Vol. 34, No. 3 (2024) | DOI: 10.37190/ord240303

OPEN ACCESS

Operations Research and Decisions

www.ord.pwr.edu.pl

OPERATIONS RESEARCH AND DECISIONS QUARTERLY

Multiple input CNN architecture for tool state recognition in the milling process based on time series signals

Michał Bukowski[1] Izabella Antoniuk[1] Karol Szymanowski[2] Artur Krupa[1] Jarosław Kurek[1*]

[1] *Department of Artificial Intelligence, Institute of Information Technology, Warsaw University of Life Sciences, Warsaw, Poland*

[2] *Department of Mechanical Processing of Wood, Institute of Wood Sciences and Furniture, Warsaw University of Life Sciences, Warsaw, Poland*

[*] *Corresponding author, email address: jaroslaw_kurek@sggw.edu.pl*

Abstract

The study presents a tailored application of a multiple-input convolutional neural network (CNN) for tool state recognition in the milling process. Our approach uniquely applies an 11-input CNN to classify tool wear in chipboard milling, utilizing scalogram images derived from time-series signals. The primary objective was to categorize tool wear into three classes: green, yellow, and red, signifying the progression of wear. The study involved 75 samples (25 samples per class), each comprising 11 signals transformed into scalograms via continuous wavelet transform. The dataset of 825 scalogram images enabled the development of a CNN-based diagnostic model, achieving a notable accuracy of 96.00%, which is an improvement over a previous methods (93.33%).

Keywords: *convolution neural network, tool condition monitoring, multiple input convolution neural network, deep learning*

1. Introduction

Furniture manufacturing is a complicated process, involving multiple steps and various problem along the way. Different stages require high levels of precision. Poor or ill-timed decisions about tool maintenance and/or exchange can result in a faulty product, not meeting the requirements and causing potential loss for the manufacturing company [6, 10, 14, 16]. One of the key elements influencing the quality of products, is tool condition monitoring during the milling process. This operation can be performed manually but in that instance it is a time consuming one, with high possibility of operator missing key indications of deteriorating tool state. Automating the process and providing additional insight is a high priority, while the overall process is widely and excessively discussed one [8, 29, 30]. Important aspect of this

Received 30 September 2023, accepted 3 July 2024, published online 17 October 2024
ISSN 2391-6060 (Online)/

The costs of publishing this issue have been co-financed by the program *Development of Academic Journals* of the Polish Ministry of Education and Science under agreement RCN/SP/0241/2021/1

procedure is the gradual deterioration of the cutting edge. In order to avoid unplanned and potentially problematic tool stoppage, an automatic and online-working solution is necessary. Such system usually needs to incorporate external identification of edge wear, such as signals recorded while the machine is operating [11, 27].

Numerous works focusing on wood-based materials are available [20, 21], with some approaches focused on determining most useful signals for identifying the tool condition during machining process [9, 13, 16, 23, 28, 31]. While the overall proceedings are well described, there still is a need for a more precise solutions, easier to incorporate in the production environment.

Machine learning is one area, where such solutions can be created. There are various methods and applications available, both for vision and signal based systems [1, 4, 5, 13, 25]. Different researchers consider problems related to the production process, its different aspects, and applications including very complex problem such as tree species recognition [7]. Such approaches show, that different algorithms can be well adapted to even most complicated tasks, assuming appropriate input data will be prepared.

There are various solutions focusing on different parts of the tool condition monitoring process. While the signals are common input in the wood industry applications, different solutions can include various machine learning approaches, often using different data. One common type of input are images. For example, CNNs are often paired with such data [6, 14, 15, 18, 19]. Such solutions often include various approaches to the problem of training artificial neural networks, like data augmentation or transfer learning using networks such as AlexNet [12, 22], prepared for ImageNet database [24, 26]. Although strictly image-based approaches are quite popular, there are some problems related to this kind of input. Usually in order to achieve satisfactory accuracy rate, large amounts of training data are necessary. Requirements for any given problem might differ, requiring tight cooperation with the manufacture. In that aspect changes in signals are easier to measure, assuming the proposed methodology is able to compute large amounts of data obtained from various sensors.

Using signals as a base input for a neural network poses series of problems. Firstly, the changes in signals are not always consistent throughout all data obtained from used sensor set. This can lead to problems in final solution, making it unable to accurately point out the key indicators of the deteriorating tool quality. Additionally, especially for signals that require very high precision while recording, the size of input data files can differ significantly between various sources. Research presented in this paper is largely based on the idea of using data thus available in a more optimal way, without losing the advantage of more precise measurements it provides.

