

# ADVANCING CHIPBOARD MILLING PROCESS MONITORING THROUGH SPECTROGRAM-BASED TIME SERIES ANALYSIS WITH CONVOLUTIONAL NEURAL NETWORK USING PRETRAINED NETWORKS

Jarosław Kurek<sup>1,\*</sup>, Karol Szymanowski<sup>2</sup>,

Leszek J. Chmielewski<sup>1</sup>, and Arkadiusz Orłowski<sup>1</sup>

<sup>1</sup>*Institute of Information Technology, Warsaw University of Life Sciences – SGGW*

<sup>2</sup>*Institute of Wood Sciences and Furniture, Warsaw University of Life Sciences – SGGW,  
Warsaw, Poland*

\*Corresponding author: Jarosław Kurek ([jaroslaw\\_kurek@sggw.edu.pl](mailto:jaroslaw_kurek@sggw.edu.pl))

**Abstract.** This paper presents a novel approach to enhance chipboard milling process monitoring in the furniture manufacturing sector using Convolutional Neural Networks (CNNs) with pretrained architectures like VGG16, VGG19, and RESNET34. The study leverages spectrogram representations of time-series data obtained during the milling process, providing a unique perspective on tool condition monitoring. The efficiency of the CNN models in accurately classifying tool conditions into distinct states ('Green', 'Yellow', and 'Red') based on wear levels is thoroughly evaluated. Experimental results demonstrate that VGG16 and VGG19 achieve high accuracy, however with longer training times, while RESNET34 offers faster training at the cost of reduced precision. This research not only highlights the potential of pretrained CNNs in industrial applications but also opens new avenues for predictive maintenance and quality control in manufacturing, underscoring the broader applicability of AI in industrial automation and monitoring systems.

**Key words:** convolutional neural networks, CNN, vgg16, vgg19, resnet34, tool state monitoring, chipboard milling.

## 1. Introduction

In the realm of furniture production, integrating sensor technology at various manufacturing stages is becoming a prevalent theme in automation research. This area is inherently intricate, involving numerous steps that demand precision. Adjustments are often necessary, especially when minor components are altered or added. The infusion of sophisticated technologies in these processes marks a significant leap forward, especially in tool condition monitoring. Mistimed or incorrect decisions regarding tool replacements could diminish product quality, potentially leading to losses for the manufacturing entity [3, 4, 5, 6, 8, 10, 11, 14, 15].

This paper primarily delves into the milling process, where precision in decision-making is crucial. Utilizing sensor-based technologies to monitor tool conditions offers a novel angle to address these challenges [7, 18, 21]. While tool state assessment can be done manually, it's a laborious process that interrupts production. Hence, automating this task is a substantial step forward in the industry.

Tool monitoring, a subject of extensive discussion and analysis, revolves around the progressive wear of the cutting edge, impacting product quality. An ideal automatic solution should prevent unnecessary production halts while ensuring timely tool replacement to avoid subpar output. The solution should be precise, offering real-time, automated feedback. Applying a dedicated array of sensors for capturing specific production line signals and analyzing this data appears to be an effective strategy.

A notable innovation in this study is the application of sensor data in addressing the complex challenges of tool condition monitoring. Although furniture manufacturing utilizes various materials, wood-based products are the most common. The data-driven approach for tool condition monitoring introduced here presents new opportunities for enhancing industry processes. Different signals are analyzed to determine their efficacy in detecting tool conditions during the machining process. Despite thorough documentation of these problems, there remains a demand for an automatic and precise solution that is easy to integrate into production environments.

In the realm of manufacturing and material processing, the milling of chipboard represents a significant domain, demanding consistent monitoring and analysis for quality control and process optimization. Recent advancements in machine learning and neural networks have opened up innovative pathways for enhancing these monitoring systems. This research delves into the intersection of these advanced technologies, focusing on the utilization of Convolutional Neural Networks (CNNs) for interpreting time series data derived from milling processes.

The core of this study lies in the exploration of spectrogram-based time series analysis, a technique which converts time series data into a more visually interpretable format, capturing both frequency and time information simultaneously. This approach leverages the intrinsic power of CNNs, particularly the renowned VGG16, VGG19, RESNET34 architectures, to analyze these spectrograms for insightful patterns and anomalies that are indicative of the chipboard milling process's condition and performance.

Applying a pretrained network [1, 8, 9, 12] as part of a CNN architecture presents a methodologically sound approach. Pretrained networks, having already learned rich feature representations from large datasets, offer a robust foundation for the model. When applied to the context of milling process monitoring, this model can detect subtle nuances and changes in the data, which are pivotal for predicting tool wear, identifying inefficiencies, and ensuring product quality.

Furthermore, this research is grounded in the analysis of real-world data, ensuring its relevance and applicability. Data collected during the milling process – such as noise, vibrations, and other pertinent physical parameters – are utilized to construct a comprehensive dataset. The application of the Short-Time Fourier Transform (STFT) and Discrete Wavelet Transform (DWT) to this dataset facilitates the generation of spectrograms. These spectrograms, in turn, serve as the input for the CNN model, allowing for a detailed and nuanced analysis of the data.

This study aims to demonstrate the effectiveness of using CNNs, particularly with the VGG16, VGG19, RESNET34 architectures, in monitoring and analyzing the milling process of chipboards. By advancing the methodological approaches to process monitoring and incorporating cutting-edge neural network technologies, this research contributes to the enhancement of quality control, process efficiency, and predictive maintenance in the manufacturing sector.

As the field of artificial intelligence continues to evolve, the implications of this research extend beyond its immediate application. The methodologies and findings presented here could inspire similar approaches in various industrial and manufacturing processes, paving the way for smarter, more efficient, and data-driven operations.

## 2. Dataset

This study aims to develop a diagnostic system that can accurately assess the wear level of tools during ongoing production activities. Data for this purpose were gathered through various means. The primary data collection was performed using a Jet 130 CNC machining center (Busellato, Italy), utilizing a 40 mm cutter head with a replaceable carbide cutting edge, supplied by Faba SA, Poland.

