💡 Project Report: Bearing Health Detection Using AI

👨‍🔬 Objective

To develop a robust AI-based system for classifying the health of bearings using vibration signal data. The system automatically detects the condition of bearings and classifies them into one of four categories:

- **Normal**
- **Inner Race Fault (IR)**
- **Outer Race Fault (OR)**
- **Ball Fault**

📂 Dataset Overview

- **Source:** `.mat` files containing time-series vibration data from different bearing conditions
- **Signal Key:** `DE_time`
- **Sampling Rate:** 12,000 Hz
- **Segment Duration:** 5 seconds
- **Augmented Dataset Size:** 4× original segments

🔧 Preprocessing Pipeline

1. **Segmentation:**
    - Raw vibration signal is divided into 5-second segments
    - Ensures uniform input duration for all samples
2. **Spectrogram Conversion:**
    - Each segment is converted to a time-frequency spectrogram
    - Output shape standardized to **(128, 128)** for CNN input
3. **Label Assignment:**
    - Labels are inferred from filenames: `Normal`, `IR`, `OR`, `B0`
    - Mapped to class indices: 0 → Normal, 1 → IR, 2 → OR, 3 → Ball Fault

---

🔄 Data Augmentation

Augmentation	Applied On	Purpose
Gaussian Noise	Raw signal	Simulates sensor/environment noise

Amplitude Scaling	Raw signal	Mimics varying signal intensities
Frequency Masking	Spectrogram	Simulates frequency dropout/occlusion

Each original segment generates:

- 1 Original spectrogram
- 1 Noised version
- 1 Amplitude-scaled version
- 1 Frequency-masked version

➡️ Dataset size increases **4x**, improving generalization.

🧐 LLM-Based Fault Reasoning System

To enhance explainability, a **Large Language Model (LLM)** is used alongside the CNN. The LLM takes **engineered features** from the raw signal and simulates how a human expert would reason about the fault.

🔢 Input Features to LLM:

- **RMS** (Root Mean Square) → Vibration energy
- **Kurtosis** → Impulsiveness of the signal
- **Skewness** → Asymmetry of the waveform
- **Peak-to-Peak Amplitude**
- **Crest Factor** → Ratio of peak to RMS
- **Dominant Frequency** → Main frequency component

⚖️ LLM Inference Flow:

1. A segment's statistical features are extracted
2. A formatted prompt is created describing the features
3. The LLM (e.g., `deepseek-llama` via Groq API) returns the predicted fault type as text

📊 Why LLM?

- Offers **interpretable reasoning**
- Acts as a **sanity check** for CNN predictions
- Helps in cases where CNN and LLM disagree, offering deeper insight

---

📊 CNN Model Architecture

Layer	Description

Input	(128, 128, 1) grayscale spectrogram
Conv2D × 3	32, 64, 128 filters with ReLU
BatchNorm + MaxPool	Normalize and reduce spatial dims
Flatten	Converts to 1D
Dense (128)	Fully connected with ReLU
Dropout (0.3)	Prevents overfitting
Output Dense (4)	Softmax over 4 classes

- **Optimizer:** Adam (lr = 0.0005)
- **Loss Function:** Sparse Categorical Crossentropy
- **Balanced Class Weights** used

⏱️ Model Training

- **Train/Validation/Test:** 64% / 16% / 20%
- **Epochs:** 35
- **Batch Size:** 32

📈 Training Trends:

- Accuracy improves steadily
- Loss reduces consistently
- Model shows strong generalization

📊 Evaluation Results

- **Test Accuracy:** ~96%
- **Precision & Recall:** High across all classes
- **Confusion Matrix:** Small confusion between IR and OR

🔢 Classification Report:

	precision	recall	f1-score
Normal	0.98	0.97	0.98
IR	0.94	0.96	0.95
OR	0.95	0.94	0.94
Ball	0.97	0.96	0.96

💻 UI for Inference

- Developed using **ipywidgets** for `.mat` file input
- Visualizes:
  - Time-domain signal
  - CNN prediction with confidence
  - LLM prediction with reasoning
  - Extracted signal features
- Highlights agreement or disagreement between models

✅ Conclusion

This project builds a complete AI-based system for automatic bearing health detection using:

- CNN trained on spectrograms
- Feature-based LLM reasoning
- Signal-domain and frequency-domain augmentations
- Interactive UI for real-time testing
