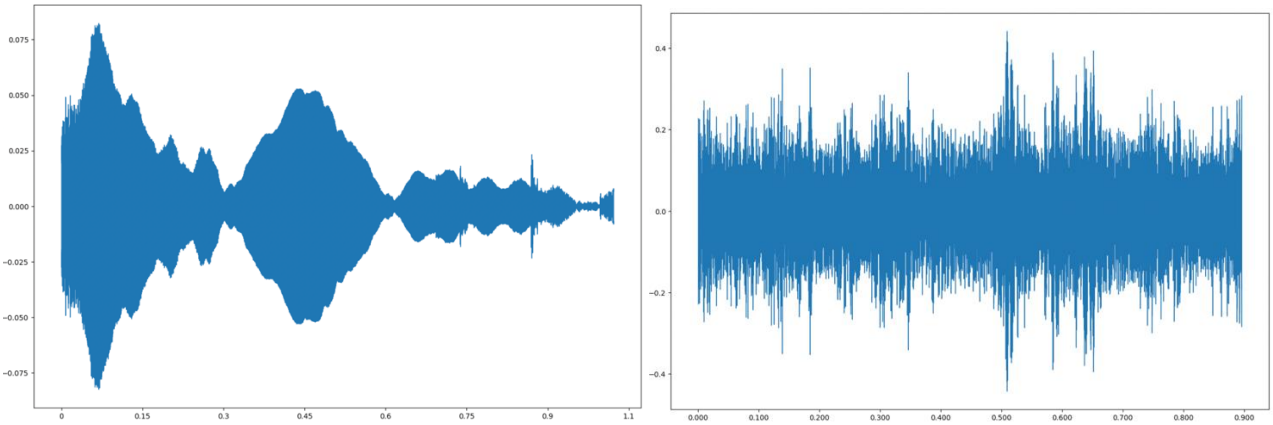
Drones detect: Predicting drones' presence from audio data.

The data I was working with was the Binary Drone Audio dataset that included WAV audio files (Waveform Audio File Format developed by Microsoft and IBM. It's a lossless or raw file format meaning that it doesn't compress the original sound recording)having two folders "yes_drone" and "unknown". The goal of the project was to train a model that detects drone sounds from background sounds with the most high accuracy and high performance metrics (false positives less than 2% and true positives over 75%). The number of the unknown audios was more than 10000 of samples while the drone audios were only in the amount of 1332. To avoid the bias of the model towards one of the classes, and to make the data balanced (or at least not imbalanced), I worked only with the 1500 files from "unknown" folder.

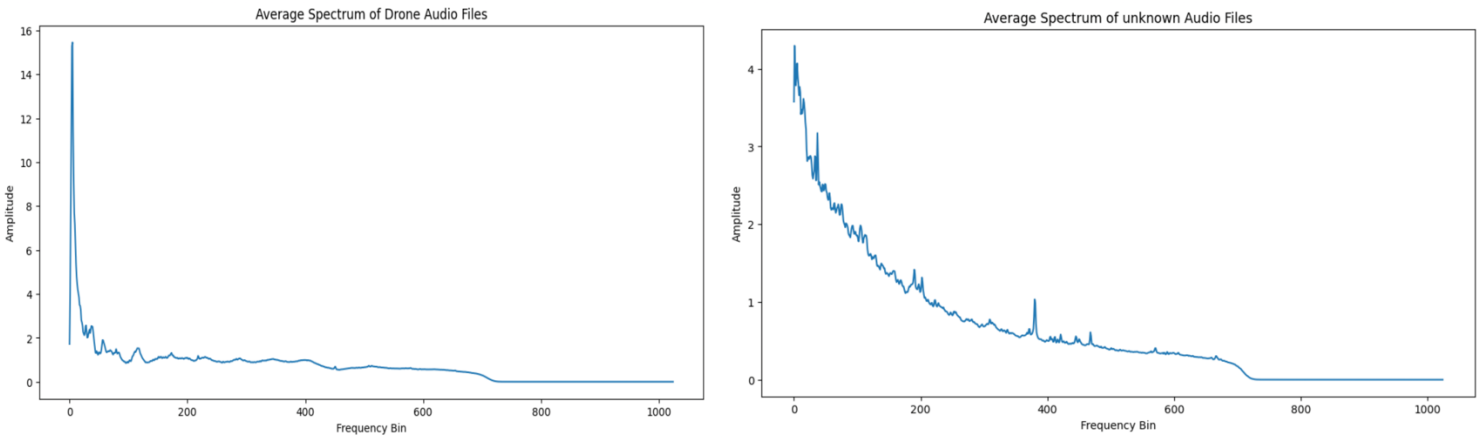
The environment used was Kaggle. Firstly, I uploaded some audios from both classes to see the exemplar waveforms. The waveform shows how the sound pressure of the audio signal varies over time.



