AI-Based Approaches for Bearing Health Classification Using the CWRU Dataset

Objective

This report presents a simplified review of five high-performing research approaches that use artificial intelligence (AI) to classify bearing health using the Case Western Reserve University (CWRU) dataset. Each method is summarized with a focus on the type of input data, model architecture, preprocessing, and overall performance.

Overview of the CWRU Dataset

The **Case Western Reserve University (CWRU) bearing dataset** is one of the most widely used benchmark datasets for machine condition monitoring, especially bearing fault diagnosis. It was created by the Bearing Data Center at CWRU and is freely available to the research community.

Key Features:

- **Data Type:** Vibration signals recorded from accelerometers mounted on the drive-end and fan-end of a motor.
- **Sampling Frequency:** Commonly 12,000 Hz and 48,000 Hz.
- **Operating Conditions:** Different motor loads and speeds were tested.
- **Fault Types:**
    - Inner Race Faults
    - Outer Race Faults
    - Ball Faults
    - Normal (Healthy bearings)
- **Fault Sizes:** Faults were artificially introduced in different sizes (0.007", 0.014", 0.021") using electro-discharge machining.

Why It Is Used:

- Standardized and well-documented
- Covers a range of real-world bearing faults
- Ideal for testing both signal processing and machine learning models

Typical Preprocessing Steps:

- **Segmentation:** Signals are split into fixed-size time windows (e.g., 5 seconds)

- **Transformation:** Raw signals are converted into features or images (e.g., FFT, STFT, or spectrogram)
- **Normalization:** Values are scaled for better learning performance

This dataset has become a key foundation for comparing different fault diagnosis models under consistent conditions.

---

1. FaultNet: Deep CNN Framework for Fault Diagnosis

Paper Title: *FaultNet: A Novel Deep CNN Architecture for Fault Diagnosis Using CWRU Dataset*

- **How it Works:** FaultNet is a Convolutional Neural Network (CNN) that not only processes vibration signals as images but also incorporates statistical summaries such as the average and median of the signal. These added features help the model better understand the input.
- **Input Type:** Vibration signals converted into grayscale images.
- **Preprocessing:** The signals are segmented and reshaped into images.
- **Performance:** Achieves extremely high accuracy (~99.98%) on the CWRU dataset.
- **Key Idea:** Enhancing the input with simple statistical values improves the model's performance and generalization.

---

2. End-to-End Convolutional Recurrent Neural Network (CRNN)

Paper Title: *An End-to-End Deep Learning Approach for Bearing Fault Diagnosis with CNN and LSTM*

- **How it Works:** This model combines CNNs (good with image-like data) and LSTMs (good at learning from sequences). Together, they learn both how the vibration looks and how it changes over time.
- **Input Type:** Raw vibration signals directly fed into the model.
- **Preprocessing:** No image conversion or manual feature extraction required.
- **Performance:** Reaches about 99.89% accuracy.
- **Key Idea:** Learns everything directly from the raw signal data.

---

3. Vision Transformer-Based Model for Bearing Fault Diagnosis

Paper Title: *Bearing Fault Diagnosis Using Vision Transformer Applied on Time-Frequency Spectrograms*

- **How it Works:** This model uses a Vision Transformer (ViT), originally developed for image recognition, to analyze spectrograms (image-like representations of sound). It captures important patterns in vibration signals through attention mechanisms.

- **Input Type:** Spectrogram images derived from vibration signals.
- **Preprocessing:** Short-Time Fourier Transform (STFT) to convert signals into images.
- **Performance:** Achieves about 98.8% accuracy on CWRU with multiple fault types.
- **Key Idea:** ViT analyzes the whole image at once, allowing it to identify complex fault characteristics.

