Audio Classification using Deep Learning

A Project Report Submitted by

Aman Pandey

Harsh Kumar

Somnath Mukherjee

In partial fulfilment of the requirements for the award of degree of

AI/ML Internship

Design and Innovation Centre, Panjab University, Chandigarh

July 2024

Declaration

I hereby declare that the work presented in this Project Report titled “Audio Classification using Deep Learning” Report submitted to the Design Innovation Centre, Panjab University, Chandigarh is a bona fide record of the research work carried out under the supervision of Professor Naveen Agarwal and Garima Joshi. The contents of this Project Report in full or in parts, have not been submitted to, and will not be submitted by me to, any other Institute or University in India or abroad.

Signature

Aman Pandey, UE215008

Harsh Kumar, UE215039

Somnath Mukherjee, UE225092

Certificate

This is to certify that the Project Report titled “Audio Classification using Deep Learning” Report, submitted by *Aman Pandey (UE215008), Harsh Kumar (UE215039), Somnath Mukherjee* (UE225092) to the Design Innovation Centre, Panjab University, Chandigarh for the award of the degree of Name of the Degree, is a bona fide record of the research work done by them under my supervision. To the best of my knowledge, the contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Signature

Prof Naveen Agarwal

Prof Garima Joshi

Acknowledgement

We would like to extend our sincere gratitude to all those who have supported and guided us throughout the development of our project *"Audio Classification using Deep Learning"*.

First and foremost, we express our deepest appreciation to *Prof Naveen Agarwal* and *Prof Garima Joshi*, whose expertise, guidance, and encouragement have been invaluable to us. Their insightful suggestions and constant support helped us navigate through various challenges and enriched our learning experience.

We also want to thank our mentors for their unwavering support and encouragement, which has been a constant source of motivation. Finally, we acknowledge each other, for our dedication, teamwork, and relentless pursuit of excellence in bringing this project to fruition. It has been a rewarding journey of innovation and collaboration. Thank you all for making this project a success.

Abstract

Audio classification using deep learning has emerged as a powerful approach to automatically identifying and categorizing audio signals into predefined classes. Leveraging advancements in neural network architectures, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), this study explores the effectiveness of deep learning techniques in audio classification tasks. The proposed methodology involves preprocessing audio signals through normalization and segmentation, followed by feature extraction using mel-frequency cepstral coefficients (MFCCs) and spectrograms to capture the essential characteristics of the audio data. A hybrid CNN-RNN model is then employed to learn spatial and temporal patterns within the audio signals. Comprehensive evaluation metrics, including accuracy, precision, recall, and F1-score, are utilized to assess the model's performance. Additionally, robustness testing is conducted to ensure the model's reliability under various conditions such as noisy data, adversarial attacks, and data distribution shifts. The results demonstrate that deep learning models can achieve high accuracy and generalize well to unseen data, making them suitable for a wide range of audio classification applications, from speech recognition to environmental sound classification. This study underscores the potential of deep learning in advancing the field of audio analysis, paving the way for more sophisticated and intelligent audio processing systems.

Table Of Contents

Introduction

Audio classification is a critical task in signal processing and machine learning, involving the automatic identification and categorization of audio signals into predefined categories. Historically, this task has relied on manual feature extraction and statistical methods, which often required domain-specific knowledge and were limited in their ability to handle complex and diverse audio signals. With the advent of deep learning, particularly neural networks, audio classification has undergone a significant transformation. Deep learning models can automatically learn and extract intricate patterns and features from raw audio data, leading to more accurate and scalable solutions.

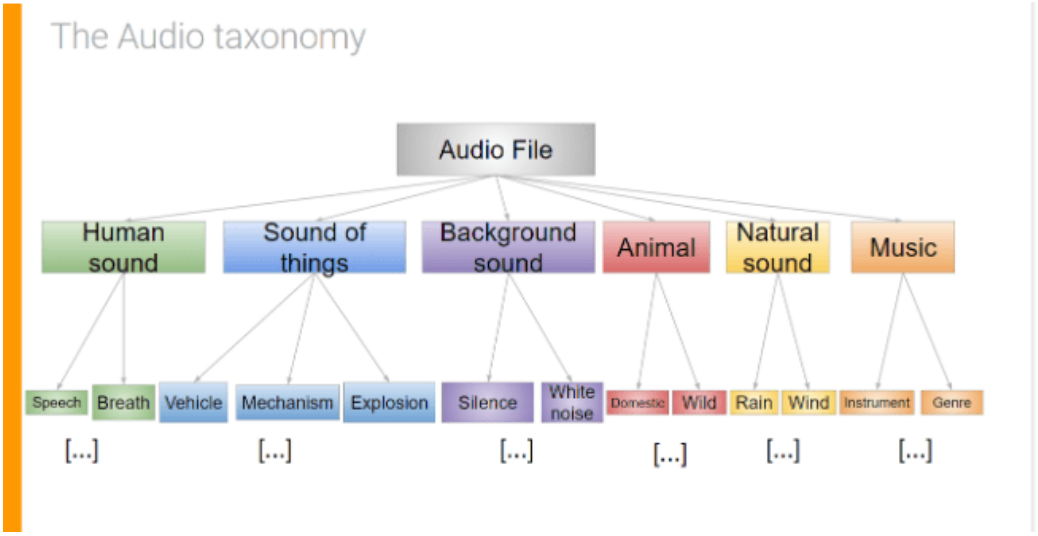


Fig-1 Sound Classification

Audio classification techniques using deep learning stems from its vast potential applications and benefits across multiple fields:

- **Music Industry**: Enhancing music recommendation systems, genre classification, and mood analysis, thereby improving user experience and engagement.
- **Healthcare**: Facilitating the detection of anomalies in biomedical audio signals such as heartbeats and respiratory sounds, which can assist in early diagnosis and monitoring of diseases.
- **Security and Surveillance**: Implementing systems that can recognize specific sounds such as alarms, gunshots, or breaking glass to enhance safety and security measures.
- **Customer Service**: Improving sentiment analysis and automated response systems in call centres, leading to better customer support and satisfaction.

- **Smart Devices**: Enabling voice-activated assistants and smart home devices to recognize and respond to a wide range of audio commands and environmental sounds.

These applications highlight the importance of accurate and efficient audio classification systems, driving the need for advanced methodologies like deep learning.

Objective

The primary objective of this project is to develop a robust and efficient deep learning-based audio classification system that can accurately identify and categorize different types of audio signals. Specific objectives include:

1. **Data Collection**: Assembling a diverse dataset of audio samples from various categories.
2. **Preprocessing**: Applying preprocessing techniques to enhance the quality of the audio data and extract meaningful features.
3. **Model Development**: Designing, training, and optimizing deep learning models to learn complex audio features.
4. **Evaluation**: Assessing the performance of the models using appropriate evaluation metrics.
5. **Optimization**: Fine-tuning the models to maximize their accuracy and efficiency.
6. **Deployment**: Implementing the trained model in a real-world application for practical use.

Scope of the Project

The scope of this project includes:

1. **Data Sources**: Utilizing publicly available datasets such as UrbanSound8K, ESC-50, and GTZAN, covering a range of sound categories including environmental sounds, speech, and music.
2. **Preprocessing Techniques**: Employing methods like noise reduction, normalization, and feature extraction techniques such as Mel-frequency cepstral coefficients (MFCCs) and spectrograms.
3. **Deep Learning Models**: Exploring and comparing different architectures including Convolutional Neural Networks (CNNs) and Artificial Neural Networks (ANNs).
4. **Evaluation Metrics**: Using metrics like accuracy, precision, recall, F1-score, and confusion matrices to evaluate model performance.
5. **Implementation Tools**: Leveraging tools and frameworks such as Python, TensorFlow, Keras, and Librosa for development and experimentation.

Description

This dataset contains 8732 labeled sound excerpts (<=4s) of urban sounds from 10 classes: air_conditioner, car_horn, children_playing, dog_bark, drilling, enginge_idling, gun_shot, jackhammer, siren, and street_music. The classes are drawn from the urban sound taxonomy.

All excerpts are taken from field recordings uploaded to www.freesound.org. The files are pre-sorted into ten folds (folders named fold1-fold10) to help in the reproduction of and comparison with the automatic classification results reported in the article above.

In addition to the sound excerpts, a CSV file containing metadata about each excerpt is also provided.

AUDIO FILES INCLUDED

8732 audio files of urban sounds (see description above) in WAV format. The sampling rate, bit depth, and number of channels are the same as those of the original file uploaded to free sound (and hence may vary from file to file).

META-DATA FILES INCLUDED

UrbanSound8k.csv

This file contains meta-data information about every audio file in the dataset. This includes:

slice_file_name:

The name of the audio file. The name takes the following format: [fsID]-[classID]-[occurrenceID]-[sliceID].wav, where:

[fsID] = the Freesound ID of the recording from which this excerpt (slice) is taken

[classID] = a numeric identifier of the sound class (see description of classID below for further details)

[occurrenceID] = a numeric identifier to distinguish different occurrences of the sound within the original recording

[sliceID] = a numeric identifier to distinguish different slices taken from the same occurrence

* fsID:

The Freesound ID of the recording from which this excerpt (slice) is taken

* start

The start time of the slice in the original Freesound recording

* end:

The end time of slice in the original Freesound recording

* salience:

A (subjective) salience rating of the sound. 1 = foreground, 2 = background.

* fold:

The fold number (1-10) to which this file has been allocated.

* classID:

A numeric identifier of the sound class:

0 = air_conditioner

1 = car_horn

2 = children_playing

3 = dog_bark

4 = drilling

5 = engine_idling

6 = gun_shot

7 = jackhammer

8 = siren

9 = street_music

* class:

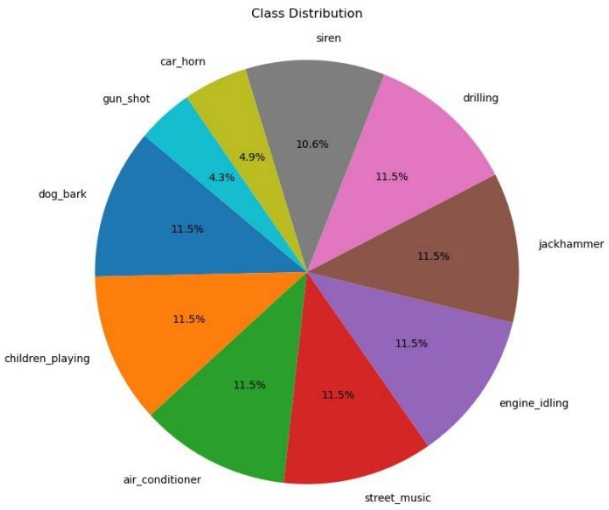


Fig-2 Distribution of Audio Classes in Dataset

The class name: air_conditioner, car_horn, children_playing, dog_bark, drilling, engine_idling, gun_shot, jackhammer, siren, street_music.

Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a critical process in data science that involves examining and summarizing the main characteristics of a dataset, often with visual methods. EDA helps in understanding the data, identifying patterns, detecting anomalies, testing hypotheses, and informing the next steps of data processing and modelling. The main objectives of EDA include:

- Understanding Data Distribution: Analysing the spread and central tendency of the data.
- Identifying Outliers: Detecting any anomalies or outliers in the data.
- Detecting Relationships: Finding relationships between variables.
- Data Cleaning: Identifying and handling missing or inconsistent data.
- Data Transformation: Deciding on transformations needed for the data.

Use of EDA in Audio Classification

EDA plays a crucial role in the process of audio classification, which involves categorizing audio signals into predefined classes. Here's how EDA is used in the context of audio classification:

1. Understanding the Audio Data

- **Waveform Analysis**: Plotting waveforms of audio signals to visualize their amplitude variations over time. This helps in understanding the basic structure of the audio signals.
    - **Example**: Visualizing the waveform of a speech signal to see patterns like silences and peaks corresponding to spoken words.
- **Spectrograms**: Visualizing the frequency spectrum of the audio signal over time. Spectrograms are essential for understanding how the energy of the signal is distributed across different frequencies.
    - **Example**: Using spectrograms to analyse bird calls, which often have distinct frequency patterns.
- **Summary Statistics**: Calculating statistics such as mean, standard deviation, and range of audio features like pitch, loudness, and duration.
    - **Example**: Summarizing the average pitch and duration of audio clips in a music dataset.

2. Identifying Patterns and Relationships

- **Feature Distribution**: Analysing the distribution of audio features like Mel-frequency cepstral coefficients (MFCCs), chroma features, and zero-crossing rate. Histograms and density plots are commonly used for this purpose.

- **Example**: Plotting the distribution of MFCCs to identify common patterns in speech vs. music.

- **Correlation Analysis**: Calculating and visualizing correlations between different audio features to identify which features are related.
    - **Example**: Examining the correlation between MFCCs and spectral contrast to understand their combined effect on classification performance.

3. Feature Engineering

- **Creating New Features**: Generating new features that capture important characteristics of the audio signals. For example, extracting rhythmic patterns, timbre features, or harmonic content.
    - **Example**: Extracting rhythmic features for classifying different genres of music.
- **Dimensionality Reduction**: Applying techniques like Principal Component Analysis (PCA) to reduce the dimensionality of audio features while retaining important information.
    - **Example**: Using PCA to reduce the number of MFCC features for faster and more efficient modelling.

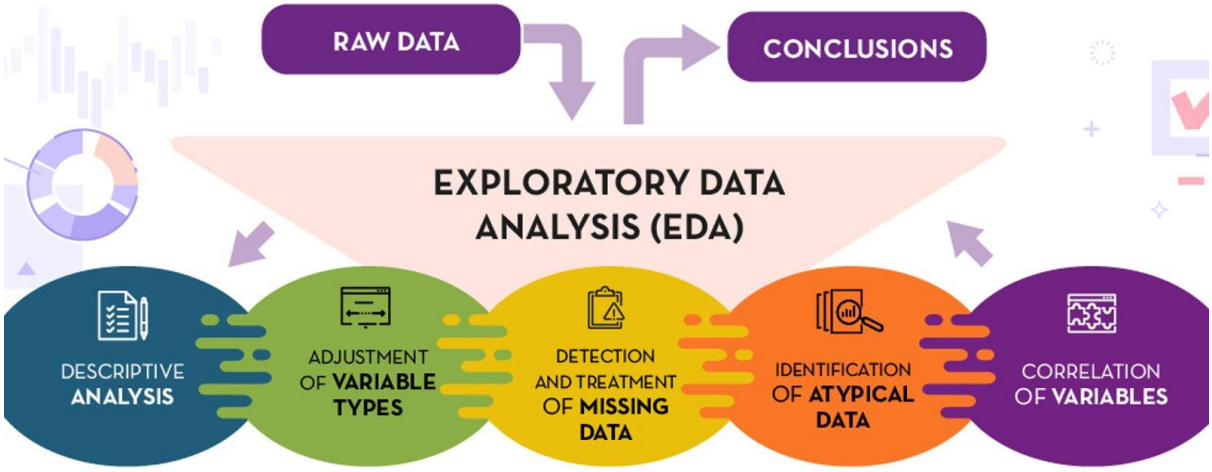


Fig-3 Exploratory Data Analysis

By employing various statistical and visual techniques, EDA provides insights that are crucial for building robust and accurate audio classification models.

