# Baden-Wuerttemberg Cooperative State ... metin, yazı tipi, ekran görüntüsü, grafik içeren bir resim Açıklama otomatik olarak oluşturuldu

# Internship Project Report

Osman Kantarcıoğlu

1.07.2024-31.08.2024

AI Application to Material’s Testing Using Acoustic Emission and Analysis of AE Spectra

## Steps

- Merging and Labelling Datasets

- Cleaning and Preprocessing Data

- Conducting Exploratory Data Analysis (EDA)

- Performing Feature Engineering

- Training Artificial Neural Networks (ANNs)

- Evaluating Model Performance

- Addressing Data Imbalance

- Drawing Final Conclusions and Recommendations

## Merging and Labelling Dataset

During the project, I worked with three distinct datasets stored in separate folders. The first step was to merge these datasets into a unified dataset. During the merging process, I labeled the data to classify wear conditions into categories based on the values of the flute 1, flute 2, and flute 3 features. Below is the logic used for labeling:

if (df2["flute\_1"].iloc[idx] >= 166) or (df2["flute\_2"].iloc[idx] >= 166) or (df2["flute\_3"].iloc[idx] >= 166):  
 status = "severe wear"  
elif (df2["flute\_1"].iloc[idx] > 66) or (df2["flute\_2"].iloc[idx] > 66) or (df2["flute\_3"].iloc[idx] > 66):  
 status = "uniform wear"  
else:  
 status = "rapid wear"

The labeled data allowed for the classification of wear conditions into three groups: Severe Wear, Uniform Wear, and Rapid Wear. This step was essential for enabling further analysis.

## Analysis of Acoustic Emission

diyagram, ekran görüntüsü, dikdörtgen, metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

This graph illustrates the relationship between wear status and acoustic emission. The following observations can be made:

- In cases of severe wear, the acoustic emission values are significantly higher, indicating increased stress or wear intensity.

- For uniform wear and rapid wear conditions, the acoustic emission values are comparatively lower and less variable.

These patterns indicate that acoustic emission metrics can be used effectively to distinguish between different wear conditions.

## Artificial Neural Network Training

I trained artificial neural networks (ANNs) using the labeled dataset. To able to handle imbalance dataset problem I used oversampling.

Before:

metin, ekran görüntüsü, çizgi, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

After:

metin, ekran görüntüsü, çizgi, dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturuldu

Below is the classification performance summary:

Class 2 (Severe Wear): High precision (0.86), recall (0.93), and F1-score (0.89).

Class 0 (Rapid Wear): Moderate performance with a recall of 0.65 and precision of 0.78.

Class 1 (Uniform Wear): Balanced performance with a recall of 0.74 and precision of 0.67.

Overall accuracy: 77%. While the model is effective for detecting severe wear, it requires improvements for identifying rapid and uniform wear conditions.

metin, ekran görüntüsü, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

## Addressing Dataset Imbalance

The dataset was imbalanced, which affected the model's generalization ability. Oversampling was applied to balance the dataset. The classification report after addressing imbalance is shown below:

metin, ekran görüntüsü, dikdörtgen, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

precision recall f1-score support  
 No 0.94 0.91 0.93 25625  
 Yes 0.91 0.94 0.93 25686  
 accuracy 0.93 51311  
 macro avg 0.93 0.93 0.93 51311  
weighted avg 0.93 0.93 0.93 51311

Balancing the dataset significantly improved classification performance across all categories, with an overall accuracy of 93%.

# Detailed Report on Fourier Transformation

**Our primary objective is to apply Fast Fourier Transform (FFT) to the data and generate plots, which will serve as inputs for the subsequent stages of our analysis.**

## Introduction

This report provides a detailed explanation of the code that performs Fourier transformation on acoustic emission data collected during material wear testing. The code processes data files, categorizes them based on wear conditions, and saves plots of the Fourier-transformed data for analysis. The primary goal is to analyze acoustic emission signals and correlate them with wear conditions.

## File Path Initialization

The function `fourier\_transformation` begins by setting up the necessary file paths for processing and saving data. These paths include:

- `file\_path`: The directory containing raw data files for a specific wear condition.  
- `file\_path2`: A CSV file that holds wear information related to the corresponding raw data.  
- `save\_path`: The output directory where processed results and plots will be saved.

## Directory Creation for Output

To organize the results effectively, the code creates subdirectories under the `save\_path` for each wear condition: `severe wear`, `uniform wear`, and `rapid wear`. This ensures that processed plots are saved in directories corresponding to their respective wear categories. The `os.makedirs` function is used to create directories if they do not already exist.