---

4. Hybrid Deep Learning Model Based on CNN, LSTM, and GRU

Paper Title: *A Hybrid CNN-LSTM-GRU Architecture for Fault Classification in Bearings*

- **How it Works:** This hybrid model combines CNN for spatial features with LSTM and GRU for learning time-based signal characteristics. This makes it powerful in learning different types of features.
- **Input Type:** Raw vibration signals.
- **Preprocessing:** Signals are divided into equal-length segments.
- **Performance:** Around 99.3% accuracy on the CWRU dataset.
- **Key Idea:** Combining multiple deep learning methods increases feature diversity and classification strength.

---

5. DRTCNN: Deep Transfer Convolutional Neural Network for Cross-Domain Learning

Paper Title: *DRTCNN: A Deep Transfer CNN with Domain Adaptation for Cross-Machine Fault Diagnosis*

- **How it Works:** This model uses a CNN with a special technique called domain adaptation. This helps the model perform well even when the testing data comes from different environments or machines.
- **Input Type:** Raw vibration signals collected under various conditions.
- **Preprocessing:** Applies domain adaptation and transfer learning to adjust for differences in datasets.
- **Performance:** About 98.5% accuracy across both CWRU and external datasets.
- **Key Idea:** Makes the model robust enough to handle data differences between machines or settings.

1. Preprocessing

Before feeding the data into any machine learning model, the raw sensor readings from the bearings need to be properly handled. These vibration readings are collected from machinery using accelerometers and are stored in `.mat` files.

To make the data suitable for analysis:

- **Segmentation**: Each long vibration signal is divided into smaller 5-second segments. This makes processing easier and ensures uniform input size.
- **Sampling**: The sampling rate is **12,000 Hz**, meaning each 5-second segment contains **60,000 readings**.
- **Spectrogram Conversion**: Instead of using raw numbers, we convert each segment into a **spectrogram** — a visual representation of how frequencies vary over time. It resembles an image and helps in identifying patterns related to bearing faults.

This conversion is done using the **Short-Time Fourier Transform (STFT)**, which breaks the signal into overlapping windows and computes the frequency spectrum for each:

The output spectrogram is resized to **128 × 128**, and treated as a grayscale image with shape **(128, 128, 1)**.

---

2. Augmentation Techniques

Data augmentation is used to artificially expand the training data and improve the model's ability to generalize to new, unseen examples. Here are the two main techniques we used:

- **Gaussian Noise**: Random noise is added to the signal. This simulates minor variations or disturbances that may occur in real-world sensor data.
  where is noise with mean 0 and standard deviation
- **Time Shifting (Rolling)**: The signal is shifted by a random number of data points: where is the shift amount, and is the length of the signal segment (**60,000**).

Each original segment generates 2 additional augmented versions, increasing the dataset **3-fold** while preserving the original label.

---

3. CNN Model: Layer by Layer

The core of the classification system is a **Convolutional Neural Network (CNN)**, which is well-suited to image-like data — in our case, spectrograms.

Layer Architecture and Dimensions

Layer Type	Output Shape	Explanation
Input	(128, 128, 1)	2D grayscale image (spectrogram)
Conv2D (32 filters)	(128, 128, 32)	3x3 kernel, padding='same'
MaxPooling2D	(64, 64, 32)	Reduces spatial size by half

Conv2D (64 filters)	(64, 64, 64)	Learns deeper image features
MaxPooling2D	(32, 32, 64)	Further spatial reduction
Conv2D (128 filters)	(32, 32, 128)	Deeper feature map for abstract representation
MaxPooling2D	(16, 16, 128)	Final pooled feature map
Flatten	(32768,)	Converts 3D tensor into 1D array for Dense layer
Dense (128 units)	(128,)	Fully connected neurons to learn final representations
Dropout (0.5)	(128,)	Turns off 50% neurons randomly (for regularization)
Output Dense (4)	(4,)	Final 4-class output (Normal, IR, OR, Ball) using Softmax

---

Softmax Output:

Each class score is converted to a probability using:

This ensures all class probabilities add up to 1.

Loss Function:

We use **Sparse Categorical Crossentropy**:

This penalizes the model when it assigns low probability to the correct class.