Data Preprocessing

Data preprocessing is a critical step in preparing audio data for classification tasks. It involves cleaning and transforming raw audio signals into a format that can be effectively used by machine learning models. This process includes several key steps: audio segmentation, feature extraction, and normalization.

Key Steps in Audio Data Preprocessing:

1. **Audio Segmentation**
2. **Feature Extraction**
3. **Normalization**

1. Audio Segmentation

Audio Segmentation involves dividing an audio signal into smaller chunks or segments. This step is essential for processing long audio recordings and extracting meaningful patterns from shorter segments. There are two primary types of segmentation:

- **Fixed-length Segmentation**: The audio signal is divided into fixed-length segments (e.g., 1-second chunks). This is simple and efficient but may not always capture the structure of the audio data.
- **Dynamic Segmentation**: The segmentation is based on specific events or changes in the audio signal, such as pauses in speech or changes in music tempo. This approach can be more complex but often yields better results for certain applications.

2. Feature Extraction:

Feature extraction involves transforming raw audio data into a set of features that can be used by machine learning algorithms. These features capture important characteristics of the audio signal.

Common Features:

- **Spectrograms:** Visual representations of the spectrum of frequencies in a sound signal as they vary with time.
- **Mel-Frequency Cepstral Coefficients (MFCCs):** Capture the power spectrum of the audio signal and are widely used in speech and music processing.
- **Spectral Centroid, Bandwidth, Contrast, and Rolloff:** Various spectral features capturing different aspects of the frequency distribution.

3. Normalization:

Normalization ensures that the extracted features have a consistent scale and distribution, which helps in improving the performance and convergence of machine learning models.

- **Techniques:**
    - **Mean Normalization:** Subtracting the mean and dividing by the range of the feature values.
    - **Standardization:** Subtracting the mean and dividing by the standard deviation, resulting in features with zero mean and unit variance.
    - **Min-Max Scaling:** Scaling features to a fixed range, usually [0, 1] or [-1, 1].

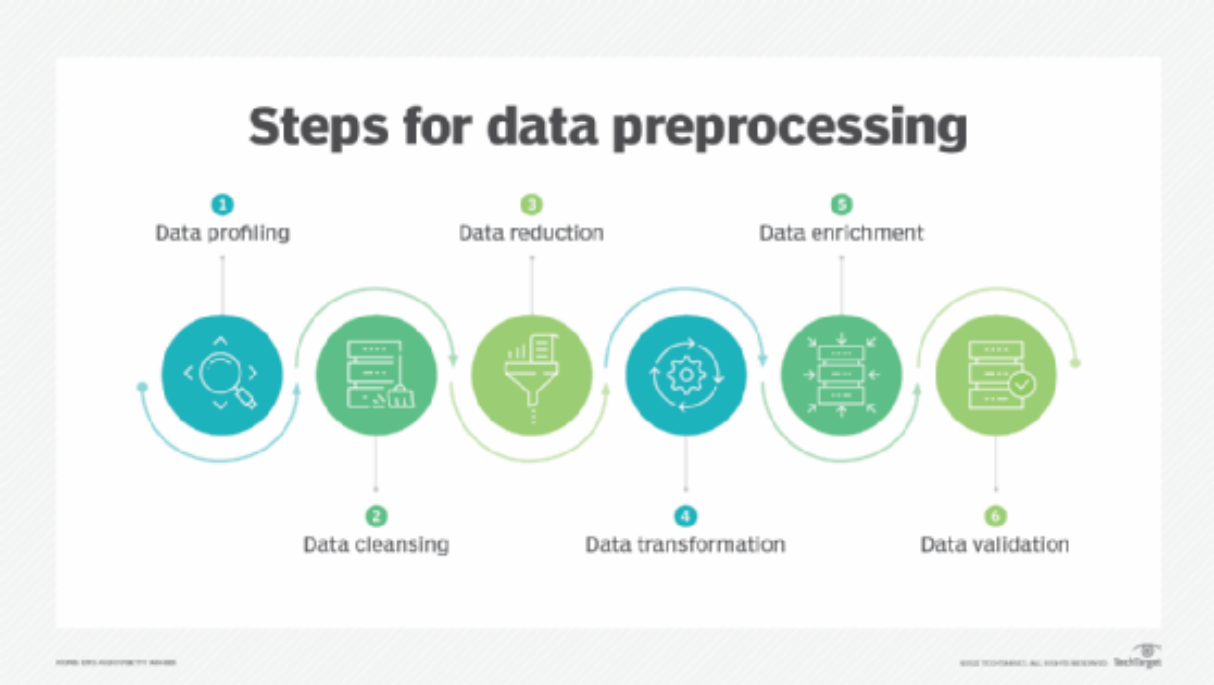


Fig- 4 Data Preprocessing Steps

Uses of Data Preprocessing in Audio Classification:

1. **Improving Model Performance:**
    - Well-preprocessed data can significantly improve the accuracy and robustness of audio classification models. By focusing on relevant features and eliminating noise, the models can learn more effectively.