The noticeable thing about the audios, that almost all of them are around one second of the time and the amplitude of the waveforms isn't that big. Other thing to mark is that librosa.load by default resamples the audio to 22050 Hz, while the actual sample rates are 16000 Hz (checked on several audio files).

Exploratory data analysis.

The analysis started from the average frequency spectrums of audio files.



The x-axis represents the frequency bins, which are essentially segments of the frequency range within the audio signal. Each bin corresponds to a specific frequency interval, which is determined by the sampling rate and the method used for the Fourier Transform (Short-Time Fourier Transform or STFT). The y-axis represents the amplitude of the audio signal. It indicates the strength or loudness of the frequencies present in the signal.

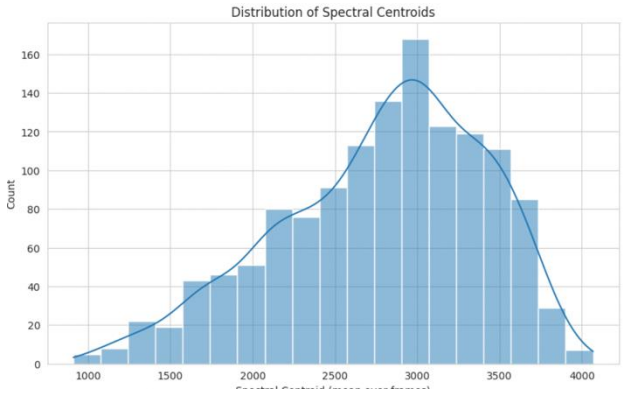
There is a prominent peak at the beginning of the plot for drone audios, which suggests that there is a strong amplitude at a low frequency. This is typical for drone audio as the sound of the rotors or propellers usually generates significant energy at lower frequencies. After the initial peak, the amplitude decreases sharply and then gradually levels off. This indicates that the energy of the audio signal is much lower at higher frequencies.

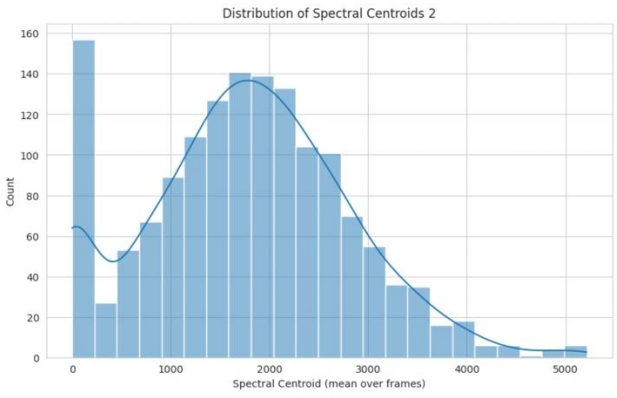
The general trend of the plot for unknown audios is a decay in amplitude as the frequency increases. This is typical for audio signals, where lower frequencies tend to have higher amplitudes. The plot shows several peaks and troughs, which indicates the presence of dominant frequencies where the amplitude is significantly higher, as well as valleys where certain frequencies are less present or absent. Compared to the previous drone audio plot, the amplitudes here are much lower, suggesting a different types of sounds that doesn't have as much energy concentrated in lower frequencies, thus, leading to more complex frequency spectrum.

Next step in EDA was to analyse the features commonly used in audio signal processing: Spectral centroids, Zero crossing rates, Spectral bandwidths, Spectral rolloffs, Mfcc coeeficients, Mel spectrograms, RMS Energy.

The spectral centroid.

The spectral centroid is a measure used to characterize a spectrum. It indicates where the center of mass of the spectrum is located and is perceived as the "brightness" of a sound. It is calculated as a weighted mean of the frequencies present in the sound, with their magnitudes as the weights.





The distribution is somewhat bell-shaped for drone spectral centroids, indicating that most of the spectral centroid values are concentrated around the middle range, with fewer occurrences toward the low and high ends. This is typical of Gaussian-like distributions. The values range from approximately 1000 Hz to 4000 Hz, with most values concentrated around 2000 Hz to 3000 Hz. The spectral centroids are more closely clustered around the mean value.