For the experimental setup, a chipboard panel measuring  $300 \times 150$  mm was applied. The panel was secured onto a measurement platform, where a 6mm deep groove was milled at a spindle speed of 18 000 rpm and a feed rate of 0.15 mm per tooth. These parameters were selected based on extensive literature review and practical experience in milling chipboard. The chosen rotational speed and feedrate are commonly used in the industry for optimal surface finish and minimal tool wear. The 6 mm cutting depth is a standard practice for milling chipboards of similar dimensions.

The condition of the tool was categorized into three distinct states: ‘Green’, ‘Yellow’, and ‘Red’. The ‘Green’ state indicates a new or well-conditioned tool, ‘Yellow’ signifies a tool in usable but worn condition, and ‘Red’ represents a tool that requires replacement due to significant wear. The classification was determined using the VBmax parameter.

The Figure 1 provides a microscopic view of the wear on a drill. This image illustrates the VBmax parameter, which is a key factor in classifying the tool’s condition. The VBmax parameter is used to categorize the wear level of the tool into different classes such as ‘Green’, ‘Yellow’, and ‘Red’, corresponding to new, moderately worn, and significantly worn conditions, respectively. This visual representation aids in comprehending how the VBmax parameter is quantified and its relevance in assessing tool wear. During the testing phase, operations were periodically halted to assess the tool’s condition using a Mitutoyo TM-505 microscope, ideal for dimensional and angular measurements. This microscope, capable of inspecting screw shapes and gears with an additional reticle, facilitated the categorization of wear states as follows:

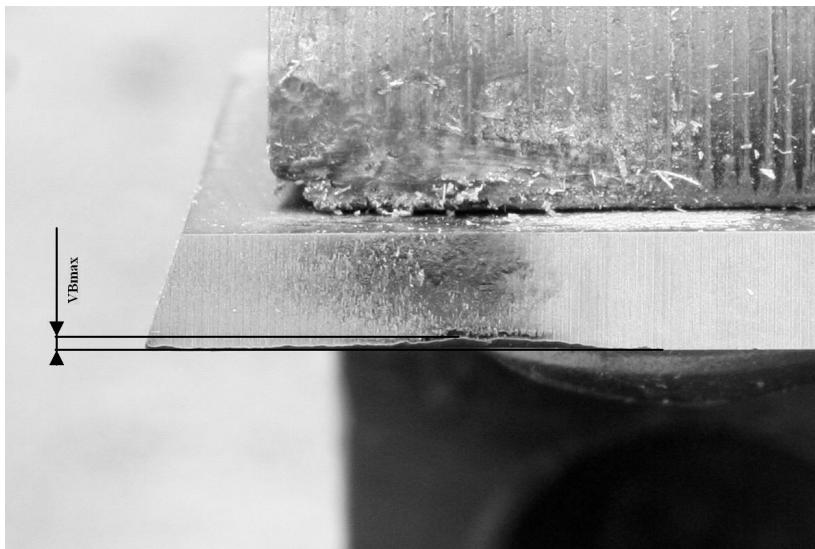


Fig. 1. Microscopic imagery showing the wear on a drill bit, highlighting the VBmax parameter pertinent to class evaluation.

- a ‘Green’ state is designated when VBmax is within the range of 0 to 0.15 mm, encompassing four levels of wear;
- a ‘Yellow’ state is defined for VBmax values between 0.151 and 0.299 mm, involving two wear levels;
- a ‘Red’ state is assigned for VBmax exceeding 0.299 mm, also with two levels of wear.

The experimental setup included a variety of sensors capable of monitoring 11 different parameters, such as:

- Force measurements in X and Y directions (Kistler 9601A sensor);
- Acoustic emission (Kistler 8152B sensor);
- Noise level (Brüel & Kjær 4189 sensor);
- Vibration intensity (Kistler 5127B sensor);
- Current and voltage ratings for the device, head, and servo (Finest HR 30 and Testec TT-Si9001 sensors).

All above items are depicted at Figure 2. This setup includes a detailed arrangement of sensors, acquisition cards, and the CNC machine tool used during the milling process. The configuration of these components is vital for capturing and analyzing data related to the milling process, such as vibrations, acoustic emissions, force measurements, and electrical signals. Understanding this setup is essential for appreciating how the data was collected, which forms the basis for the subsequent analysis using CNNs.

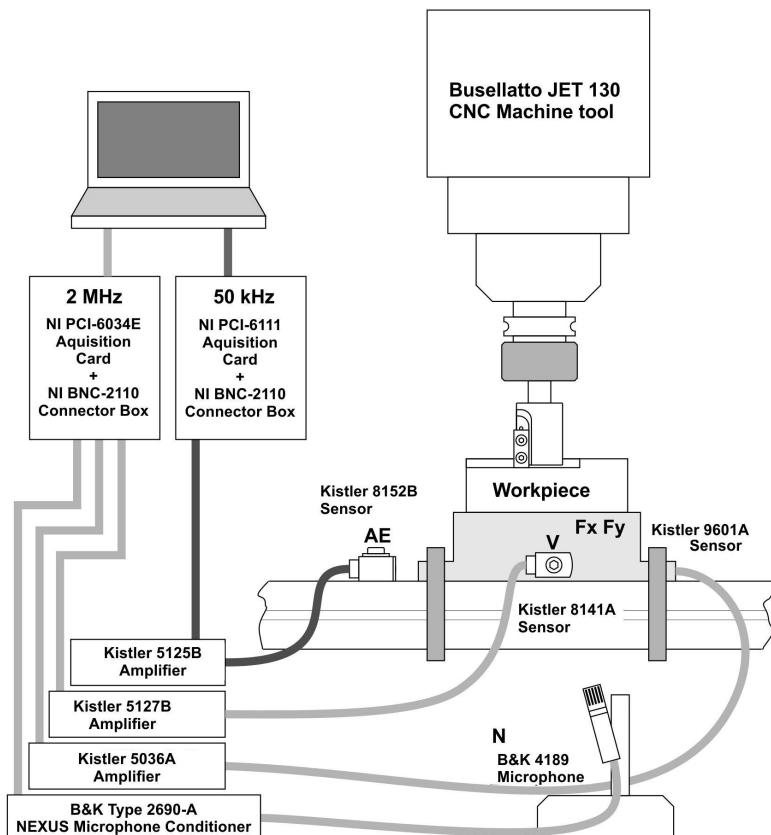


Fig. 2. Diagram of the experimental setup, illustrating the configuration of sensors, acquisition cards, and the CNC machine tool utilized in the experimental trials.

