Audio Pattern Recognition - Emotion Classification

Alan Cai, Ali Saleh, Manraj Singh, Mekdes Teklehaimanot

RAVDESS Dataset

The RAVDESS dataset contains 1440 files: 60 trials per actor x 24 actors = 1440.

The RAVDESS contains 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent.

Speech emotions includes calm, happy, sad, angry, fearful, surprise, and disgust expressions. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression.

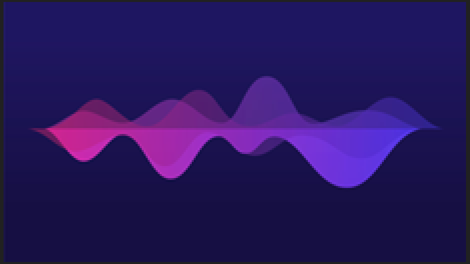
Samples:

Angry

Sad

Calm

Goal & Motivation



- **Goal:** To build a robust and accurate audio pattern recognition system that can identify different emotions from audio data. The approach will utilize various machine learning models including MLP, SVM, RNN, and CNN to explore which model works best for this task.
  - In addition to comparing the performance of different models, we will explain the underlying workings of the CNN model used for audio classification, and on the key aspects that drive its decision-making process.

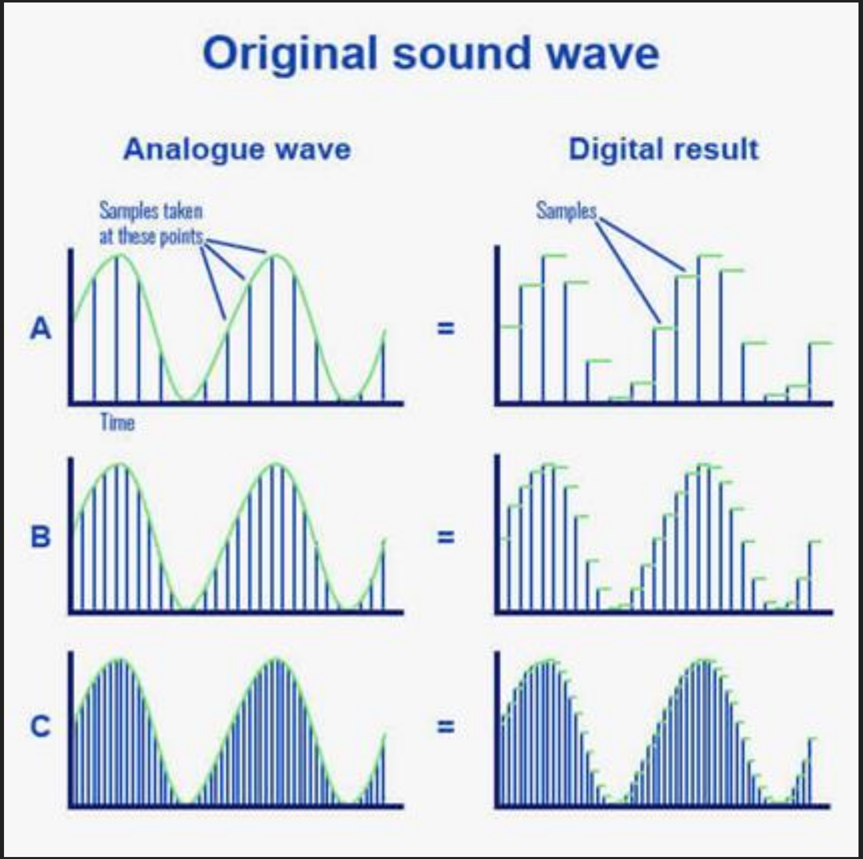
- **Motivation:** Our motivation is to address the increasing need for emotion recognition in various fields, such as mental health, customer service, entertainment, and virtual assistants. Developing a robust and accurate system will facilitate more natural and effective communication between humans and computers, leading to better decision making, improved user experiences, and more personalized interactions across multiple industries.

Raw Audio Data

Using the librosa python module, the RAVDESS audio samples were loaded using their original sample rate (44100 Hz).

Raw audio data consists of a continuous sequence of digital samples that represent the amplitude of an audio signal at specific points in time. These samples are taken at a fixed rate called the sampling rate, in this case 44,100 samples per second.

