classify emotions from speech into a set of predefined categories, especially calm, happy, fearful, and disgust. Feature extraction methods such as Mel Frequency Cepstral Coefficients (MFCC), chroma, and mel spectrograms have been used to characterize the acoustic properties of speech. The features are being computed, and are used as input to train an MLP classifier that has been successful in predicting emotional states from audio inputs.

Nomenclature

Term	Description
MFCC	Mel Frequency Cepstral Coefficients
MLP	Multi-Layer Perceptron
SER	Speech Emotion Recognition
Chroma	A feature representing the 12 different pitch classes
Mel Spectrogram	Representation of audio based on mel scale frequencies
Emotion Classes	Calm, Happy, Fearful, Disgust (used in this study)
Streamlit	A framework used to deploy the model on a web interface

1.1. Speech Emotion Recognition

Speech Emotion Recognition (SER) is a new area of artificial intelligence that deals with the recognition of human emotions from speech signals. Emotions are central to human communication, driving decision-making, perception, and interaction. Through the use of machine learning techniques and sophisticated audio signal processing methods, SER systems intend to bridge the gap in human-computer interaction by making machines capable

of understanding and acting upon emotional stimuli. This work introduces an end-to-end solution for emotion recognition from speech based on Mel Frequency Cepstral Coefficients (MFCC), Chroma, and Mel spectrogram features with a trained Multi-Layer Perceptron (MLP) classifier. Merging SER into practical application can transform customer support, mental health tracking, and voice assistants into more human-like, empathetic interactions between human and machine.

1.2. MLP Classifier

The Multi-Layer Perceptron (MLP) Classifier is a type of feedforward artificial neural network often used in supervised machine learning for tasks like classification and regression. As a key part of deep learning, the MLP classifier has a layered structure that includes an input layer, one or more hidden layers, and an output layer. Each layer contains multiple nodes, called neurons, which are connected to nodes in the next layer through weighted links. The MLP classifier is particularly effective because it can capture complex, non-linear relationships in data, making it useful for various applications such as image recognition, speech emotion recognition, natural language processing, and financial forecasting.

At the heart of the MLP classifier is forward propagation. Here, input data moves through the network. It is transformed using weighted sums and non-linear activation functions before passing to the next layer. Common activation functions include the Rectified Linear Unit (ReLU), sigmoid, and tanh functions. Each contributes to the network's ability to model non-linear patterns. The output layer provides classification results, often using the softmax function for multi-class tasks. The MLP classifier's learning process is driven by the backpropagation algorithm, which calculates the error between predicted outputs and actual labels. This error moves backward through the network to update the weights using optimization methods like Stochastic Gradient Descent (SGD) or the Adam optimizer. This cycle repeats until the model reaches a satisfactory level of accuracy.

One major strength of the MLP classifier is its ability to handle both linearly separable and non-linearly separable data. Unlike simpler models, such as logistic regression or linear discriminant analysis, the MLP can model complex data distributions thanks to its multi-layered design. The number of hidden layers and neurons in each layer, known as the network's architecture, is critical to the model's capacity and performance. A well-structured MLP with suitable hyperparameters can generalize effectively from training data to new, unseen data, reducing both bias and variance. However, if the network is too complex, it may overfit, memorizing training data instead of learning general patterns. Techniques like dropout, early stopping, and regularization help reduce overfitting and improve model robustness.

In real-world applications, the MLP classifier has shown great success across different domains. In Speech Emotion Recognition (SER), the MLP classifier is skilled at learning from audio features like Mel Frequency Cepstral Coefficients (MFCC), chroma features, and Mel spectrograms. By examining these features, the MLP can classify emotions such as happiness, sadness, anger, fear, and surprise from speech data. This ability helps create emotionally intelligent systems that improve human-computer interaction, support mental health evaluations, and enhance user experiences in customer service.Despite its advantages, the MLP classifier has some limitations, especially when dealing with large datasets or highly sequential data. In those cases, models like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) might perform better. Still, for structured and moderately sized datasets, the MLP is a reliable and effective option. As computational resources and optimization algorithms continue to progress, the MLP classifier remains a vital part of machine learning and artificial intelligence, striking a balance between simplicity and effectiveness in prediction.