Data from these sensors were captured using National Instruments PCI-6111 and PCI-6034E measurement cards. The recording was conducted on a PC running National Instruments' Lab View™ software. The AE signal required a high-frequency card (2 MHz, 0.3 s window), while other signals were recorded with a 50 kHz card, using a 1.1 s window. Connections to the cards were made via BNC-2110 boxes.

To avoid training disruption by extraneous noise or sound variations, sensor positions relative to the workpiece and cutting zone were consistently maintained. The structure of the collected data is detailed in Table 1. It is organized into three columns and details various parameters measured across different categories of data. Each row represents a unique measurement type with its corresponding length and sampling rate in Hz. For

Tab. 1. Data Set Variable Configuration.

Variable Name	Measured Parameter	FS [Hz]
DataHigh	Acoustic Emissions	5 000 000
DataLow0	X-Axis Force	200 000
DataLow1	Y-Axis Force	200 000
DataLow2	Sound Level	200 000
DataLow3	Vibration Intensity	200 000
DataCurrent0	Device Current	50 000
DataCurrent1	Device Voltage	50 000
DataCurrent2	Head Unit Current	50 000
DataCurrent3	Head Unit Voltage	50 000
DataCurrent4	Servo Motor Current	50 000
DataCurrent5	Servo Motor Voltage	50 000

Tab. 2. Summary of Records and Data Classification.

Category	Number of Records	Number of Registered Signals per Record
Green Class	25	11
Yellow Class	25	11
Red Class	25	11
Total	75	

instance, the ‘High-Resolution Data’ category includes ‘Acoustic Emissions’ with a significant length of 27 999 960 and a very high sampling rate of 5 000 000 Hz. In contrast, ‘Standard Data’ encompasses measurements like ‘X-Axis Force’ and ‘Y-Axis Force’, each with 700 000 lengths and a sampling rate of 200 000 Hz. Lastly, ‘Electrical Data’ records parameters such as ‘Device Current’ and ‘Device Voltage’, each with a length of 30 000 and a sampling rate of 50 000 Hz.

The Table 2 condenses the dataset into three main classes: ‘Green’, ‘Yellow’, and ‘Red’. Each class has the same number of records, 25, and the same number of registered signals per record, 11. A total row is also included, summing the records to 75 and confirming the uniform number of signals per record across all classes. This table provides a high-level overview of the dataset’s classification structure.

The classification of tool wear during the milling process is significantly enhanced by the analysis of device current signals through spectrograms. In Fig. 3 a comparative study is presented of the spectral characteristics for three distinct tool condition classes – ‘Green’, ‘Yellow’, and ‘Red’ – as manifested in the device current signal. Spectrograms offer a two-dimensional representation where one axis represents time, the other frequency, and the color intensity indicates signal power at a given frequency and time.

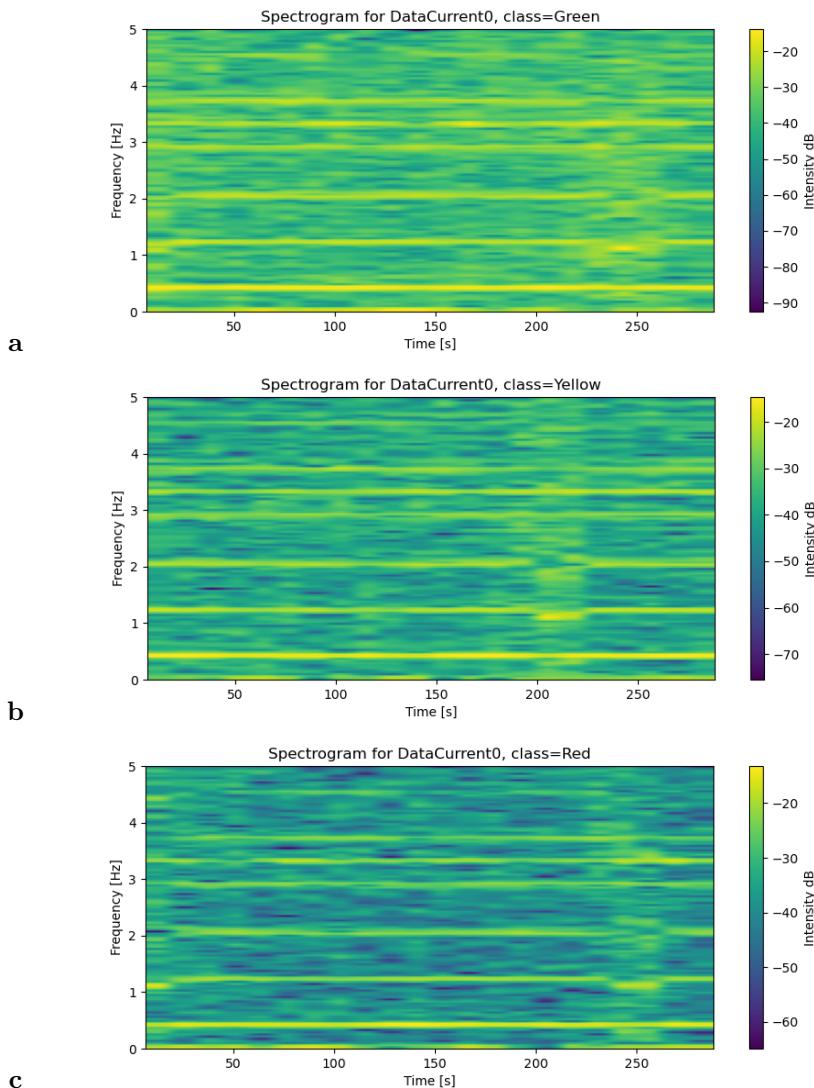


Fig. 3. Examples of spectrograms used in constructing a CNN model for classifying different conditions of a milling chipboard process. Each figure represents a unique class condition. (a) Spectrogram for class='Green', indicating a specific condition or state. (b) Spectrogram for class='Yellow', indicating a transition phase or intermediate state. (c) Spectrogram for class='Red', indicating a critical condition or state.