Higher sampling rates result in a more accurate representation of the original analog audio signal.



Raw Data Models

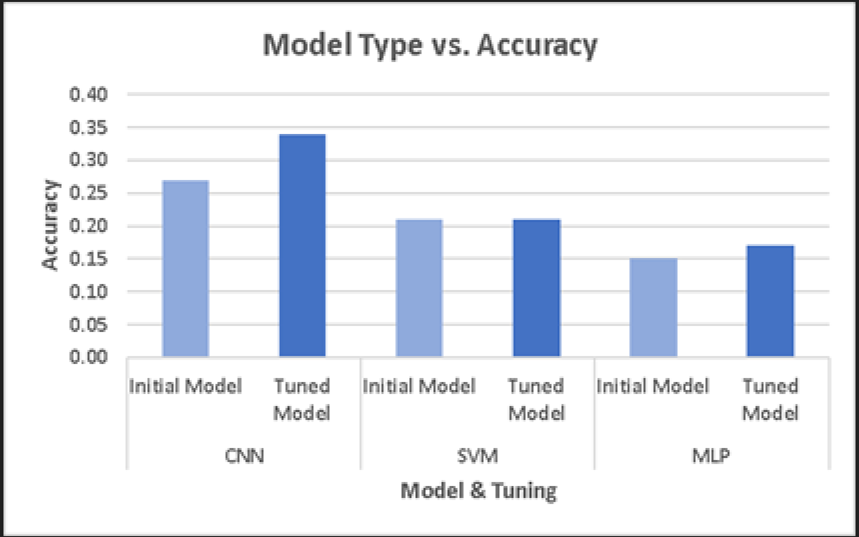
Each of these models were trained and evaluated on the raw audio data to determine their classification accuracies.

The results showed that the CNN model outperformed the other models with the highest accuracy and improvement.

Tuning the hyperparameter has improved the accuracies for all the models.

RNN was not feasible for the raw data due to the high dimensionality (253,053) and complex temporal structure of the data, making the input sequences too long and computationally expensive to train the model.

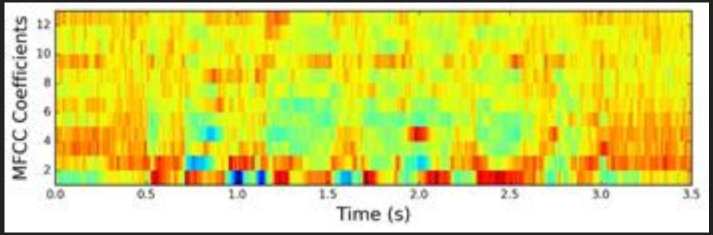
However, all models had relatively low accuracies indicating the raw audio data alone is not sufficient for accurate emotion recognition, and that steps such as *feature extraction and data augmentation* would be investigated.

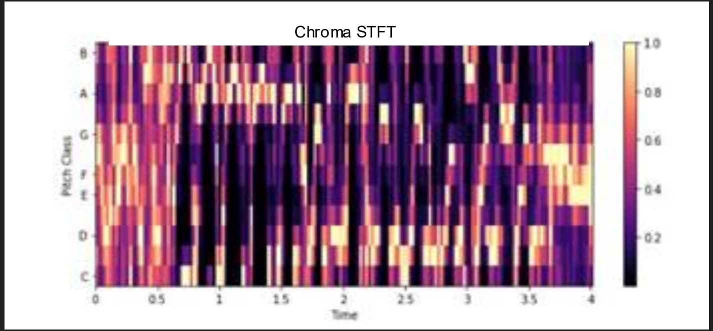


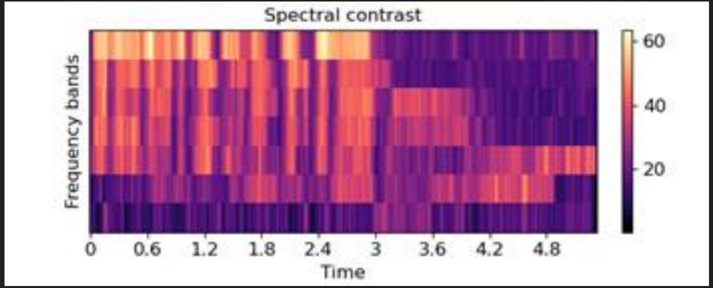
Feature Extraction

Three types of features that are commonly used in audio analysis and have been found to be effective in previous studies* for speech emotion recognition are MFCC, chroma, and spectral contrast.