1.3. Audio Feature Extraction

Audio feature extraction is a key process in speech processing, machine learning, and audio analysis. It plays an important role in tasks like speech emotion recognition, speaker identification, music classification, and sound event detection. This process involves turning raw audio signals, which can be complex and high-dimensional, into a compact set of features that highlight the essential traits of the sound. This transformation is necessary because raw audio data is rich in information but not suitable for machine learning models. Its continuous, noisy, and intricate nature makes it challenging. By extracting meaningful features, we can improve the model’s ability to understand and process auditory content, aiding in classification, recognition, and pattern detection.

One of the most commonly used audio features in speech and audio analysis is the Mel Frequency Cepstral Coefficients (MFCCs). MFCCs capture the short-term power spectrum of an audio signal and represent it on a mel scale. This scale reflects how listeners perceive pitches as evenly spaced. This method mimics the human ear's non-linear response to frequencies, ensuring that the model focuses more on frequencies that are important to humans. The MFCC extraction process typically includes several steps: pre-emphasis to boost high frequencies, framing the signal into short segments, using a window function to reduce spectral leakage, performing a Fast Fourier Transform (FFT) to change the signal to the frequency domain, and then applying mel filter banks followed by a Discrete Cosine Transform (DCT). The resulting coefficients effectively describe the timbral and phonetic characteristics of the sound.

Another important feature is the Chroma Feature, which captures the energy distribution across the twelve distinct pitch classes, or semitones of the musical octave. Chroma features are especially useful in music applications but also help in speech analysis. They assist in identifying tonal characteristics and pitch variations, which are often linked to emotional states. These features help us understand the harmonic and melodic content of the audio.

Spectral features like Spectral Centroid, Spectral Rolloff, Spectral Bandwidth, and Spectral Contrast offer insights into the shape of the audio spectrum. The spectral centroid shows where the center of mass of the spectrum is located and usually relates to how bright the sound seems. Spectral rolloff measures the frequency below which a certain percentage of the total spectral energy is found, indicating how sharp the spectrum is. These features are crucial for distinguishing between different types of sounds, speech patterns, and even emotions since changes in pitch and tone can reflect emotional shifts.

In addition, the Mel Spectrogram provides a powerful representation where the audio signal is transformed into a spectrogram with frequencies placed on the mel scale. This visual representation captures both time and frequency information and is often used as input for convolutional neural networks (CNNs) in deep learning tasks, improving performance in audio classification.

Zero Crossing Rate (ZCR) and Root Mean Square Energy (RMSE) are time-domain features often used in audio analysis. ZCR indicates how often the signal changes from positive to negative or vice versa, which helps in distinguishing between voiced and unvoiced speech. RMSE gives an estimate of the signal's energy and can indicate the intensity or loudness of speech, which may vary with emotional states.

Choosing features for a specific task is essential because different features capture different aspects of the sound. A well-thought-out combination of time-domain, frequency-domain, and perceptual features ensures that the model receives a complete representation of the audio signal. This improves its predictive accuracy and generalization

1.4. MFCC

Okay, here's a more human way to explain MFCCs:

MFCCs are like a really popular way to pull out important features from speech and audio. They're a go-to tool for things like speech recognition, figuring out who's talking, and figuring out how someone is feeling just by listening to them.

The cool thing about MFCCs is that they're designed to work a bit like how our ears work. Humans don't hear all frequencies the same way. We're way more sensitive to lower sounds than higher ones. MFCCs take that into account. They change the sound signal to match our perception.

So, how do they do it? It's basically a bunch of steps:

1. Pre-emphasis: First, they boost the high-frequency parts of the audio. This balances things out since low frequencies usually dominate speech.
2. Framing: Next, the audio is broken down into tiny chunks, like little slices of time.
3. Windowing: Each slice is then smoothed out to avoid weird breaks when analyzing the frequencies.
4. FFT: Then, they use something called FFT to see what frequencies are present in each slice.
5. Mel Filter Bank: After that, they run the frequencies through a bunch of filters that mimic how our ears hear different sounds.
6. Logarithm: They compress the sound to copy how sensitive our ears react to sound levels. The quieter sounds are amplified.
7. DCT: The final step is something called DCT, which cleans up the data and spits out the MFCCs.

In the end, you get these MFCC features that sum up the important parts of the sound. Usually, you only need the first 12 or 13 because that's where the gold is.