## Processing Each Data File

The code iterates through each file in the raw data directory (`file\_path`) and processes them one by one. For each file, the following steps are executed:

**1. Sampling and Column Renaming**

A random 1% sample of the data is taken using `df.sample(frac=0.01)` to reduce the computational load while retaining statistical significance. The sampled data's columns are renamed to descriptive names, such as 'Acceleration in x axis', 'x vibration', and 'acoustic emission'.

**2. Wear Status Categorization**

The wear condition for each file is determined based on values in the `flute\_1`, `flute\_2`, and `flute\_3` columns from the wear information CSV (`file\_path2`). The logic for categorization is:

- `Severe Wear`: Any flute value >= 166.  
- `Uniform Wear`: Any flute value > 66 and < 166.  
- `Rapid Wear`: All other cases.

**3. Zero-Centering and Fourier Transformation**

The acoustic emission data is zero-centered by subtracting its mean to remove the DC component. A Fourier transformation is then performed on the zero-centered data using the Fast Fourier Transform (FFT) algorithm. This transforms the time-domain data into the frequency domain, revealing the signal's frequency components.

Key parameters used in the FFT calculation:  
- `n`: Length of the acoustic emission data.  
- `sample\_spacing`: The interval between samples, set to 1/50,000 seconds.

**4. Plotting Fourier Transformation Results**

For each file, the magnitude of the Fourier-transformed data is plotted against frequency. The graph provides insights into the frequency components of the acoustic emission signal. The plot is titled with the file number and corresponding wear status for easy identification.

**5. Saving Results**

The generated plot is saved as a PNG file in the subdirectory corresponding to the wear status (`severe wear`, `uniform wear`, or `rapid wear`). The naming convention used for the files includes the file number and wear status, ensuring clear organization.

## Conclusion

This code systematically processes acoustic emission data to analyze frequency characteristics of different wear conditions. By leveraging Fourier transformation, it provides valuable insights into the relationship between acoustic signals and material wear. The well-organized output structure ensures efficient data management, making it easier for researchers to interpret results and draw conclusions.

Sample Plots:

Rapid Wear Before Fourier Transformation:

öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram, metin, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Rapid Wear After Fourier Transformation:  
metin, ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Uniform Wear Before Fourier Transformation:

metin, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Uniform Wear After Fourier Transformation:metin, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, ekran görüntüsü, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Severe Wear Before Fourier Transformation:

öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram, metin, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Severe Wear After Fourier Transformation:

metin, ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

# Image Classification Using Deep Learning

## Introduction

This section describes the implementation of a deep learning model for image classification to categorize wear patterns. The approach involves preprocessing images, training a Convolutional Neural Network (CNN), and evaluating its performance. Additionally, predictions are made on new images based on the trained model.

## Loading and Preprocessing Data

The code begins by defining the paths to the data folders containing images of different wear patterns. Images are loaded from these folders using the `load\_images\_from\_folder` function, which performs the following steps:

1. Iterates over all files in the folder.  
2. Loads each image, resizes it to 128x128 pixels, and normalizes the pixel values.  
3. Assigns labels based on the folder name.

Once images are loaded, the data from multiple folders is concatenated to form a comprehensive dataset. Labels are encoded using `LabelEncoder` and converted to categorical format for compatibility with the CNN.

## Train-Test Split

The dataset is split into training and testing subsets using an 80-20 split. This ensures that the model is evaluated on unseen data after training, allowing for a fair assessment of its performance.

## Model Architecture

The deep learning model is implemented using Keras. It consists of the following layers:

- Three convolutional layers with increasing filter sizes (32, 64, and 128), each followed by max-pooling layers.  
- A flatten layer to convert the 2D feature maps into a 1D vector.  
- A dense layer with 512 units and ReLU activation.  
- A dropout layer with a rate of 0.5 to prevent overfitting.  
- A final dense layer with a softmax activation function for multi-class classification.

## Training the Model

The model is compiled using the Adam optimizer, categorical cross-entropy loss, and accuracy as the evaluation metric. The training process involves 50 epochs, during which the model learns patterns in the data and minimizes the loss. Validation data is used during training to monitor performance and prevent overfitting.

## Model Evaluation

After training, the model is evaluated on the test set. The results include metrics such as loss and accuracy, which provide an overview of the model's performance. A classification report is also generated to analyze precision, recall, and F1-score for each class.

## Prediction on New Images

A prediction function is implemented to classify new images using the trained model. The function performs the following steps:

1. Loads and preprocesses the input image.  
2. Passes the image through the model to obtain predictions.  
3. Determines the class label with the highest probability and returns a human-readable output.

## Conclusion

This implementation demonstrates the use of deep learning techniques for image classification. The model successfully categorizes wear patterns and achieves high accuracy on the test set. The prediction function further enables the application of the trained model to new data, showcasing its practical utility.