The distribution for unknown audios is also bell-shaped but with a broader base and a less pronounced peak (around 1500 Hz to 2000 Hz). This suggests a greater variance in the spectral centroid values. The spectral centroid values extend from around 0 Hz to beyond 5000 Hz, which is a wider range than the first plot.

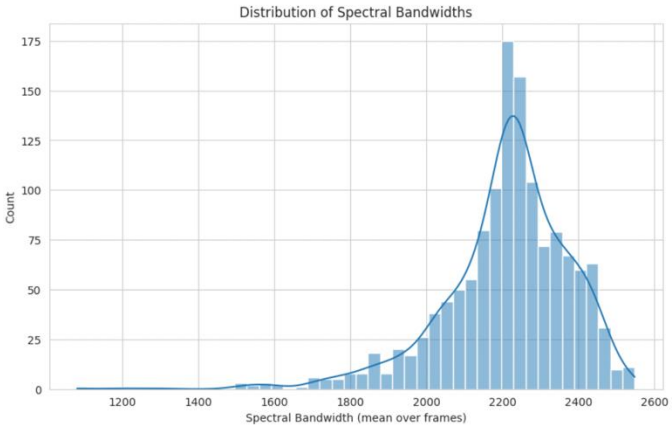
The most common spectral centroid value in the first plot is higher than in the second one, which could imply that the set of drone audio files generally has a brighter or more high-frequency dominant sound.

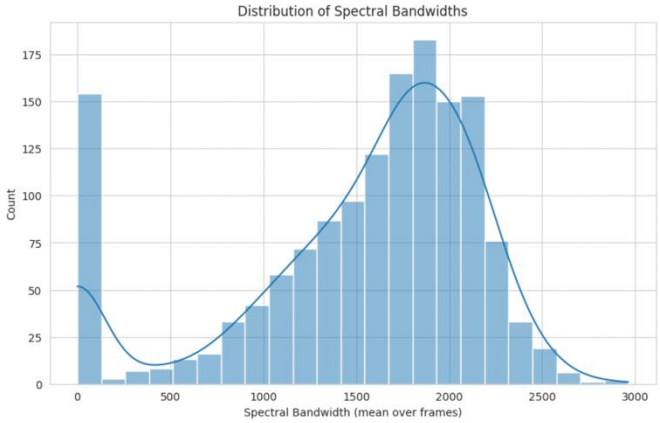
The zero-crossing rate.

The zero-crossing rate is the rate at which the audio signal changes sign, or crosses zero. It is a measure of the noisiness of the sound. Both sets show the most common zero-crossing rates are in the lower range, but the first plot has a mean (0.2) that is slightly higher than the second plot (0.16-0.17). Spread of values is again wider for the unknown audio dataset. The drone plot is more symmetric, suggesting a more uniform distribution of the zero-crossing rates, whereas the second plot is left-skewed, indicating a prevalence of lower zero-crossing rate values. However, the skewness in the second plot might suggest that the unknown audio files are generally less noisy than those in the drone dataset.

Spectral Bandwidths

The spectral bandwidth refers to the width of a band of frequencies and is defined as the width of the spectrum at half the spectral peak's maximum height. It can be used to describe the spread of the spectrum and is related to the timbre of the sound.