2. **Reducing Computational Load:**
    - Segmentation and feature extraction can reduce the amount of data that needs to be processed, thereby lowering the computational load and speeding up training and inference times.
3. **Enhancing Generalization:**
    - Normalization and other preprocessing techniques help in making the models more generalizable to new, unseen data by ensuring that the input features have a consistent representation.
4. **Handling Variability in Audio Data:**
    - Preprocessing steps like silence detection and feature extraction help in managing the inherent variability in audio signals, such as differences in volume, background noise, and speaker characteristics.

Model Creation

Model creation in audio classification involves designing and training machine learning models to recognize patterns in audio data. The goal is to build a model that can classify audio into predefined categories, such as speech, music, or environmental sounds.

Key Components of Model Creation:

1. **Neural Network Architecture:**

   Neural network architecture defines the structure of the model, including the arrangement and types of layers. The choice of architecture impacts the model's ability to learn and generalize from audio data.

   - **Common Architectures:**
     - **Convolutional Neural Networks (CNNs):** Effective for analysing spectrograms or other visual representations of audio. CNNs can capture spatial hierarchies and patterns in the frequency domain, making them suitable for tasks like music genre classification and speech recognition.
     - **Artificial Neural Networks (ANNs):** ANNs, inspired by the brain, are machine learning models with interconnected processing units (neurons). These neurons learn by adjusting connections based on data. They are organized in layers: input, hidden (for processing), and output. ANNs are great for pattern recognition in tasks like image or audio classification but can be computationally expensive and difficult to interpret.
     - **Hybrid Models:** Combining CNNs and ANNs to leverage both spatial and temporal features. For example, a model might use CNNs to extract features from spectrograms and ANNs to capture temporal patterns.

2. **Hyperparameter Tuning:**

- **Purpose:** Hyperparameter tuning involves selecting the best set of hyperparameters for the model to optimize its performance. Hyperparameters are parameters that are set before the training process and control aspects like learning rate, batch size, and the number of layers.
- **Key Hyperparameters:**
  - **Learning Rate:** Determines the step size during gradient descent. A well-chosen learning rate helps in converging to the optimal solution efficiently.
  - **Batch Size:** The number of samples processed before the model's parameters are updated. Larger batch sizes can speed up training but may require more memory.

- **Number of Layers and Units:** Specifies the depth and width of the neural network. More layers or units can increase the model's capacity but may also lead to overfitting if not managed properly.
- **Activation Functions:** Functions like ReLU, Sigmoid, or Tanh that introduce non-linearity into the model. The choice of activation function can impact the model's ability to learn complex patterns.
- **Regularization Techniques:** Methods like dropout or L2 regularization that help prevent overfitting by penalizing complex models or randomly dropping units during training.

Uses of Model Creation in Audio Classification:

1. **Classifying Audio Signals:**
   - The primary use of audio classification models is to categorize audio signals into predefined classes. For example, classifying audio clips into different music genres, identifying spoken words, or detecting environmental sounds.
2. **Improving Accuracy:**
   - Well-designed neural network architectures and optimized hyperparameters can enhance the accuracy and robustness of audio classification models. Different architectures and tuning strategies can be explored to achieve better performance on specific tasks.
3. **Handling Complex Audio Data:**
   - Advanced architectures like CNNs and RNNs can handle complex audio data by capturing intricate patterns in both the time and frequency domains. This is crucial for tasks that require detailed analysis, such as speech-to-text conversion or music emotion recognition.
4. **Enhancing Model Generalization:**
   - Proper hyperparameter tuning and architecture design contribute to a model's ability to generalize to new, unseen audio data. This is important for real-world applications where the model encounters diverse audio inputs.
5. **Optimizing Computational Efficiency:**
   - Effective model creation involves balancing model complexity with computational efficiency. Optimized architectures and hyperparameters help in achieving good performance while managing computational resources and training times.

Testing the Artificial Neural Network (ANN) Model

Testing an Artificial Neural Network (ANN) model involves evaluating its performance using various metrics, analysing the confusion matrix, and conducting robustness testing to ensure the model's reliability under different conditions.

Evaluation Metrics:

Evaluating the performance of an ANN model involves using various metrics to measure its accuracy, precision, recall, and other relevant aspects.

1. **Accuracy:**
    - **Definition:** The ratio of correctly predicted instances to the total instances.
    - **Formula:** $\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$
    - **Explanation:** Accuracy measures how often the model makes correct predictions. It is a useful metric when the class distribution is balanced.
2. **Precision:**
    - **Definition:** The ratio of correctly predicted positive observations to the total predicted positives.
    - **Formula:** Precision =TP/(TP+FP)
    - **Explanation:** Precision indicates how many of the predicted positive instances are actually positive. It is important when the cost of false positives is high.
3. **Recall (Sensitivity or True Positive Rate):**
    - **Definition:** The ratio of correctly predicted positive observations to all the actual positives.
    - **Formula:** Recall = TP/(TP+FN)
    - **Explanation:** Recall measures how many actual positive instances were correctly predicted. It is important when the cost of false negatives is high.
4. **F1-Score:**
    - **Definition:** The harmonic mean of precision and recall.
    - **Formula:** F1-Score = 2x (Precision*Recall) / (Precision + Recall)
    - **Explanation:** The F1-score balances precision and recall, providing a single metric when both false positives and false negatives are important.
5. **Other Metrics:**
    - **AUC-ROC (Area Under the Receiver Operating Characteristic Curve):** Measures the model’s ability to distinguish between classes.