## Classification Report

The classification report provides detailed metrics for the performance of the trained model on the test set. The report includes precision, recall, F1-score, and support for each class, as well as overall accuracy and macro averages.

precision recall f1-score support  
 0 0.95 1.00 0.98 21  
 1 0.76 0.80 0.78 20  
 2 0.75 0.67 0.71 18  
  
 accuracy 0.83 59  
 macro avg 0.82 0.82 0.82 59  
weighted avg 0.83 0.83 0.83 59

The classification report reveals the following insights:  
- Class 0 demonstrates excellent performance with a precision of 0.95, recall of 1.00, and an F1-score of 0.98, indicating that the model performs exceptionally well in identifying this class.  
- Class 1 has a moderate performance with a precision of 0.76 and a recall of 0.80. This suggests that while the model identifies most instances, there is room for improvement in reducing false positives.  
- Class 2 shows the lowest performance, with a precision of 0.75 and a recall of 0.67. This indicates that the model struggles to correctly identify instances of this class, leading to a lower F1-score of 0.71.  
- The overall accuracy of the model is 83%, and the macro average metrics indicate a balanced performance across classes, with an F1-score of 0.82.

# Comparison Between ANN and CNN Models

This section compares the performance of the Artificial Neural Network (ANN) constructed earlier in the project with the Convolutional Neural Network (CNN) used for image classification. The comparison evaluates metrics such as accuracy, precision, recall, F1-score, and the overall ability to generalize to unseen data.

## Performance of ANN

The ANN model trained on labeled acoustic emission data achieved an overall accuracy of 77%. The model performed exceptionally well in detecting Class 2 (Severe Wear), with a precision of 0.86, recall of 0.93, and F1-score of 0.89. However, it struggled with Class 0 (Rapid Wear) and Class 1 (Uniform Wear), with lower recall values of 0.65 and 0.74, respectively. This highlights the limitations of ANN in handling complex patterns in the data, especially when features are not spatially localized.

## Performance of CNN

The CNN model, designed for image classification, demonstrated a superior overall accuracy of 83%. It excelled in identifying Class 0 (Rapid Wear), with a precision of 0.95 and recall of 1.00, indicating its effectiveness in correctly classifying this class. While its performance for Class 1 (Uniform Wear) and Class 2 (Severe Wear) was slightly lower, the CNN showed a better balance across all classes compared to the ANN. The use of convolutional layers enabled the CNN to capture spatial dependencies in the images, making it more robust for image-based tasks.

## Direct Comparison of Metrics

The table below summarizes the performance metrics for both models across all classes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Class | Precision | Recall | F1-Score |
| ANN | Class 0 (Rapid Wear) | 0.78 | 0.65 | 0.71 |
| ANN | Class 1 (Uniform Wear) | 0.67 | 0.74 | 0.70 |
| ANN | Class 2 (Severe Wear) | 0.86 | 0.93 | 0.89 |
| CNN | Class 0 (Rapid Wear) | 0.95 | 1.00 | 0.98 |
| CNN | Class 1 (Uniform Wear) | 0.76 | 0.80 | 0.78 |
| CNN | Class 2 (Severe Wear) | 0.75 | 0.67 | 0.71 |

## Conclusion

The comparison indicates that while the ANN model performed well for acoustic emission data, its limitations in capturing spatial relationships reduced its effectiveness for complex tasks. On the other hand, the CNN model leveraged its convolutional architecture to excel in image classification, achieving higher accuracy and better balance across all classes. For future work, combining both approaches—using ANN for acoustic emission data and CNN for images—could provide a comprehensive system for wear condition monitoring.

References:

Data Link: <https://www.kaggle.com/datasets/rabahba/phm-data-challenge-2010>

Chen, Bo-Xiang & Chen, Yi-Chung & Loh, Chee-Hoe & Chou, Ying-Chun & Wang, Fu-Cheng & Su, Chwen-Tzeng. (2022). Application of Generative Adversarial Network and Diverse Feature Extraction Methods to Enhance Classification Accuracy of Tool-Wear Status. Electronics. 11. 2364. 10.3390/electronics11152364. The means of accurately determining tool-wear status has long been important to manufacturers. Tool-wear status classification enables factories to avoid the unnecessary costs incurred by replacing tools too early and to prevent product damage caused by overly worn tools. While researchers have examined this topic for over a decade, most existing studies have focused on model development but have neglected two fundamental issues in machine learning: data imbalance and feature extraction. In view of this, we propose two improvements: (1) using a generative adversarial network to generate realistic computer numerical control machine vibration data to overcome data imbalance and (2) extracting features in the time domain, the frequency domain, and the time–frequency domain simultaneously for modeling and integrating these in an ensemble model. The experiment results demonstrate how both proposed modifications are reasonable and valid.