- **MFCC** captures the distribution of the power across different frequency bands, emphasizing the frequencies that are most relevant for human speech perception.
- **Chroma** features represent the distribution of energy across different pitch classes (notes) within an octave. They help capture the variations in pitch and intonation.
- **Spectral contrast** measures the spectral differences between a sound signal and its background noise, which can provide information about the sharpness and clarity of the sound.




Chroma STFT


Spectral contrast

*ERANNs: Efficient residual audio neural networks for audio pattern recognition
Verbitskiy et al. - Pattern Recognition Letters - 2022

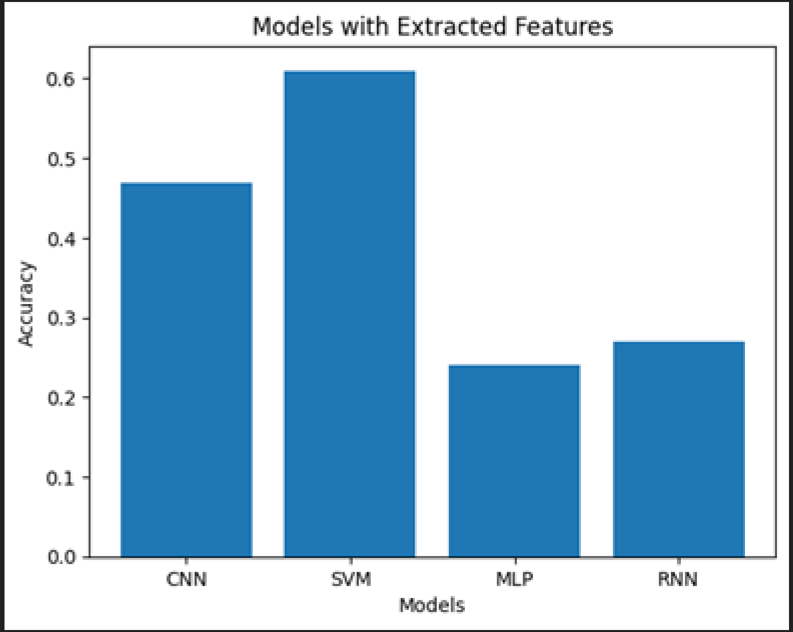
*Speech Emotion Recognition Using MLP Classifier
Poojary et al. - International Journal of Scientific Research in Science and Technology- 2021

Initial Feature Extracted Models

The SVM model showed the best performance initially when classifying emotions, which could be attributed to the fact that the RAVDESS dataset is relatively small, which makes the simpler model structure of the SVM less prone to overfitting.

However, despite the initial success of the SVM, the CNN model was still used for further experimentation, as it is a better choice for more complex problems.

CNNs perform well with audio data*, especially with complex patterns, they are robust to noise, and are more invariant to translation making them advantageous for different datasets and applications.



*ERANNs: Efficient residual audio neural networks for audio pattern recognition
Verbitskiy et al. - Pattern Recognition Letters - 2022

CNN Model

Data Augmentation

Noise Addition:

- Increases model robustness to real-world noisy data
- Adds random noise scaled by a noise factor

Shifting:

- Enhances invariance to pattern starting point
- Rolls audio data left or right by a shift factor

Benefits:

- Diverse samples for model training
- Improved model generalization
- Increased dataset size

Model Architecture

Input:

Audio spectrograms

2 Convolutional layers

- High number of filters
- Activation function: ReLU

Dropout layer

2 Dense layers

- Activation functions: ReLU and Softmax

Output:

Emotion classification prediction

Training Details

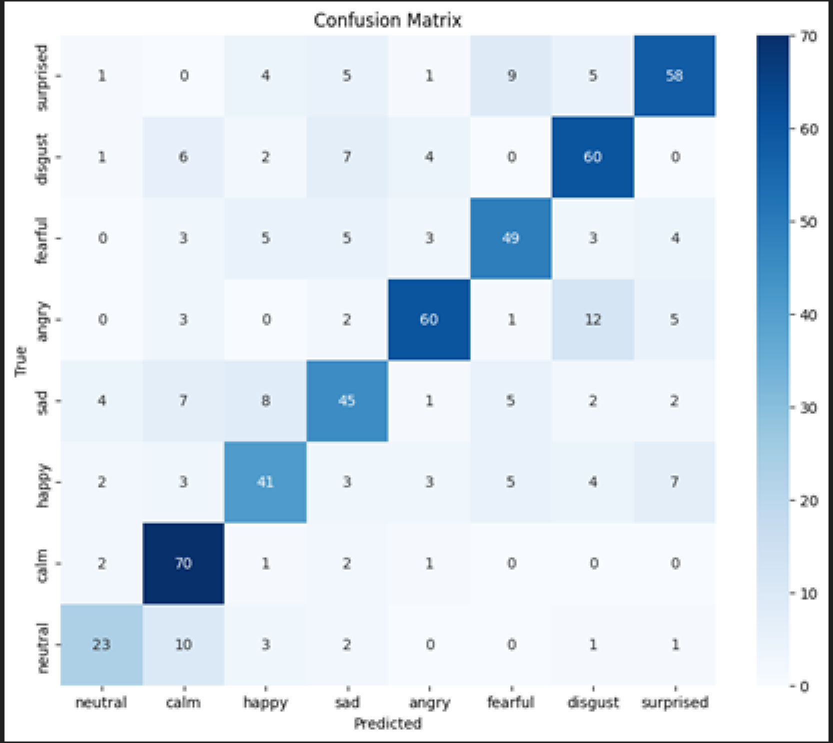
100 epochs

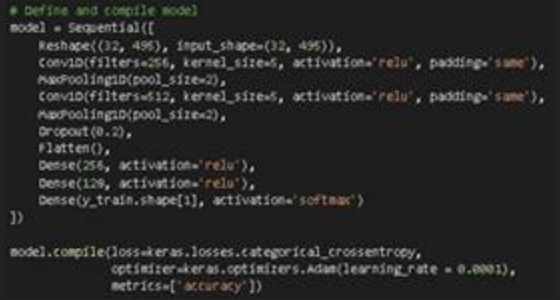
Loss function: Categorical cross-entropy

Optimizer: Adam

Results

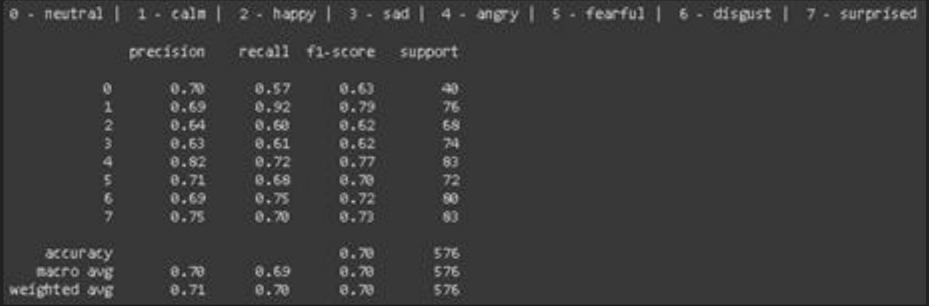
0.71 validation accuracy, 0.70 F1-score





```
# Define and compile model
model = Sequential([
    Reshape((32, 495), input_shape=(32, 495)),
    Conv1D(filters=256, kernel_size=5, activation='relu', padding='same'),
    MaxPooling1D(pool_size=2),
    Conv1D(filters=512, kernel_size=5, activation='relu', padding='same'),
    MaxPooling1D(pool_size=2),
    Dropout(0.2),
    Flatten(),
    Dense(256, activation='relu'),
    Dense(128, activation='relu'),
    Dense(y_train.shape[1], activation='softmax')
])

model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adam(learning_rate = 0.0001),
              metrics=['accuracy'])
```



```
0 - neutral |  1 - calm |  2 - happy |  3 - sad |  4 - angry |  5 - fearful |  6 - disgust |  7 - surprised

              precision    recall  f1-score   support

           0       0.70      0.57      0.63        40
           1       0.69      0.92      0.79        76
           2       0.64      0.60      0.62        68
           3       0.63      0.61      0.62        74
           4       0.82      0.72      0.77        83
           5       0.71      0.68      0.70        72
           6       0.69      0.75      0.72        80
           7       0.75      0.70      0.73        83

    accuracy                           0.70       576
   macro avg       0.70      0.69      0.70       576
weighted avg       0.71      0.70      0.70       576
```