MFCCs are used everywhere because they're good at dealing with noise and because they mimic human hearing. They can pick up on subtle changes in tone and pitch, which is great for emotion recognition.

They're also pretty quick to compute. So, when you plug MFCCs into machine learning models such as support vector machines, CNN, or multi-layer perceptrons, you can get really good at identifying speech.

Basically, MFCCs are a key ingredient in making computers understand speech and emotions, helping tech interact with humans in a smarter manner.

1.5. Librosa

Librosa is a cool Python library a lot of people use for looking at audio and music. It's got a bunch of tools to help pull out info from audio, see it in different ways, and break it down. People mainly use it for things like figuring out what kind of music it is or working with speech.

It's free to use and works with other common Python tools like NumPy and matplotlib, so it runs smoothly and can show you nice pictures. What's really neat is that Librosa can turn audio into stuff that computers can easily understand for machine learning. Things like MFCCs and spectrograms? It can get those for you super quick. These things are useful for telling if someone's happy or sad based on their voice, sorting music into genres, figuring out who's talking, or spotting different sounds.

Librosa can also do fancy stuff with time, like guess the tempo, find the beat, spot when sounds start, and track the pitch. This lets you check out the rhythm and melody in music or speech. It can also load and save audio files, change the sample rate, filter out noise, and prep audio. It can basically handle the whole audio analysis process. Plus, it hooks up with tools that let you make cool graphs, like waveforms and spectrograms, so you can really see what's going on in the audio.

One of the best parts is that Librosa is easy to use, and it has good instructions, so anyone from newbies to pros can get the hang of it. The people who made it keep updating it, so it gets better all the time. Overall, Librosa is a must-have if you're playing with audio and machine learning. It gives you what you need to mess with audio data for all sorts of creative projects.

1.6. File naming and delivery

Please title your files in this order 'procediaacronym_conferenceacronym_authorslastname'. Submit both the source file and the PDF to the Guest Editor.

Artwork filenames should comply with the syntax "aabbbbbb.ccc", where:

- a = artwork component type
- b = manuscript reference code
- c = standard file extension
- Component types:
- gr = figure
- pl = plate
- sc = scheme
- fx = fixed graphic

1.7. Emotion Classification

Figuring out emotions is a special part of AI that tries to spot and sort human feelings using things like voice, text, face moves, or body signals. Using voice to figure out emotions is getting lots of looks because it is easy to get the information you need. Feelings are tricky and show up in how we talk like our pitch, tone, and speed. If machines can pick up on these feelings, they can get what we are feeling and make talking with them way better.

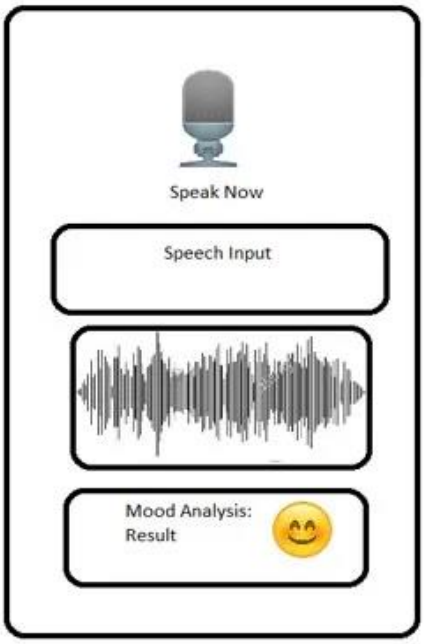
When it comes to voice, sorting emotions usually goes like this: cleaning up the sound, pulling out key details, teaching a model, and then guessing the emotion. Cleaning makes the audio sound better by cutting out noise and making everything the same level. Pulling out details is super important because it turns the audio into numbers that computers can understand. Stuff like MFCCs, Chroma features, and Mel spectrograms are used to grab things like frequency and energy, which helps tell apart feelings like joy, sadness, anger, fear, disgust, and surprise.

Computers learn to sort these emotions using the details we give them. Things like Support Vector Machines (SVM), Random Forests, and Multi-Layer Perceptrons (MLP) work pretty well. Deep learning, like using Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), makes things even better by learning the tricky stuff in the data all on its own.