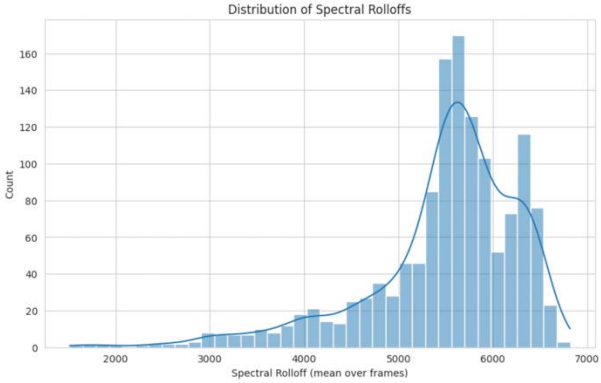


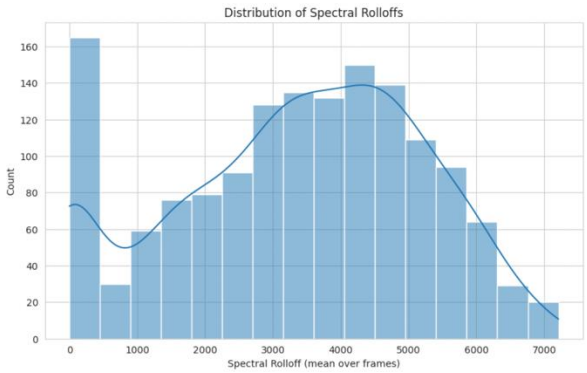


Both plots have the bulk of their values in a central range, but the first plot has a higher central tendency around 2000 Hz whereas the second plot's central tendency is slightly lower around 1500-2000 Hz. Both show a right skew, indicating a tail of frames with higher spectral bandwidth values. However, the second plot seems to have a longer tail to the right, suggesting more frames with very wide spectral bandwidths. The second plot has a wider spread of spectral bandwidth values, suggesting a greater variety in the frequency content of the audio frames or files.

Spectral Rolloffs

The spectral rolloff is a measure of the shape of the signal's spectrum. It is the frequency below which a certain percentage of the total spectral energy, often between 85% and 95%, is contained.





The drone audios' plot shows that there are two significant groups of frames or files, one centered around 4000 Hz and another around 6000 Hz. The unknown audios' plot, while also showing multiple groups, has a stronger concentration around 3000 Hz.

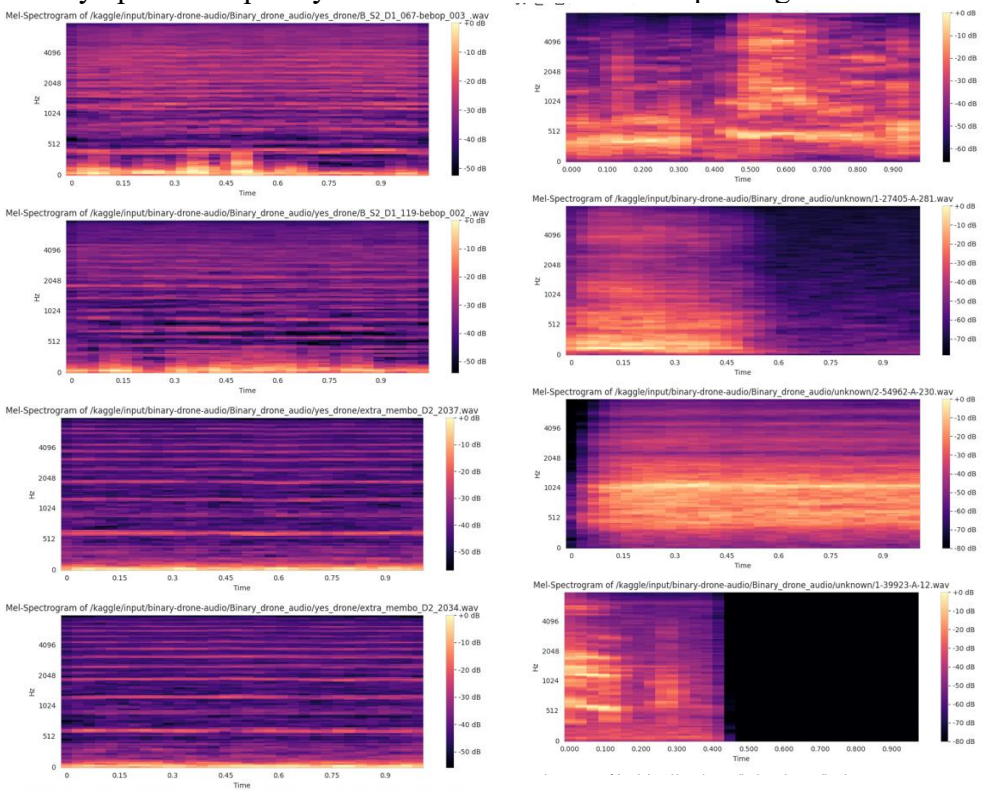
MFCC Coefficients (Mel-frequency Cepstral Coefficients)

MFCCs are coefficients that make up an MFC. They are derived from a type of cepstral representation of the audio clip (a nonlinear "spectrum-of-a-spectrum"). These are obtained by taking the Fourier transform of the log power spectrum of a signal and then mapping the powers of the spectrum obtained onto the mel scale, followed by the inverse Fourier transform. They are widely used in speech and audio processing as they mimic the human ear's behavior and are robust to noise.