Explaining CNN Model

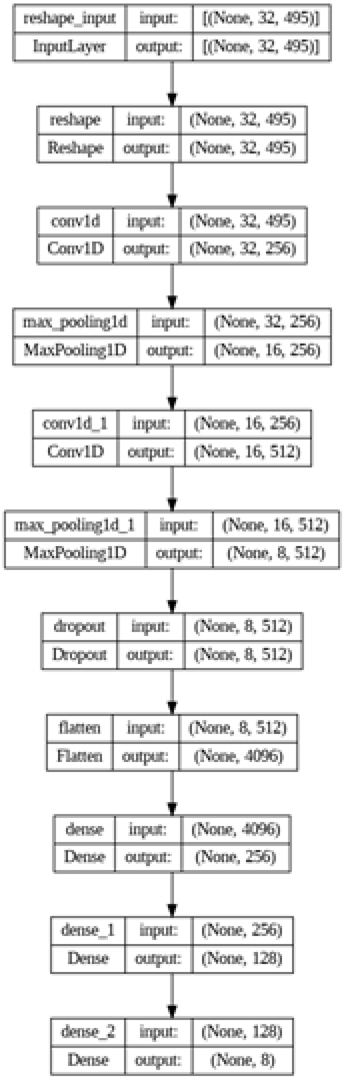
- Conv1D (1st): This is the first 1D convolutional layer with 256 filters. It learns local patterns in the input features by applying multiple filters (kernels) along the time axis. Each filter detects a specific pattern in the input, helping the model capture important information from the audio features for emotion classification.

- MaxPooling1D (1st): Reduces spatial dimensions, extracting significant features and lowering computational complexity.

- Conv1D (2nd): Learns more complex patterns using 512 filters, capturing nuanced information from audio features.

- MaxPooling1D (2nd): Further reduces spatial dimensions, extracting significant features and reducing computational complexity.

- Flatten: This layer flattens the 3-dimensional tensor output from the previous layer into a 1-dimensional tensor. This is necessary for connecting the convolutional layers to the fully connected (dense) layers.



reshape_input InputLayer | input: [(None, 32, 495)] | output: [(None, 32, 495)]

reshape Reshape | input: (None, 32, 495) | output: (None, 32, 495)

conv1d Conv1D | input: (None, 32, 495) | output: (None, 32, 256)

max_pooling1d MaxPooling1D | input: (None, 32, 256) | output: (None, 16, 256)

conv1d_1 Conv1D | input: (None, 16, 256) | output: (None, 16, 512)

max_pooling1d_1 MaxPooling1D | input: (None, 16, 512) | output: (None, 8, 512)

dropout Dropout | input: (None, 8, 512) | output: (None, 8, 512)

flatten Flatten | input: (None, 8, 512) | output: (None, 4096)

dense Dense | input: (None, 4096) | output: (None, 256)

dense_1 Dense | input: (None, 256) | output: (None, 128)

dense_2 Dense | input: (None, 128) | output: (None, 8)

Conclusion

Summary:

- Our CNN model achieved an accuracy of 0.71 and an F1-score of 0.70 on the RAVDESS audio dataset.
- The data augmentation techniques of noise addition and shifting, as well as feature extraction using chroma, MFCC, and spectral contrast improved the performance of the model.
- PCA was explored and for dimensionality reduction in trials, but was unfeasible as it removed time structure of the data

Next Steps:

- The performance of the model can be further improved by splitting the dataset based on gender, as audio depiction of emotions may be expressed differently between males and females.

Future Applications:

- Our approach can be extended to other audio datasets and may be useful for emotion recognition in real-world applications.
- Future work may include incorporating other types of audio features, exploring different architectures for the CNN model, and testing the model on a larger and more diverse dataset to improve generalization performance.