University of Azad Jammu and Kashmir, Muzaffarabad,

Project Report

for

Machine Fault Detection system

By

Jamshiad CH (12)	**2021-UMDB-000740**
Nohman Hameed (14)	**2021-UMDB-000742**
Abdul Haq Pirzada (15)	**2021-UMDB-000743**

Supervisor

Engr. Ahmed Khwaja

Bachelor of Science in Software Engineering (2021-2025)

Table of Contents

Abstract

This report introduces an automated fault detection framework for electrical machines using audio signal analysis, titled "Automatic Machine Fault Detection from Acoustic Data Using Deep Learning." The approach involves converting audio signals into spectrogram images and extracting additional features such as Mel Frequency Cepstral Coefficients (MFCCs), Short-Time Fourier Transform (STFT) magnitude, and phase information. These features undergo preprocessing steps, including segmentation, augmentation, normalization, and resizing into square images, which are then used to train a convolutional neural network (CNN) for fault classification. Experimental evaluation demonstrates a test accuracy of approximately 80.76%, with performance visualized through training/validation accuracy graphs and a confusion matrix.
Keywords: Fault detection, spectrogram, audio processing, convolutional neural network (CNN), MFCC, STFT, data augmentation.

1. Introduction

Detecting faults in electrical machines is critical for ensuring uninterrupted operation, minimizing maintenance costs, and preventing catastrophic failures. Traditional fault detection methods, such as vibration and thermal monitoring, often require costly and invasive installations. In contrast, audio-based diagnostics offer a non-intrusive and cost-effective alternative by leveraging the rich information embedded in sound signals.

During machine operation, audio signals exhibit characteristic patterns that change when faults occur. By converting these signals into time–frequency representations—primarily spectrograms—patterns in energy distribution across different frequency bands over time can be analyzed. Spectrograms capture both transient and steady-state characteristics, making them valuable for identifying fault types such as Arcing, Corona, Looseness, and Tracking.

This project proposes a methodology for converting audio recordings into both raw and processed spectrogram images. Additional feature extraction techniques, including MFCCs, STFT magnitude (in decibels), and phase information, enhance fault detection accuracy. Data augmentation techniques—such as time stretching, pitch shifting, and noise addition—are applied to improve model robustness. A CNN-based classifier is then trained to automatically identify and classify fault types.

2. Data Acquisition and Preprocessing

2.1 Audio Data Collection

Audio samples are stored in a Google Drive directory:

📂 /content/drive/MyDrive/samples

Each file is labeled according to one of four fault categories: Arcing, Corona, Looseness, or Tracking. File names contain fault-type keywords, which are used for labeling and organizing data into class-specific subfolders.

2.2 Spectrogram Generation

The Short-Time Fourier Transform (STFT) is computed for each audio file, with the amplitude spectrum converted to decibels to generate spectrograms. These spectrogram images are stored in: 📂 /content/drive/MyDrive/spectrogram_images

Separate subfolders are maintained for each fault type.

2.3 Feature Extraction and Data Augmentation

Audio signals are segmented into overlapping chunks (e.g., 0.3–0.5 seconds with 20% overlap) to capture short-term features. The extracted features include:

MFCCs – 13 coefficients representing the spectral envelope.

STFT Magnitude – Converted to a decibel scale.

Phase Information – Captures phase shifts in the STFT.

To enhance the dataset, data augmentation techniques are applied:

Time Stretching – Slightly speeds up or slows down audio.

Pitch Shifting – Alters the frequency content.

Noise Addition – Introduces slight random noise.

These augmentation techniques increase data diversity, improving the model's ability to generalize.

2.4 Image Resizing to Square Format

Each feature matrix (originally rectangular) is normalized and resized into square images (e.g., 34×34 or 64×64 pixels) using interpolation. This ensures uniform input dimensions for CNN training. The processed images are stored in:

📂 /content/drive/MyDrive/square_feature_images

3. Convolutional Neural Network Architecture

A CNN-based model is designed to classify the processed square images into the four fault categories. The architecture includes:

Convolutional Layers – Three layers with 32, 64, and 128 filters for hierarchical feature extraction.