The spectrogram corresponding to the ‘Green’ class, as shown in the first panel of Figure 3a, exhibits a pattern characterized by uniformity across the frequency bands. This uniformity correlates with a well-conditioned tool exhibiting consistent operational behavior without any signs of wear. The frequency bands remain relatively undisturbed, indicating stable current consumption and, implicitly, a lack of significant resistance or stress on the milling apparatus.

The ‘Yellow’ class spectrogram, depicted in the second panel of Figure 3b, begins to show variations in the signal’s intensity across its frequency bands. These variations suggest the onset of tool wear, with certain frequencies becoming more pronounced, likely due to irregularities in the milling process as the tool encounters increasing resistance and degradation. This intermediate state of wear calls for closer monitoring to preempt any potential quality issues in the milling outcome.

The ‘Red’ class spectrogram, presented in the third panel of Figure 3c, shows a distinct pattern with significant disruptions in frequency stability. These disruptions reflect a critical level of tool wear, where the tool’s inefficiency is evidenced by erratic fluctuations in current consumption. The ‘Red’ condition indicates an immediate need for tool replacement to prevent substandard milling results and potential damage to the milling equipment.

A comparative analysis of the three spectrograms underscores the efficacy of using device current signal analysis for tool wear monitoring. The progression from ‘Green’ to ‘Red’ class is marked by a discernible transition in the signal’s spectral content, underscoring the potential of this approach in predictive maintenance and automated monitoring systems in industrial settings.

The spectral analysis of device current signals through spectrograms provides a robust mechanism for classifying tool condition in real-time. This method facilitates the early detection of tool wear, enabling timely interventions that can significantly enhance the efficiency and quality of the milling process.

### **3. Conversion of Time Series Signal into Spectrograms**

This chapter outlines the methodology applied to transform the time series data collected from the milling process into spectrograms. These spectrograms serve as inputs for the subsequent classification using CNNs. The conversion process is crucial for extracting meaningful features from the time series data, which are then leveraged by the CNN for accurate classification.

In this study, each color-coded spectrogram corresponds to a distinct class condition, with ‘Green’ representing a normal state, ‘Yellow’ an intermediate state, and ‘Red’ a critical state. The spectrograms provide a visual and quantitative way to discern the differences between these conditions over time.

### 3.1. Data Preparation and Interpolation

Initially, the time series data undergo extensive preprocessing. Given the varying lengths of the time series, a uniform length is necessary to ensure consistent analysis. To achieve this, the data is interpolated to a common length, maintaining the integrity of the original series. The interpolation is conducted using Python's `interp1d` function [16] from the `scipy` library, which allows for linear interpolation of the data. The following code snippet (Algorithm 1) demonstrates the interpolation process:

---

#### Algorithm 1 Interpolate time series data

---

```

1: procedure INTERPOLATETS(data, TargetLength, SamplingRate)
2:   dataSampled  $\leftarrow$  data[:: SamplingRate]
3:   XOld  $\leftarrow$  linspace(0, 1, length(dataSampled))
4:   XNew  $\leftarrow$  linspace(0, 1, TargetLength)
5:   interpolator  $\leftarrow$  interp1d(XOld, dataSampled, fillValue="extrapolate")
6:   return interpolator(XNew)
7: end procedure

```

---

### 3.2. Spectrogram Generation

Post-interpolation, each time series is converted into a spectrogram. a spectrogram is a visual representation of the spectrum of frequencies in a signal as they vary with time. This conversion is crucial for visualizing the frequency content of the time series data, which is key for the CNN's feature extraction phase.

Spectrograms play a pivotal role in the analysis of time series data, especially in the context of monitoring the milling process of chipboards. They offer a visual representation of the frequency spectrum of signals as they change over time. a spectrogram is a visual way of representing the signal strength, or "loudness", of a signal over time at various frequencies present in a particular waveform. It is a powerful tool for analyzing the frequency components of a signal that evolves over time. In our context, the spectrogram provides insights into the milling process by depicting the frequency content of vibrations, acoustic emissions, and other relevant signals.

The main parameters of the `specgram` [13] function used in this study are:

`seriesData` – the input signal data.

`Fs` (samplingRate) – the sampling frequency. It defines the number of data points sampled per second in the time series. In our case `Fs` depends on signal (Table 1).

`NFFT` – the number of data points used in each block for the Fast Fourier Transform (FFT). a higher number improves the frequency resolution of the spectrogram. In our case `NFFT=256`.

`noverlap` – the number of points of overlap between blocks. Increasing this value increases the continuity of the frequency spectrum over time but reduces temporal resolution. In our case `noverlap`=128.