For Both sets, 20 MFCCs were used for analysis. The x-axis for the drone set has much narrower values suggesting less variation in the audio features that the coefficient represented. However, the shapes of MFCCs of the unknown audio dataset were more symmetrical compared to drones, suggesting consistency. As for the common value, the unknown dataset had the value of around 0, suggesting that most of the audios have very silent spectral characteristic.

Mel Spectrograms

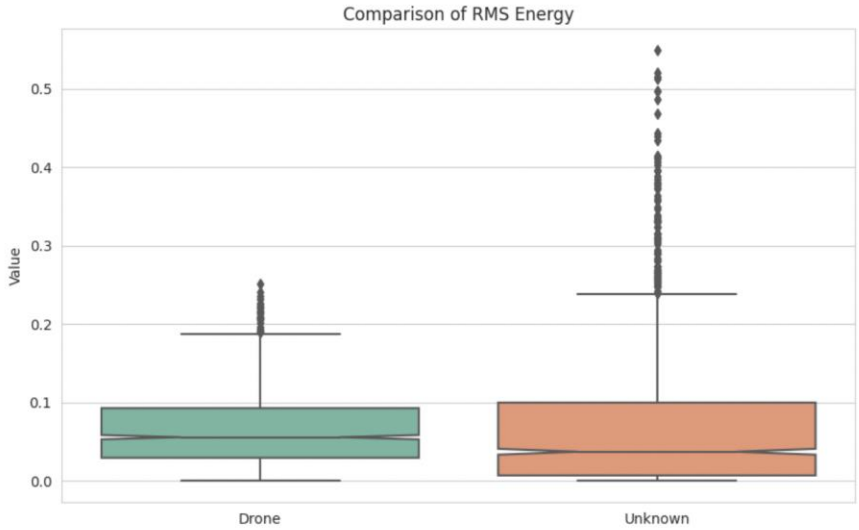
A mel spectrogram is a spectrogram where the frequencies are converted to the mel scale, which is a perceptual scale of pitches judged by listeners to be equal in distance from one another. The purpose of the mel scale is to mimic the human ear's response more closely than the linearly-spaced frequency bands used in the standard spectrogram.



The Mel spectrograms of drone audio dataset show very consistent spread of amplitudes over the time (the left screen), whereas the unknown dataset has great variability of amplitudes over the time.

RMS Energy

Root Mean Square (RMS) energy is a measure of the power of the audio signal. It is the square root of the arithmetic mean of the squares of the signal's values. RMS energy is often used to compute the loudness of an audio signal.



The unknown group has higher energy levels on average compared to the drone group. However, drone group has much more consistent RMS energy level, while unknown group is more varied and has a lot of extreme outliers.

Data Pre-processing. Features Extraction.

Checking again on MFCCs, STFT, and Mel-Spectrograms, I decide to choose time-averaged MFCCs as the features for audio analysis. The reasons are following:

1. Averaged MFCCs provide a compact representation of the spectral properties of the sound over the entire audio clip. This reduces the dimensionality of the feature space, which can help to alleviate the curse of dimensionality in machine learning models.
2. MFCCs are less sensitive to variations in the audio signal that are not relevant to the tasks at hand, such as minor changes in pitch or speed. They focus on the timbral characteristics of the sound, which are more stable.
3. By using the mean of the MFCCs, I can reduce the amount of data that needs to be processed by the machine learning model. This can make training and inference faster and require less computational resources.
4. MFCCs are designed to mimic the human ear's response to sound, making them particularly well-suited for tasks related to recognition. This is because they take into account the non-linear perception of frequency bands by the human auditory system.
5. MFCCs are a well-established feature in audio signal processing and have been proven effective in various tasks.
6. MFCCs are often used with classical machine learning models such as SVMs, random forests, and even with neural networks. The time-averaged vector is easily ingested by these models without requiring complex input structures.