Being able to tell emotions has tons of uses and can make a big deal. In health, it can watch over how people are doing mentally and catch signs of the blues or worries. For helping customers, it can make things better by changing answers based on how folks feel. Also, it is a big help in making assistants that get emotions, improving games, and making learning online better by reacting to how students feel.In short, using voice to guess emotions is a game-changer that mixes sound smarts, computer learning, and how people think. As things get better, it could make systems that get us, talk back better, and are cleverer.

2. Illustrations

- The system takes audio in the .wav format.
- Then, it cleans up the audio by getting rid of noise and making the signal consistent.
- Next, it changes the audio into numbers, like MFCC, Chroma, and Mel spectrograms. These numbers describe the sound.
- After that, these numbers go into a pre-trained MLP model to guess the emotion.
- Finally, the system shows the emotion it thinks is in the audio. It can also show an emoji or icon to make it easier to understand.



This Speech Emotion Recognition (SER) project uses machine learning and deep learning to figure out how people are feeling just from their voice. When we talk to each other, our emotions come out not just in what we say, but also in how we say it – things like tone, pitch, and how loud we are. The SER system listens to these voice clues to guess what emotion someone is feeling. This helps computers understand us better and act more like real people.

Here's how it works: First, the system grabs audio data. This means using recordings of people talking, where each recording is tagged with the emotion being expressed. Datasets such as RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) give a bunch of different emotions to work with, like happiness, sadness, anger, and fear.

After getting the data, the next thing is to prep the audio and pull out the important bits. The system makes sure all the sound files are the same, and it cleans up any background noise that could mess things up. Then, it grabs key features from the audio, like MFCCs (Mel Frequency Cepstral Coefficients), Chroma Features, and Mel Spectrograms, using tools like Librosa. MFCCs are good at showing how we hear sound, so they're helpful for guessing emotions. Chroma features look at the pitch of the voice, and Mel Spectrograms show the sound's frequency.

With these features in hand, the data goes into the machine learning part. The data is split into two groups: one to train the system and one to test it. This makes sure the system learns well and isn't just memorizing stuff. A Multi-Layer Perceptron (MLP) Classifier, which is a type of neural network, does the main job of guessing the emotion. It learns from the training data and gets better at figuring out which voice features match which emotions.

To see how well the system works, it's tested using accuracy, precision, recall, and F1-score. These scores show if the system can guess emotions correctly on new audio it hasn't seen before. To get the emotions right, the system uses a LabelEncoder to turn emotion labels into numbers during training and then back into words when it makes a guess. The trained system is saved using Pickle, so it can be used later without needing to be retrained.

To put it all together, the project uses Streamlit, a tool for making web apps. People can upload their voice recordings, and the system will guess their emotion in real time. It shows the guess and how sure it is, sometimes using emojis to make it fun. This makes it easy for anyone to use the system.

This kind of system could be used in lots of places. Customer service centers could use it to tell if someone is getting annoyed. Doctors could use it to check on a patient's mental health. Robots could use it to understand how people are feeling and respond in a nicer way.

Basically, this project brings together sound processing, machine learning, and user-friendly design to make a system that understands emotions from speech. It's a step toward making computers more aware of how we feel, leading to better interactions between people and machines.

Equations

1. Short-Time Fourier Transform (STFT):
STFT{x(t)}(m,ω)=∑n=−∞∞x[n]·w[n−m]·e−jωnSTFT\{x(t)\}(m, \omega) = \sum_{n=-\infty}^{\infty} x[n] \cdot w[n - m] \cdot e^{-j\omega n}

2. Mel Scale Conversion:
M(f)=2595log⁡10(1+f700)M(f) = 2595 \log_{10} \left(1 + \frac{f}{700}\right)

3. MFCC (Discrete Cosine Transform step):
cn=∑k=1Klog⁡(Sk)·cos⁡[n(k−12)πK]c_n = \sum_{k=1}^{K} \log(S_k) \cdot \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K} \right]

4. MLP Forward Propagation:
z(l)=W(l)a(l−1)+b(l)z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)}
a(l)=f(z(l))a^{(l)} = f(z^{(l)})

5. Softmax Function:
σ(z)j=ezj∑k=1Kezk\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}

6. Categorical Cross-Entropy Loss:
L=−∑i=1N∑c=1Cyiclog⁡(y^ic)L = - \sum_{i=1}^{N} \sum_{c=1}^{C} y_{ic} \log(\hat{

7. Accuracy:
Accuracy=Number of correct predictionsTotal number of predictions\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}

8. F1 Score:
F1=2×Precision×RecallPrecision+RecallF1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}

4. Online license transfer

This Speech Emotion Recognition (SER) project uses open-source stuff like Librosa, Scikit-learn, NumPy, and Soundfile – they're all under the BSD License. That means you can use them for free, change them, and share them, just give credit where it's due. Streamlit, which we use for the user interface, has an Apache 2.0 license, so it's free to use too. The code we wrote for this project is under the MIT License, so anyone can use it, tweak it, and share it as long as they give us credit. This open approach means everyone can work together to make emotion recognition better.