The spectrograms are generated using the `specgram` function from Python. This function computes the spectrogram for each time series, showcasing the intensity of frequencies over time. Each spectrogram is then saved as an image file, which is later used as input for the CNN model. The code for generating the spectrograms is as follows (Algorithm 2):

---

**Algorithm 2** Generate spectrograms for each time series

---

```

1: procedure GENERATESPECTROGRAMS(DataSet, minLength, SamplingRate)
2:   for k in range(length(DataSet)) do
3:     for i in range(length(DataSet[k])) do
4:       DS  $\leftarrow$  DataSet[k][i]
5:       Define DataHigh0, DataLow0, ..., DataCurrent5 from DS
6:       DataHigh0_interp  $\leftarrow$  INTERPOLATES(DataHigh0, ...)
7:       ... ▷ Repeat for other data series
8:       for SeriesName, SeriesData in [('DataHigh0', DataHigh0_interp), ...] do
9:         Generate a spectrogram from SeriesData
10:        Save the spectrogram as an image file
11:      end for
12:    end for
13:  end for
14: end procedure
```

---

## 4. Model architecture with pretrained networks

### 4.1. CNN Pretrained networks

Convolutional Neural Networks are mainly known for their role in computer vision and image processing, which is a big part of deep learning. Traditionally associated with visual data analysis, the versatility of CNNs extends far beyond, as evidenced in this study. By repurposing these networks, initially designed for interpreting complex imagery, we have demonstrated their remarkable adaptability in processing time series data for industrial process monitoring. This innovative application leverages the inherent strengths of pretrained CNN models such as VGG16, VGG19, and RESNET34, traditionally employed in visual tasks, to analyze and interpret signals in the context of milling chipboard processes. This approach underscores a significant advancement in the application of AI,

extending the capabilities of CNNs from their conventional domain of image-based analysis to the intricate realm of signal processing and time series analysis in an industrial setting.

CNNs, especially those with pretrained architectures, have revolutionized the field of deep learning, particularly in image processing and computer vision. Among the most prominent of these are VGG16 [17, 19], VGG19 [17, 20], and RESNET34 [2], each having unique characteristics and strengths.

**VGG16** is renowned for its simplicity and depth. It consists of 16 layers, with a design focused on small convolution filters of size  $3 \times 3$  and the use of max pooling to reduce spatial dimensions. The VGG16 network has proven its effectiveness in feature extraction due to its depth and uniform architecture.

**VGG19**, an extension of VGG16, includes 19 layers. The additional layers in VGG19 allow it to learn more complex features from images, making it slightly more powerful than VGG16 at the expense of increased computational cost. Its effectiveness in handling more intricate image patterns makes it a preferred choice for complex image recognition tasks.

**RESNET34** marks a significant departure from traditional sequential architectures like VGG. The innovative use of residual connections, or ‘skip connections’, allows it to train deeper networks by addressing the vanishing gradient problem. This architecture comprises 34 layers and is exceptionally efficient in training due to these residual connections, which help preserve the gradient flow through the network. The RESNET34 architecture is known for its high performance in various tasks, including but not limited to image classification.

The application of these pretrained networks extends beyond traditional image processing. In various research and industry applications, these networks have been repurposed and fine-tuned for tasks such as signal processing, time-series analysis, etc. The pretrained aspect of these networks implies that they have been previously trained on large datasets, like ImageNet, allowing them to have a deep understanding of a wide range of features. This pre-training makes them incredibly versatile and efficient when adapted to new tasks.

In conclusion, the developed CNN model demonstrates significant potential for automating the monitoring of milling processes in the furniture manufacturing industry. By harnessing the power of advanced machine learning techniques, this model paves the way for more efficient, accurate, and automated tool condition monitoring.

The performance evaluation of the CNN model is a critical step in the development process. It allows us to assess the model’s ability to generalize to new data, which

is indicative of its practical utility in real-world applications. This section details the evaluation metrics and results obtained from testing the CNN model on the dataset for tool condition monitoring during the milling process.

#### **4.2. Model architecture**

The proposed model integrates the high-level feature extraction capabilities of the aforementioned pretrained networks, as illustrated in Figure 4. Each pretrained network serves as a separate feature extractor for a different input data stream. The architecture allows for the parallel processing of multiple data signals, each passing through a frozen base of a pretrained network. These signals correspond to the various parameters captured during the milling process, such as acoustic emissions, force measurements, and electrical signals.

Each data stream, denoted as DataHigh, DataLow0 through DataLow3, and DataCurrent0 through DataCurrent5, is processed through a distinct instance of the pretrained network. The pretrained network bases are kept frozen to preserve the knowledge they have acquired from vast amounts of visual data, ensuring the effectiveness of the feature extraction as seen in Figure 4.

Post feature extraction, the outputs from all pretrained networks are flattened and then concatenated into a single, common feature vector. This concatenated vector encapsulates comprehensive feature information important to the milling process, which is then passed through a series of fully connected (Dense) layers.

The first Dense layer consists of 512 neurons followed by batch normalization and a ReLU activation function. This is followed by a dropout layer with a dropout rate of 0.1 to mitigate the risk of overfitting. Subsequent layers reduce the dimensionality from 256 to 128 neurons, each followed by batch normalization, ReLU activation, and dropout layers.

The final layer in our architecture is a Dense layer with a softmax activation function. This layer outputs the probabilities corresponding to the different conditions of the tool – ‘Green’, ‘Yellow’, and ‘Red’ – indicating the level of wear and tear experienced by the milling apparatus.

The model’s versatility is underpinned by its ability to harness the strengths of each pretrained network. By adjusting the trainable parameters in the fully connected layers and maintaining the sophisticated feature extraction capabilities of the pretrained bases, our model is adept at providing nuanced classifications of tool conditions, thereby significantly enhancing the monitoring process.

The architecture of the model leverages the power of pretrained networks to analyze complex data streams effectively. The incorporation of these networks into our model reflects a significant step towards advanced monitoring and predictive maintenance in the domain of manufacturing.