- **Log Loss:** Measures the performance of a classification model where the prediction is a probability value between 0 and 1.

Confusion Matrix:

A confusion matrix is a table used to evaluate the performance of a classification model. It shows the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions.

1. **Construction:**
    - Create a 2x2 table (for binary classification) where:
        - Rows represent the actual classes.
        - Columns represent the predicted classes.
2. **Interpretation:**
    - **True Positives (TP):** Correctly predicted positive instances.
    - **True Negatives (TN):** Correctly predicted negative instances.
    - **False Positives (FP):** Incorrectly predicted positive instances.
    - **False Negatives (FN):** Incorrectly predicted negative instances.

The confusion matrix helps in calculating various evaluation metrics:

- **Accuracy:** (TP+TN) / Total Instances
- **Precision:** TP / (TP+FP)
- **Recall:** TP / (TP+FN)
- **F1-Score:** 2 x (Precision x Retail) / (Precision + Retail)

Robustness Testing:

Robustness testing ensures the ANN model performs well under various challenging conditions.

1. **Noisy Data:**
    - **Purpose:** To test how the model handles data with noise or errors.
    - **Technique:** Add random noise to the input data and evaluate the model's performance. This helps in understanding the model's tolerance to data imperfections.
2. **Adversarial Attacks:**
    - **Purpose:** To test the model's resilience against malicious attempts to deceive it.
    - **Technique:** Generate adversarial examples (slightly altered inputs designed to fool the model) and assess the model's ability to correctly classify them.
3. **Data Distribution Shifts:**

- **Purpose:** To evaluate the model's performance when the data distribution changes.
- **Technique:** Train the model on one dataset and test it on a slightly different dataset (e.g., different demographics, seasonal data) to see how well it generalizes.

4. **Cross-Validation:**
   - **Purpose:** To ensure the model performs consistently across different subsets of the data.
   - **Technique:** Use k-fold cross-validation where the data is divided into k subsets, and the model is trained and tested k times, each time with a different subset as the test set.

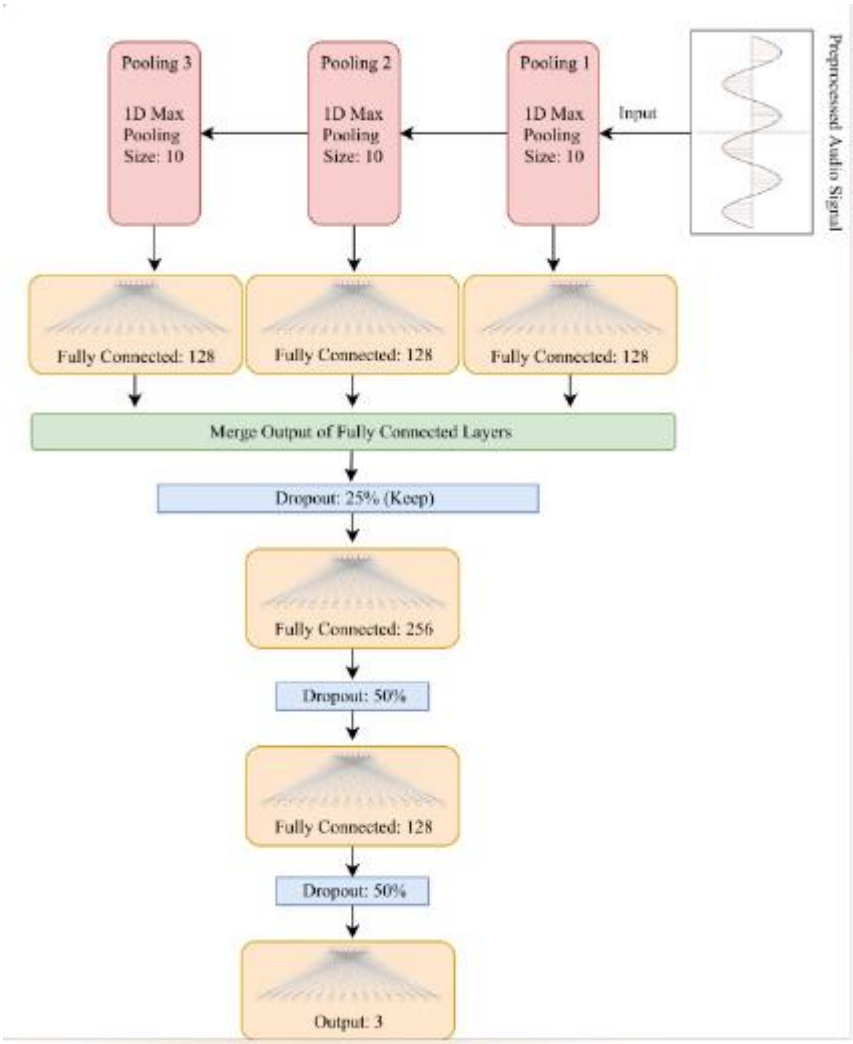


Fig – 5 Architecture of ANN

Testing the Convolution Neural Network (CNN) Model

Testing a Convolutional Neural Network (CNN) model involves evaluating its performance using various metrics, analysing its robustness, and ensuring that it generalizes well under different conditions. Here's a comprehensive guide:

Evaluation Metrics:

1. **Accuracy:**
    - **Definition:** The ratio of correctly predicted instances to the total instances.
    - **Formula:** Accuracy = (TP + TN) / (TP + TN + FP + FN)
    - **Explanation:** Accuracy measures the overall correctness of the model's predictions. It is useful when the classes are balanced.
2. **Precision:**
    - **Definition:** The ratio of correctly predicted positive observations to the total predicted positives.
    - **Formula:** Precision = TP / (TP + FP)
    - **Explanation:** Precision indicates the accuracy of the positive predictions and is important when the cost of false positives is high.
3. **Recall (Sensitivity or True Positive Rate):**
    - **Definition:** The ratio of correctly predicted positive observations to all the actual positives.
    - **Formula:** Recall = TP / (TP + FN)
    - **Explanation:** Recall measures the model's ability to detect all positive instances and is important when the cost of false negatives is high.
4. **F1-Score:**
    - **Definition:** The harmonic mean of precision and recall.
    - **Formula:** F1-Score = 2× (Precision × Recall) / (Precision + Recall)
    - **Explanation:** The F1-score balances precision and recall, providing a single metric that is useful for imbalanced datasets.
5. **AUC-ROC (Area Under the Receiver Operating Characteristic Curve):**
    - **Definition:** Measures the model's ability to distinguish between classes.

- **Explanation:** The AUC-ROC curve plots the true positive rate against the false positive rate at various threshold settings. The area under the curve (AUC) provides a single measure of model performance across all thresholds.

6. **Log Loss:**
    - **Definition:** Measures the performance of a classification model where the prediction is a probability value between 0 and 1.
    - **Explanation:** Log loss penalizes incorrect classifications based on the predicted probability. It is particularly useful for evaluating probabilistic predictions.

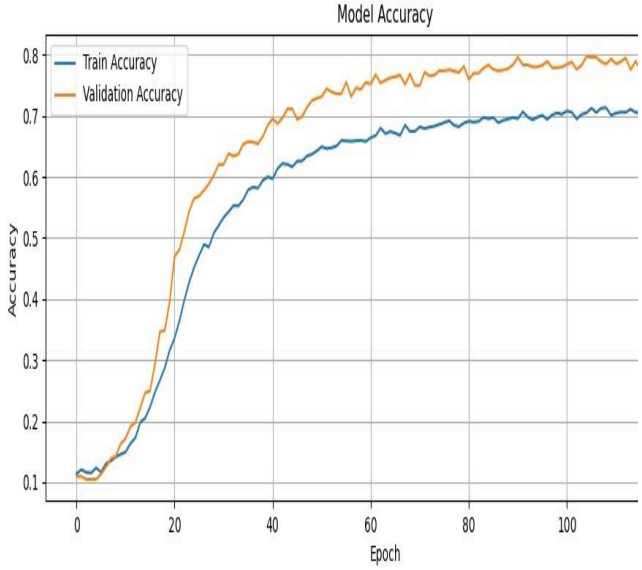
Robustness Testing:

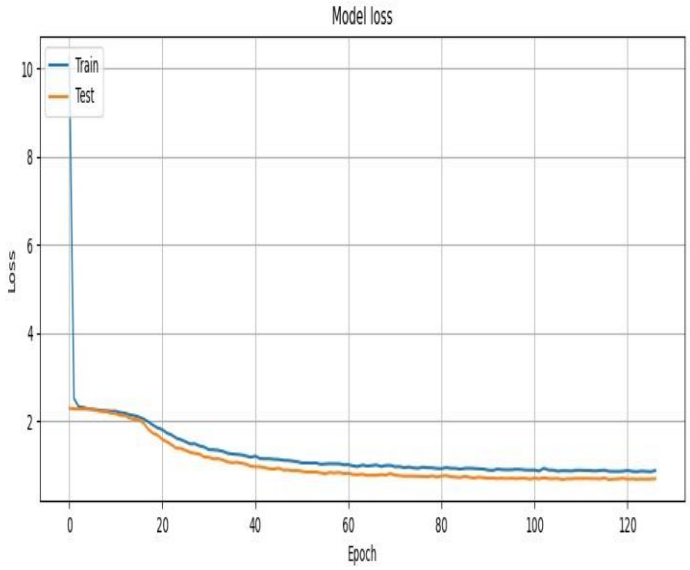
1. **Noisy Data:**
    - **Purpose:** To evaluate how the model performs when the input data contains noise or errors.
    - **Technique:** Add random noise to the input data and assess the model's performance. This helps determine the model's tolerance to data imperfections and noise.
2. **Adversarial Attacks:**
    - **Purpose:** To test the model's resilience against intentional attempts to deceive it.
    - **Technique:** Generate adversarial examples by slightly altering the input data to mislead the model and evaluate its robustness to these perturbations. Common methods include the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD).
3. **Data Distribution Shifts:**
    - **Purpose:** To examine the model's performance when the data distribution changes.
    - **Technique:** Train the model on one dataset and test it on a slightly different dataset (e.g., different demographics, seasonal variations) to assess its generalization. This can include testing on different datasets, synthetic data, or real-world data variations.
4. **Cross-Validation:**
    - **Purpose:** To ensure consistent performance across different subsets of the data.
    - **Technique:** Use k-fold cross-validation, where the data is divided into k subsets, and the model is trained and tested k times, each time with a different subset as the test set. This helps in assessing the model's robustness to variations in the training data.