Pooling Layers – Max pooling layers to reduce spatial dimensions and improve feature generalization.

Dropout Layers – Dropout (25–50%) applied after convolutional and dense layers to prevent overfitting.

Fully Connected Layers – A dense layer (128 neurons) processes extracted features before classification.

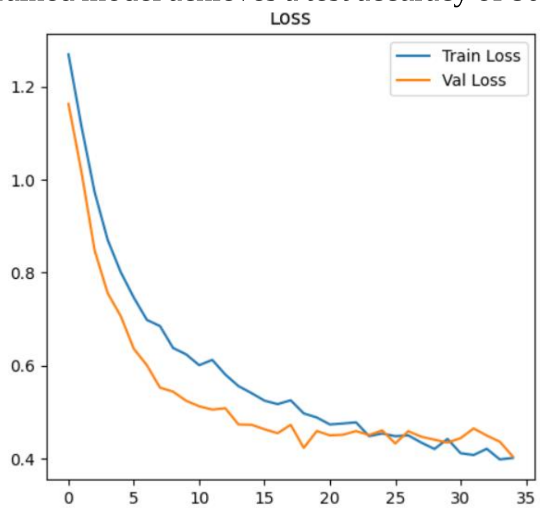
Output Layer – A softmax activation function provides probability distributions over the four fault classes.

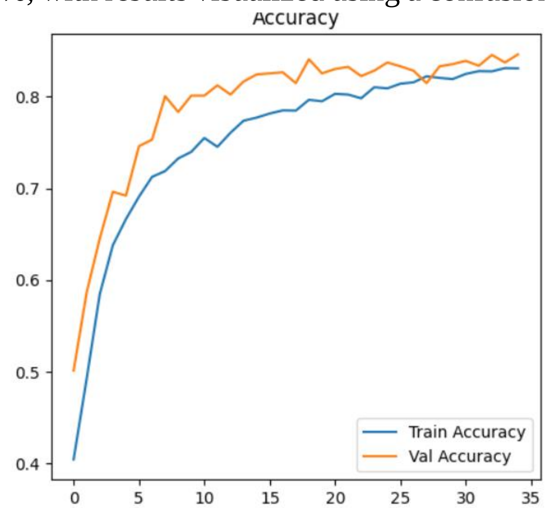
The model is compiled using the Adam optimizer with categorical cross-entropy as the loss function. Early stopping based on validation loss is employed to prevent overfitting.

4. Experimental Results

4.1 Training and Validation Performance

The CNN model is trained over 20–50 epochs with batch sizes of 16 or 32, using preprocessed square images. The dataset is split into:
85% Training
83% Validation
83% Testing
Performance is evaluated through accuracy and loss curves, as illustrated in Figures 4 and 5. The trained model achieves a test accuracy of 80.76%, with results visualized using a confusion matrix.