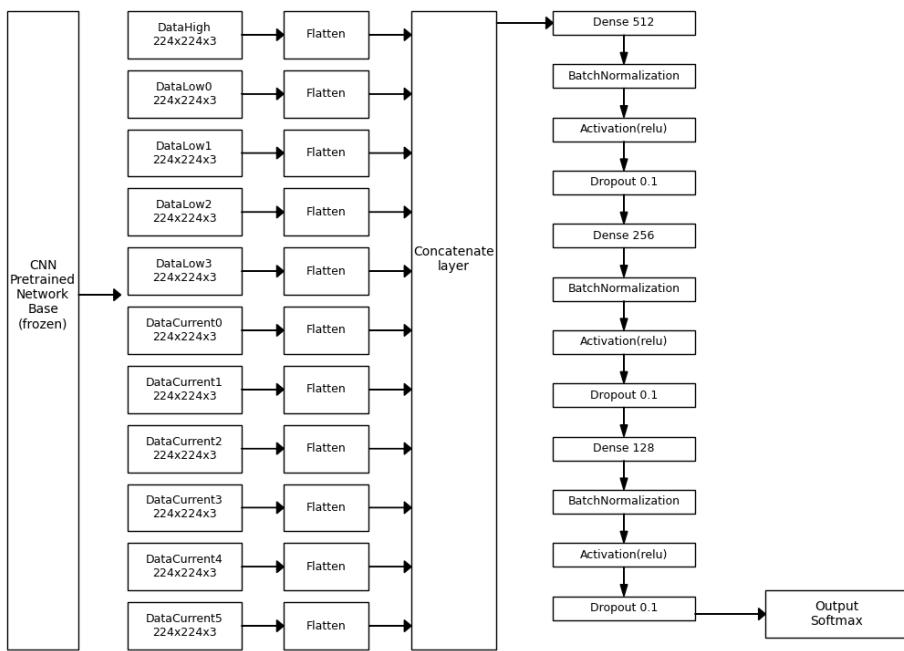


Fig. 4. Schematic representation of the CNN model utilizing pretrained networks for chipboard milling process monitoring.

## 5. Numerical Experiments

### 5.1. Cross-Validation Methodology

In this study, we applied cross-validation as a robust approach to evaluate the performance of our machine learning models. Specifically, we used the stratified  $K$ -fold cross-validation technique, ensuring that each fold is a good representative of the whole. Stratified  $K$ -fold divides the dataset into  $K$  equally sized folds (in our case  $K = 10$ ), with each fold containing approximately the same percentage of samples of each target class as the complete set.

For each fold in the cross-validation, every model was trained on the training subset and evaluated on the validation subset. The evaluation metrics included accuracy, confusion matrix, and a detailed classification report highlighting precision, recall, and F1-score for each class.

We also conducted a detailed error analysis by examining the confusion matrices.

Particular attention was given to extreme classification errors between the most dissimilar classes.

The following pseudocode provides an overview of our cross-validation implementation (Algorithm 3):

---

**Algorithm 3** Cross-Validation for Model Evaluation
 

---

```

data, labels ← loaddata()
kf ← StratifiedKFold( $n_{splits} = 10$ )
for fold, ( $train_{index}$ ,  $val_{index}$ ) in kf.split(data, labels) do
    model ← createmodel()
     $train_{data}, val_{data} \leftarrow data[train_{index}], data[val_{index}]$ 
     $train_{labels}, val_{labels} \leftarrow labels[train_{index}], labels[val_{index}]$ 
    model.fit( $train_{data}, train_{labels}$ )
     $YPred \leftarrow model.predict(val_{data})$ 
    Calculate and Analyze  $YPred$  against  $val_{labels}$ 
    Generate classification reports and confusion matrices for current fold
end for
Generate Final classification report and confusion matrix for all 10-folds.
  
```

---

The results of the cross-validation process, including accuracy scores, confusion matrices, and classification reports, are compiled and analyzed to assess the overall performance and robustness of every the models.

## 6. Hardware Configuration

Below Table 3 details the hardware setup utilized in our computational experiments. The hardware configuration played a crucial role in the successful implementation and execution of our CNN models. The table below summarizes the key hardware components used in our setup.

Our system was equipped with dual Intel(R) Xeon(R) Gold 5420+ processors. These high-performance processors are well-suited for demanding computational tasks, such as training deep neural networks. They ensure efficient parallel processing and fast computation speeds, which are essential for handling large datasets and complex algorithms.

The system's memory architecture included 2240 KiB of L1 cache, 56 MiB of L2 cache, and 52 MiB of L3 cache. Additionally, it boasted a substantial RAM capacity of 256 GiB, comprising 16 DIMM slots, each fitted with a 16 GiB module operating at 4800 MHz. This extensive memory setup was critical for managing the high data throughput and storage requirements of our CNN models.

For operating system and software, the setup included 2×128 GB NVMe disks, known

Tab. 3. Summary of hardware configuration.

Type	Hardware Item
Processor #1	Intel(R) Xeon(R) Gold 5420+
Processor #2	Intel(R) Xeon(R) Gold 5420+
Memory Cache L1	2240KiB L1 cache
Memory Cache L2	56MiB L2 cache
Memory Cache L3	52MiB L3 cache
Memory RAM	256GiB System Memory (16*16GiB DIMM 4800 MHz)
OS Disk	2x128 GB NVMe disk
Storage	26TB RAID
Network	4xNetXtreme BCM5720 Gigabit Ethernet PCIe
GPU #1	NVIDIA A40, 48 GB GDDR6 with ECC
GPU #2	NVIDIA A40, 48 GB GDDR6 with ECC
GPU #3	Integrated Matrox G200eW3 Graphics Controller

for their high-speed data transfer rates. The main data storage was a robust 26 TB RAID system, offering both large storage capacity and data redundancy, which is crucial for maintaining data integrity in large-scale computational experiments.

The network configuration comprised 4 NetXtreme BCM5720 Gigabit Ethernet PCIe adapters. This setup provided high-speed network connectivity, ensuring efficient data transfer within our computational network and facilitating remote access to the computational resources.

The computational rig was equipped with two NVIDIA A40 GPUs, each offering 48 GB of GDDR6 memory with ECC. These GPUs were instrumental in accelerating the training and inference processes of our CNN models. The integrated Matrox G200eW3 Graphics Controller served as an auxiliary GPU, primarily handling display outputs and less intensive graphical tasks.

This hardware configuration provided a robust and efficient platform for conducting our advanced computational experiments, particularly in training and evaluating deep neural network models.

## 7. Results and Discussion

This section presents the results obtained from the application of different pretrained CNN models, namely VGG16, VGG19, and RESNET34, in the monitoring of the milling chipboard process. The performance of each model is evaluated based on the confusion matrices and various performance metrics as summarized in Table 1.