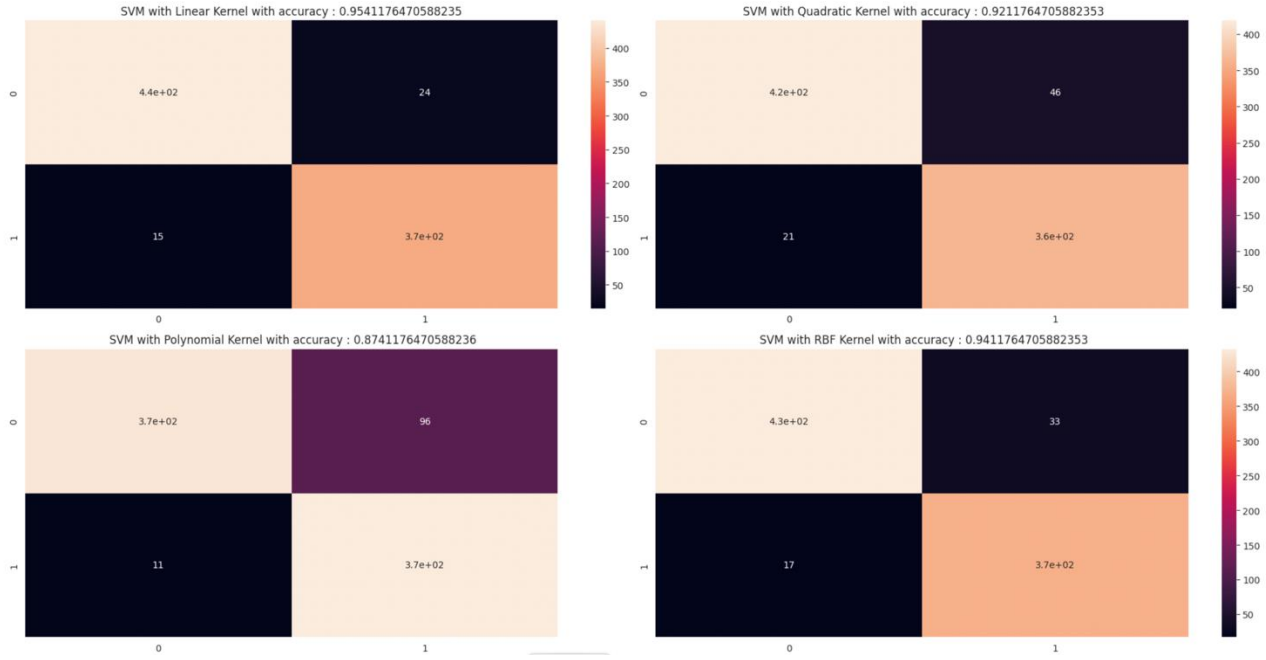
The extracted features have a shape of (40, 45) in amount of 2882 – unknown dataset mixed with drone dataset. To work further, I created the data frame, containing ‘features’ column and ‘class’ column (‘unknown’, ‘Drone’). In further analysis, I mapped the “Unknown” as 0 label, and “Drone” as 1 label.

Modelling.

SVM

Support Vector Machine (SVM) are known for their effectiveness in handling high-dimensional spaces. MFCC features, even when averaged, can still form a high-dimensional feature space, especially if other supplementary features are included. SVMs can handle this complexity efficiently. Moreover, SVMs are inherently suited for binary classification but can also be extended to multi-class classification problems, common in audio analysis.

To make some experimenting, I decided to make for types of Kernels for SVM: linear, degree 3 polynomial, degree 2 polynomial (quadratic), RBF kernel. The results are following:



The best results were shown by Linear Kernel (Accuracy is 0.9541176470588235), and RBF (Accuracy is 0.9411764705882353).

As for the confusion matrixes, Linear SVM has TP=441, FN = 15, FP = 24, TN = 370. Based on the provided confusion matrix, the calculated performance metrics are as follows:

- Precision: 0.948 (94.8%)
- Recall: 0.967 (96.7%)
- Specificity: 0.939 (93.9%)
- F1 Score: 0.958 (95.8%)

RBF SVM has TP = 432, FN = 17, FP = 33, TN = 368. Based on the provided confusion matrix, the calculated performance metrics are as follows:

- Precision: 0.929 (92.9%)
- Recall: 0.962 (96.2%)
- Specificity: 0.918 (91.8%)
- F1 Score: 0.945 (94.5%)