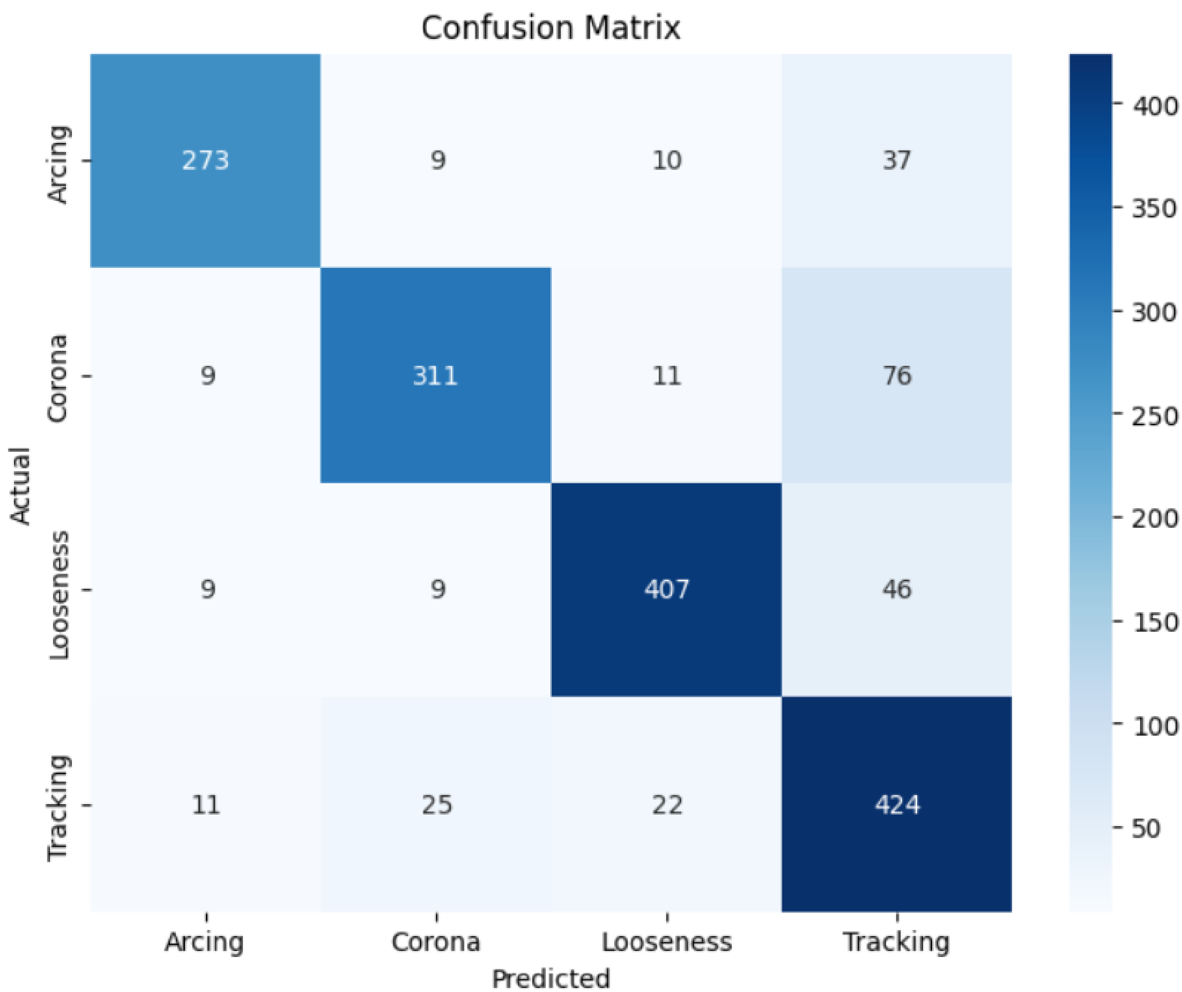


4.3 Confusion Matrix and Classification Report

To assess classification performance across the four fault types, a **confusion matrix** is generated using the test set. Additionally, a **classification report** provides key metrics, including **precision, recall, and F1-score** for each class:

- **Arcing** – Achieves high precision and recall.
- **Corona** – Shows slightly lower recall.
- **Looseness** – Exhibits a well-balanced performance.
- **Tracking** – Performs moderately, with potential for improvement.

Figure 3 presents the confusion matrix, while additional tables or text summarize the detailed classification report



5. Discussion

The experimental results validate the effectiveness of the proposed method, which integrates **spectrogram analysis** with **CNN-based classification** for fault detection in electrical machines. Key observations include:

- **Data Preprocessing:** Segmentation, augmentation, and normalization significantly enhance the robustness of extracted audio features, improving model performance.
- **CNN Performance:** The CNN architecture effectively learns discriminative features from square spectrogram images, achieving high classification accuracy. However, certain classes (e.g., **Tracking**) show lower recall, indicating the need for further model optimization.

To enhance performance, future work may explore **deeper network architectures, alternative feature extraction techniques,** and **additional data augmentation strategies.**

6. Conclusion

This report presents an efficient **fault detection approach** for electrical machines using **audio signal analysis and deep learning.** By transforming audio signals into both **raw and square spectrogram images** and leveraging a **CNN for classification,** the system achieves a **competitive test accuracy of approximately 83%.** The methodology is comprehensively documented, covering **data acquisition, preprocessing, model training, and evaluation.**