Conclusion

ANN Model Evaluation Metrices

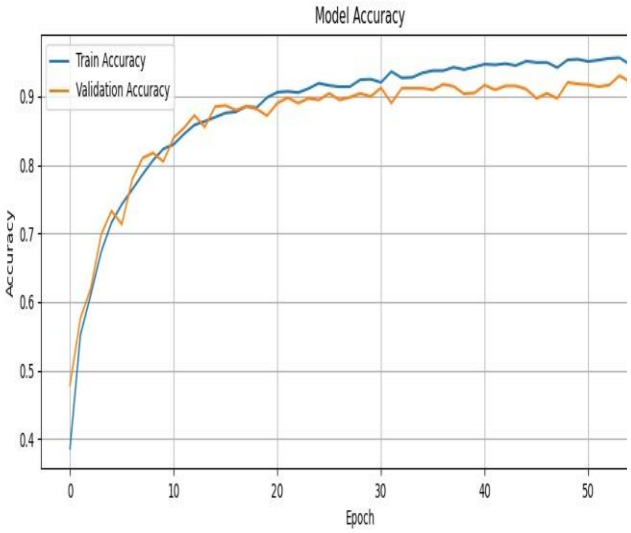
Achieved 80% Accuracy

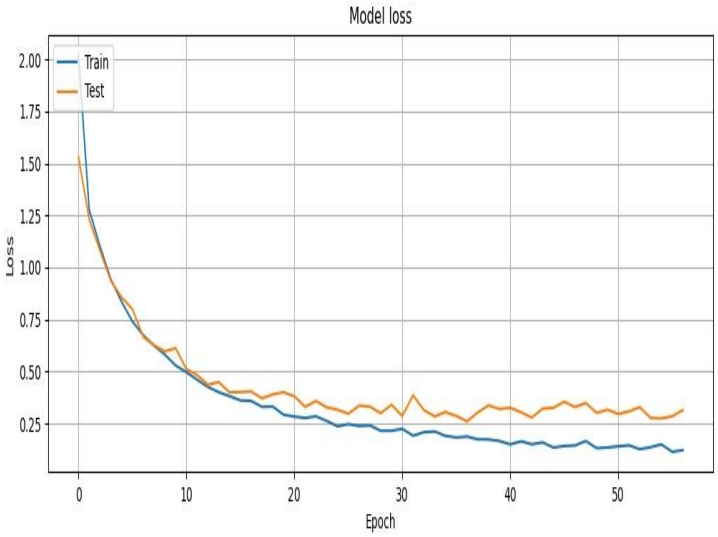




CNN Model Evaluation Metrices

Achieved 92% Accuracy





Future Scope

The future of audio classification using deep learning is brimming with exciting possibilities:

1. **Increased Accuracy and Generalizability:** Advancements in deep learning architectures and training techniques will lead to even more accurate audio classification models. These models will be better equipped to handle complex and noisy environments.
2. **Focus on Explainability and Trustworthiness:** As deep learning models become more intricate, there will be a growing emphasis on interpretability. This will allow us to understand how models arrive at decisions, fostering trust and reliability in applications.
3. **Emerging Applications:** Deep learning will revolutionize various fields:
    - **Environmental monitoring:** Classifying animal sounds for biodiversity studies or detecting environmental threats like deforestation.
    - **Healthcare:** Analysing medical audio data (heartbeats, lungs) for early disease detection.
    - **Enhanced Human-Computer Interaction:** Voice assistants with superior speech recognition and background noise cancellation.
4. **Edge Computing and Resource Efficiency:** Deploying deep learning models on resource-constrained devices at the edge of networks will enable real-time audio processing without relying on the cloud.
5. **Audio Event Detection and Segmentation:** Going beyond classification, future models will be able to pinpoint and categorize specific sound events within audio streams, like identifying a car horn in a traffic recording.

Overall, deep learning will continue to push the boundaries of audio classification, leading to more accurate, adaptable, and impactful applications across various domains.

References

1. Piczak, K. J. (2015). "Environmental Sound Classification with Convolutional Neural Networks." *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1-6.

2. Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R. J., Wilson, K. (2017). "CNN Architectures for Large-Scale Audio Classification." *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 131-135.

3. Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2017). "Convolutional Recurrent Neural Networks for Music Classification." *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2392-2396.

4. Schlüter, J., & Grill, T. (2015). "Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks." *16th International Society for Music Information Retrieval Conference (ISMIR)*, 121-126.

5. Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., & Ritter, M. (2017). "Audio Set: An Ontology and Human-Labeled Dataset for Audio Events." *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 776-780.

6. Goodfellow, I., Bengio, Y., & Courville, A. (2016). "Deep Learning." *MIT Press.*

7. **UrbanSound8K:** "A dataset containing 8,732 labeled sound excerpts (<=4s) of urban sounds from 10 classes." UrbanSound8K