ANN

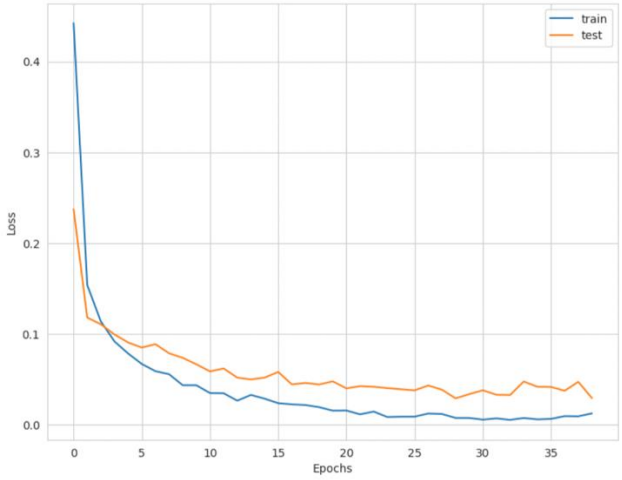
ANNs are capable of capturing complex and non-linear relationships within the data. Audio data contains intricate patterns that are not easily separable by linear methods. ANNs, through their hidden layers and non-linear activation functions, can learn these complexities. Moreover, ANNs can learn interactions between different features. In the case of MFCCs, which are individual coefficients, an ANN can learn how combinations of these coefficients relate to specific audio classes.

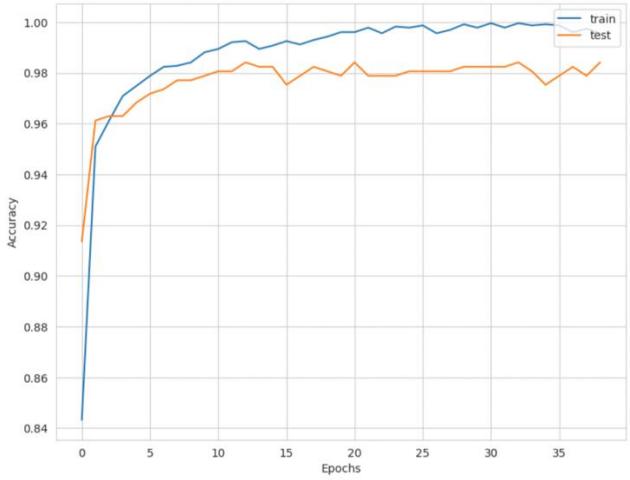
The model is built using Keras and follows a Sequential architecture. It begins with a Dense layer of 100 neurons, tailored to process input vectors representing the 40 MFCC features. The model incorporates the ReLU activation function for non-linearity and dropout layers with a rate of 0.5 for regularization to combat overfitting.

The architecture expands to include a second Dense layer with 200 neurons, followed by a reduction back to 100 neurons in a subsequent layer, both interleaved with dropout layers for robustness. The final layer is a Dense layer with a neuron count equal to the number of class labels, employing a softmax activation function for multi-class classification.

The model is compiled with the 'sparse_categorical_crossentropy' loss function, 'accuracy' as the performance metric, and the 'adam' optimizer. The number of total parameters is equal to 44602.

The model is trained for 100 epochs with a batch size of 128. To enhance the training process, a ModelCheckpoint callback is employed, which saves the best model to a specified file path only when there is an improvement in validation accuracy. Resulting test accuracy is approximately 97.9% (0.9788235425949097).





From the left graph, which represents the model's loss, we observe a sharp decline in the training loss during the initial epochs, indicating rapid learning. The validation loss also decreases but levels off, which suggests that the model is generalizing well to the unseen data in the validation set without overfitting significantly, as the two curves converge and remain close together.

The right graph is indicative of the model's accuracy over the same epochs. The training accuracy increases steeply initially and then plateaus, which is expected behavior as the model begins to fit the training data. The validation accuracy also rises quickly and appears to stabilize, mirroring the training accuracy quite closely throughout the process. This close alignment between training and validation accuracy is a good sign that the model is performing consistently and not overfitting.

The final test accuracy, as previously mentioned, is approximately 97.9%. When combined with the patterns observed in these graphs, it can be inferred that the model has learned effectively and generalized well to new data. The high accuracy coupled with the stable loss and accuracy curves over time underscores the success of the training regimen and the suitability of the model architecture for the audio classification task at hand.