## 7.1. Performance Evaluation

The evaluation of the pretrained CNN models involved analyzing their ability to accurately classify tool conditions into ‘Green’, ‘Yellow’, and ‘Red’ states. Figures 5a, b and c depict the confusion matrices for the VGG16, VGG19, and RESNET34 models, respectively (Table 4).

- The VGG16 and VGG19 models both achieved an accuracy of 80%, whereas the RESNET34 model showed a slightly lower accuracy of 70.67%.
- Critical errors, which are severe misclassifications, were slightly higher for the VGG19 model (3 errors) compared to VGG16 and RESNET34 (2 errors each).
- The training times varied among the models, with VGG19 taking the longest at approximately 1 hour and 10 min, followed by VGG16 at 55 min, and RESNET34 being the fastest at 32 minutes.

## 7.2. Class-Specific Analysis

A detailed analysis of each class’s performance reveals significant insights into the models’ classification capabilities:

- For the VGG16 model (Table 5), the ‘Green’ class showed the highest precision and specificity, indicating its effectiveness in correctly identifying well-conditioned tools

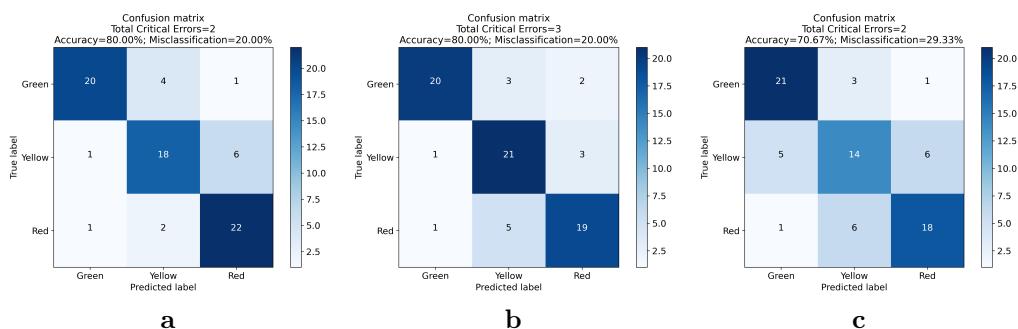


Fig. 5. Confusion matrices for the CNN model with three pretrained networks: (a) VGG16, (b) VGG19, (c) RESNET34.

Tab. 4. Comparison of Performance Metrics for Different Pretrained Neural Networks.

Pretrained network	Accuracy [%]	Critical Errors	Training Time
VGG16	80.00	2	55 min
VGG19	80.00	3	1 h 10 min
RESNET34	70.67	2	32 min

Tab. 5. Comparative analysis of VGG16 pretrained network performance across different classes.

Class	Precision [%]	Sensitivity [%]	F1-Score [%]	Specificity [%]
‘Green’	90.91	80.00	85.11	96.00
‘Yellow’	75.00	72.00	73.47	88.00
‘Red’	75.86	88.00	81.48	86.00

and minimizing false positives. The ‘Yellow’ and ‘Red’ classes had lower precision and sensitivity, pointing towards challenges in distinguishing between slightly worn and critically worn tool conditions.

- The VGG19 model (Table 6), while similar in precision and sensitivity for the ‘Green’ class as VGG16, showed better sensitivity for the ‘Yellow’ class and a balanced precision-sensitivity trade-off for the ‘Red’ class.
- The RESNET34 model (Table 7) exhibited lower overall precision and sensitivity across all classes, with the ‘Yellow’ class being the most challenging in terms of precision and sensitivity.

Tab. 6. Comparative analysis of VGG19 pretrained network performance across different classes.

Class	Precision [%]	Sensitivity [%]	F1-Score [%]	Specificity [%]
‘Green’	90.91	80.00	85.11	96.00
‘Yellow’	72.41	84.00	77.78	84.00
‘Red’	79.17	76.00	77.55	90.00

Tab. 7. Comparative analysis of RESNET34 pretrained network performance across different classes.

Class	Precision [%]	Sensitivity [%]	F1-Score [%]	Specificity [%]
‘Green’	77.78	84.00	80.77	88.00
‘Yellow’	60.87	56.00	58.33	82.00
‘Red’	72.00	72.00	72.00	86.00

### 7.3. Discussion

The results indicate that while all three models are capable of classifying tool conditions effectively, there are distinct differences in their performance. VGG16 and VGG19 models are more accurate but require longer training times, which might be a trade-off in real-time applications. The RESNET34 model, despite its lower accuracy, offers a faster training process, which could be beneficial in scenarios where rapid model deployment is necessary.

The higher precision in the ‘Green’ class across all models suggests that they are well-suited for identifying tools in good condition. However, the lower sensitivity in ‘Yellow’ and ‘Red’ classes, especially in the RESNET34 model, highlights the need for further model optimization to improve the detection of worn and critically worn tools.

In conclusion, the choice of a pretrained CNN model for tool condition monitoring in milling processes should be based on the specific requirements of accuracy, training time, and the ability to distinguish between different wear conditions. Future work may explore combining these models or employing ensemble methods to enhance overall performance and reliability.

## **8. Conclusion**

This study presented a comprehensive approach to monitoring and analyzing the milling chipboard process in furniture manufacturing using Convolutional Neural Networks with pretrained networks like VGG16, VGG19, and RESNET34. Our research demonstrated the effectiveness of these models in accurately classifying tool conditions into ‘Green’, ‘Yellow’, and ‘Red’ states based on their wear levels, utilizing spectrogram representations of time-series data collected during the milling process.

The experimental results showed that VGG16 and VGG19 achieved higher accuracy compared to RESNET34, although with a longer training time. This highlights a trade-off between accuracy and computational efficiency that must be considered in practical applications. While all models proved capable in identifying well-conditioned tools, challenges remain in differentiating between slightly worn and critically worn conditions, especially for the ‘Yellow’ and ‘Red’ classes.

Furthermore, the application of machine learning in this domain has opened new approaches for predictive maintenance and quality control in manufacturing. By enabling early detection of tool wear, the implemented models can significantly contribute to optimizing the milling process, reducing downtime, and ensuring product quality.