Acknowledgements

I'm super grateful for all the help I got during this Speech Emotion Recognition (SER) project. First off, huge thanks to my teachers and professors. They always had great ideas and feedback, and their smarts in machine learning, deep learning, and signal stuff really made my work better and helped me learn a lot.

Also, shout out to my school for the support and resources. Having the right software, computers, and tech stuff made it way easier to build, test, and improve things for this project.

Big thanks to the people who made Librosa, Scikit-learn, NumPy, Pandas, Soundfile, and Streamlit. These free tools were super important for my work. Building a good SER system in time would've been way harder without them.

I'm also thankful for my classmates and friends. They kept me going, helped me work through problems, and gave great suggestions. Working with them was both fun and made me feel excited about the work.

My family also deserves a ton of credit for their never-ending support and patience. They always believed in me, which pushed me to keep going even when things got tough.

I also want to thank the people who created the RAVDESS dataset. It was the main resource for training and testing my emotion recognition models. Having such a well-organized dataset is key for doing work in this field.

Finally, I want to acknowledge all the pros, researchers, and innovators whose previous work in speech emotion recognition, machine learning, and

human-computer stuff paved the way for my project. Their work has been a guide, and my project just builds on what they've already done.

I couldn't have finished this project without all the support, knowledge, and inspiration from everyone.

REFERENCES

1. 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.
2. Schuller, B., Batliner, A., Steidl, S., & Seppi, D. (2011). Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge. *Speech Communication*, 53(9-10), 1062-1087.
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.
4. Han, K., Yu, D., & Tashev, I. (2014). Speech emotion recognition using deep neural network and extreme learning machine. In *INTERSPEECH* (pp. 223-227).
5. Fayek, H. M., Lech, M., & Cavedon, L. (2017). Evaluating deep learning architectures for Speech Emotion Recognition. *Neural Networks*, 92, 60-68.
6. Wöllmer, M., Eyben, F., Schuller, B., & Rigoll, G. (2010). A multi-modal LSTM recurrent neural network for continuous emotion recognition. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 3482-3485).
7. Ringeval, F., Schuller, B., & Cowie, R. (2013). Interacting with sensitive artificial listeners: Embedding emotion recognition in a virtual agent. *Affective Computing and Intelligent Interaction*, 163-172.
8. Lee, C. M., & Narayanan, S. S. (2005). Toward detecting emotions in spoken dialogs. *IEEE Transactions on Speech and Audio Processing*, 13(2), 293-303.
9. Trigeorgis, G., et al. (2016). Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network. *ICASSP*, 2016.
10. Ververidis, D., & Kotropoulos, C. (2006). Emotional speech recognition: Resources, features, and methods. *Speech Communication*, 48(9), 1162-1181.
11. Satt, A., Rozenberg, S., & Hoory, R. (2017). Efficient emotion recognition from speech using deep learning on spectrograms. In *INTERSPEECH* (pp. 1089-1093).
12. Wu, S., Falk, T. H., & Chan, W. Y. (2011). Automatic speech emotion recognition using modulation spectral features. *Speech Communication*, 53(5), 768-785.
13. Schuller, B., Steidl, S., & Batliner, A. (2009). The INTERSPEECH 2009 Emotion Challenge. *INTERSPEECH 2009*, 312-315.
14. Eyben, F., Weninger, F., Groß, F., & Schuller, B. (2013). Recent developments in openSMILE, the Munich open-source multimedia feature extractor. In *Proceedings of the 21st ACM International Conference on Multimedia* (pp. 835-838).
15. He, L., & Chao, W. (2019). Speech emotion recognition based on optimal short-term features. *International Journal of Speech Technology*, 22, 355-363.