CNN

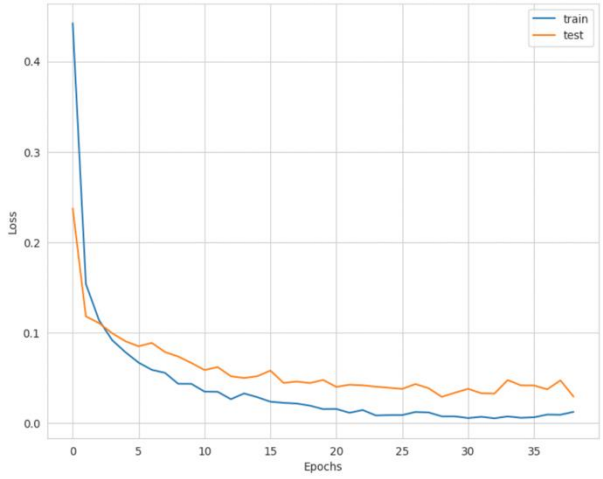
CNNs are known for their ability to learn hierarchical representations of data. In the context of audio, this means a CNN can automatically learn to identify features that are important for classification directly from the MFCCs, without the need for manual feature engineering. Also, CNNs use shared weights in their convolutional filters, which means they require fewer parameters than fully connected networks of a similar size. This can lead to more efficient training and less risk of overfitting.

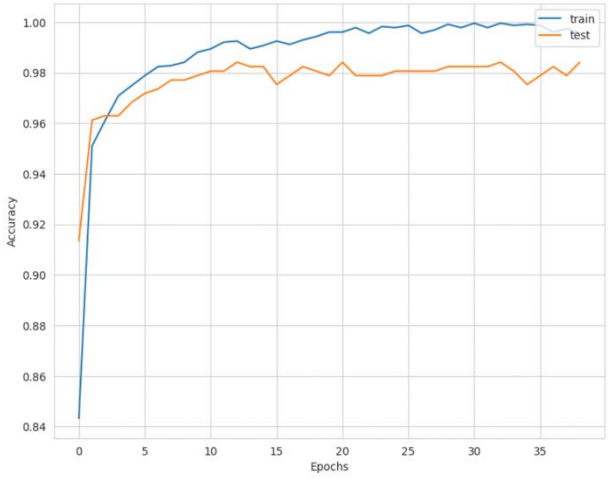
The model's architecture, designed using TensorFlow and Keras, includes two convolutional layers, each followed by batch normalization and max pooling for feature extraction and dimensionality reduction. The first convolutional layer has 32 filters, while the second layer expands this to 64 filters, both employing a kernel size of (1, 3). The LeakyReLU activation function is utilized in the second convolutional layer to introduce non-linearity and mitigate the "dying ReLU" problem.

Following the convolutional base, the model features a flattening layer that transitions to a fully connected network. This dense segment comprises 128 neurons and includes a dropout rate of 0.3 to prevent overfitting. The output layer consists of 2 neurons, corresponding to the classification categories, and employs a softmax activation function for probabilistic distribution.

The model is compiled with the Adam optimizer and categorical crossentropy as the loss function, focusing on optimizing accuracy as the primary metric. The designed CNN is tailored to process inputs of shape (1, max_length, 1), aligning with the time-series nature of audio data processed into MFCC form. The total number of parameters is equal to 72642.

The model was trained employing a robust suite of Keras callbacks to enhance performance and efficiency. Early Stopping was configured to monitor validation loss, with a patience of 10 epochs, restoring weights to the best observed model state upon minimal improvement. A Model Checkpoint was set to save the model with the highest validation accuracy to a designated file. Training was executed over a maximum of 100 epochs with a batch size of 128, while actual run time was curtailed by the Early Stopping mechanism, ensuring training concluded once no significant progress was observed. Resulting test accuracy is approximately 98.2% (0.9823633432388306).





From the left graph, which represents the model's loss, we observe that training and validation loss both show a sharp decrease in the initial epochs, indicating rapid initial learning and successful reduction of error in predictions. As the epochs continue, both curves flatten out, which suggests that the model is starting to converge and that further improvements are incremental. The training and validation loss remain close together throughout the training process, which suggests good generalization and minimal overfitting.

The right graph is indicative of the model's accuracy over the same epochs. The training accuracy increases quickly at the beginning and starts to plateau, which is typical as the model begins to fit the training data.The validation accuracy also rises sharply and then fluctuates slightly around a high value, closely following the training accuracy. There's no significant divergence between training and validation accuracy, indicating that the model is generalizing well to new data and not just memorizing the training set.

Overall, the graphs reflect a well-performing model with high accuracy and low loss on both the training and validation sets. The close correspondence between the training and validation lines in both graphs indicates that the model is not overfitting and is likely to perform well on unseen data, which is corroborated by the reported high test accuracy of approximately 98.2%.