This research also underscores the potential of transfer learning, where pretrained models, originally developed for different tasks, can be successfully adapted and applied to specific industrial processes. It paves the way for further exploration of deep learning techniques in manufacturing, where similar strategies can be employed to enhance various aspects of production and monitoring systems.

In conclusion, the findings of this study contribute valuable insights into the application of advanced neural networks in industrial settings. Future work may focus on further refining these models, exploring ensemble methods for improved accuracy, and extending this approach to other manufacturing processes to fully harness the potential of AI in industrial automation and monitoring.

## References

- [1] M. Gao, D. Qi, H. Mu, and J. Chen. A transfer residual neural network based on ResNet-34 for detection of wood knot defects. *Forests*, 12(2), 2021. doi:10.3390/f12020212.
- [2] P. Iakubovskii (qubvel). Classification models Zoo – Keras (and TensorFlow Keras). [https://github.com/qubvel/classification\\_models](https://github.com/qubvel/classification_models), 2023. [Accessed: 2023-12-01].
- [3] P. Iskra and R. E. Hernández. Toward a process monitoring and control of a cnc wood router: Development of an adaptive control system for routing white birch. *Wood and Fiber Science*, 42(4):523–535, 2010. <https://wfs.swst.org/index.php/wfs/article/view/567>.
- [4] A. Jegorowa, J. Górska, J. Kurek, and M. Kruk. Initial study on the use of support vector machine (SVM) in tool condition monitoring in chipboard drilling. *European Journal of Wood and Wood Products*, 77(5):957–959, 2019. doi:10.1007/s00107-019-01428-5.
- [5] A. Jegorowa, J. Górska, J. Kurek, and M. Kruk. Use of nearest neighbors (K-NN) algorithm in tool condition identification in the case of drilling in melamine faced particleboard. *Maderas: Ciencia y Tecnología*, 22(2):189–96, 2020. doi:10.4067/S0718-221X2020005000205.
- [6] A. Jegorowa, J. Kurek, I. Antoniuk, W. Dołowa, M. Bukowski, et al. Deep learning methods for drill wear classification based on images of holes drilled in melamine faced chipboard. *Wood Science and Technology*, 55(1):271–293, 2021. doi:10.1007/s00226-020-01245-7.
- [7] R. J. Kuo. Multi-sensor integration for on-line tool wear estimation through artificial neural networks and fuzzy neural network. *Engineering Applications of Artificial Intelligence*, 13(3):249–261, 2000. doi:10.1016/S0952-1976(00)00008-7.
- [8] J. Kurek, I. Antoniuk, J. Górska, A. Jegorowa, B. Świderski, et al. Classifiers ensemble of transfer learning for improved drill wear classification using convolutional neural network. *Machine Graphics & Vision*, 28(1/4):13–23, 2019. doi:10.22630/MGV.2019.28.1.2.
- [9] J. Kurek, I. Antoniuk, J. Górska, A. Jegorowa, B. Świderski, et al. Data augmentation techniques for transfer learning improvement in drill wear classification using convolutional neural network. *Machine Graphics and Vision*, 28(1/4):3–12, 2019. doi:10.22630/MGV.2019.28.1.1.
- [10] J. Kurek, I. Antoniuk, B. Świderski, A. Jegorowa, and M. Bukowski. Application of siamese networks to the recognition of the drill wear state based on images of drilled holes. *Sensors (Switzerland)*, 20(23):1–16, 2020. doi:10.3390/s20236978.
- [11] J. Kurek, B. Świderski, A. Jegorowa, M. Kruk, and S. Osowski. Deep learning in assessment of drill condition on the basis of images of drilled holes. In: *Eighth International Conference on Graphic and Image Processing (ICGIP 2016)*, vol. 10225, pp. 375–381. SPIE, 2017. doi:10.1117/12.2266254.
- [12] S. Mascarenhas and M. Agarwal. A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for image classification. In: *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, vol. 1, pp. 96–99, 2021. doi:10.1109/CENTCON52345.2021.9687944.
- [13] matplotlib.pyplot.specgram – matplotlib 3.5.1 documentation. [https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.specgram.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.specgram.html), 2023. [Accessed: 15-11-2023].
- [14] S. Osowski, J. Kurek, M. Kruk, J. Górska, P. Hoser, et al. Developing automatic recognition system of drill wear in standard laminated chipboard drilling process. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, pp. 633–640, 2016. doi:10.1515/bpasts-2016-0071.
- [15] S. S. Panda, A. K. Singh, D. Chakraborty, and S. K. Pal. Drill wear monitoring using back propagation neural network. *Journal of Materials Processing Technology*, 172(2):283–290, 2006. doi:10.1016/j.jmatprotec.2005.10.021.

- [16] SciPy.interpolate.interp1d — scipy v1.8.0 reference guide. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html>, 2023. [Accessed: 15-11-2023].
- [17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2015. ArXiv.1409.1556. doi:10.48550/arXiv.1409.1556.
- [18] K. Szwajka and T. Trzepieciński. Effect of tool material on tool wear and delamination during machining of particleboard. *Journal of Wood Science*, 62(4):305–315, 2016. doi:10.1007/s10086-016-1555-6.
- [19] tf.keras.applications.vgg16.vgg16. [https://www.tensorflow.org/api\\_docs/python/tf/keras/applications/vgg16/VGG16](https://www.tensorflow.org/api_docs/python/tf/keras/applications/vgg16/VGG16). [Accessed: 2023-12-01].
- [20] tf.keras.applications.vgg19.vgg19. [https://www.tensorflow.org/api\\_docs/python/tf/keras/applications/vgg19/VGG19](https://www.tensorflow.org/api_docs/python/tf/keras/applications/vgg19/VGG19). [Accessed: 2023-12-01].
- [21] W. Wei, Y. Li, T. Xue, S. Tao, C. Mei, et al. The research progress of machining mechanisms in milling wood-based materials. *BioResources*, 13(1):2139–2149, 2018. doi:10.15376/biores.13.1.Wei.