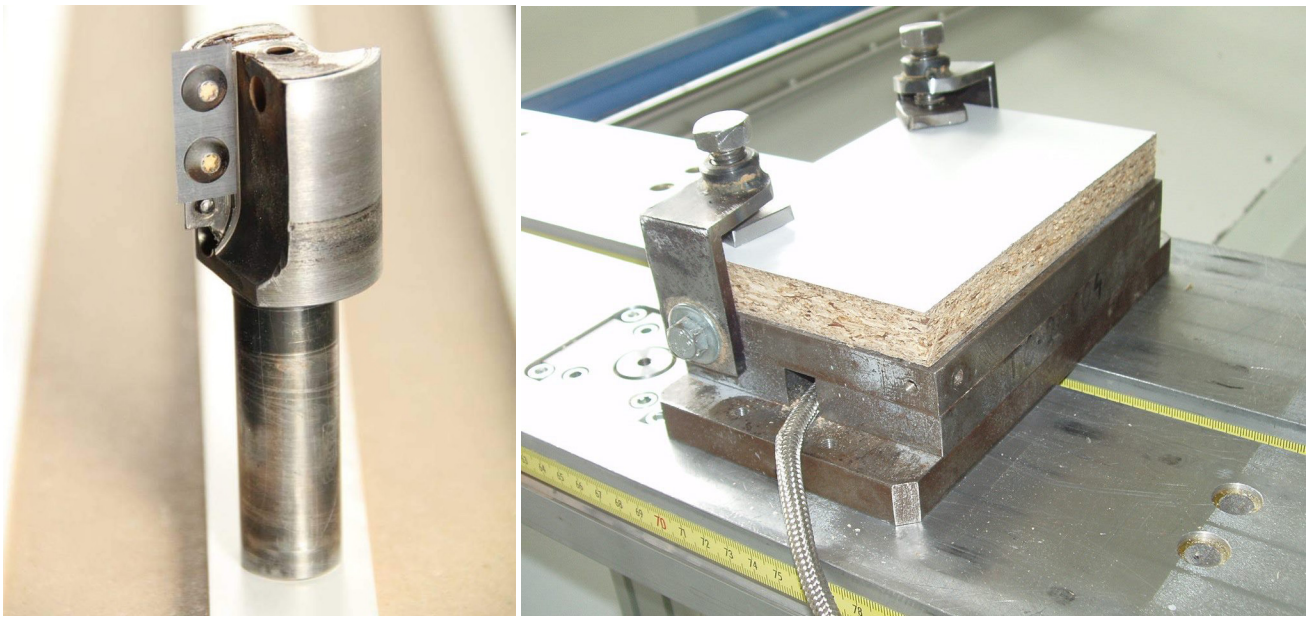
A relatively small section of approaches in this area considers transferring signals into images. For examplem, in [3] sound signals are converted to images using short-time Fourier transform. Authors first denoise the original signal and then convert it to images. After that process, the pre-trained CNN model is used to perform the deep feature extraction [18]. Last stage of method uses a support vector machine for the classification. Different solution [2], similarly to the approach presented in this paper, uses set of signals converted to scalograms. Authors use constant-Q transform with non-stationary Gabor transform, fusing features from vibration and acoustic signals with multiple input CNN in order to diagnose state of induction motor. They chose above methodology, with the assumption that computing continuous wavelet transform (CWT) is too time consuming, and in turn sacrificing the overall method accuracy.

In the presented research, a multiple input CNN architecture, using total of 11 separately recorded signals, is prepared and tested. Obtained data is down-sampled and converted to scalograms using CWT, in order to retain as much information from the original source as possible. To the best of authors knowledge, the proposed network architecture is first of its kind, both in general approach to signal preprocessing and incorporation as input, as well as the overall structure.

The rest of the work is organized as follows. In section 2, an overall discussion of used materials and data acquisition process is presented. Section 3 outlines the data preparation process, including the preprocessing stage and scalogram generation. The resulting dataset, CNN structure and performed experiments are described in section 4. Final paper conclusions are presented in section 6.

2. Materials

The main goal of the experiment was to build a diagnostic system capable to accurately measure the level of tool wear. The tests were done using a CNC machining centre (Jet 130; Busellato, Thiene, Italy), equipped with single edge cutter head with exchangeable carbide cutting edge, 40mm in diameter (Faba SA, Baboszewo, Poland), presented in Figure 1a). The mounted piece of chipboard is shown in Figure 1b).



a) **b)**

Figure 1. Edge cutter head (a) and example of chipboard piece mounted on the machining centre (b)

The state of the used tool was denoted as one of the tree states: green which means tool in a good state, yellow, defining tool in an intermediate state, and red, denoting tool that needs to be exchanged due to high wear level. The ranges for each state were measured using VB-max parameter defined as maximal flank wear. The ranges for each class were defined as follows:

- VB-max below 0.15 – green class;
- VB-max in the range: (0.151; 0.25) – yellow class;
- VB-max equal or above 0.3 – red class.

During the tests, a sample of a chipboard (DecoBoard P2, Pfleiderer) with dimensions of 300×150 mm was mounted on a measuring platform and a groove with a depth of 6 mm was milled. The spindle speed was set at 18,000 rpm with the feed rate equal to 0.15 mm per tooth.

2.1. Data acquisition

In order to ensure that acquired data is of the highest possible quality, a set of specialized sensors was selected for the measurement process. The entire set included following elements:

- X and Y forces (50kHz sampling rate) (Kistler 9601A sensor),
- acoustic emission (Kistler 8152B),
- noise (Brüel & Kjær 4189),
- vibrations (Kistler 5127B),
- finest HR 30 current.

During the tests, the milling process was stopped in order to determine the tool wear at different stages. The Mitutoyo TM-505 workshop microscope was used for those measurements. Standards used during the experiments are presented in Table 3 and the real physical properties are listed in Table 2. Full list of used equipment is shown in Table 1, while Figure 2 outlines the test stand setup.

Table 1. Apparatuses used to test the physical and mechanical parameters of wood-based materials

Parameter	Measuring apparatus
Density, kg/m^3	laboratory scale, calipers
Static bending strength, MPa	VebThuringerIndustriewerk SP 10
Elasticity modulus, MPa	VebThuringerIndustriewerk SP 10
Screw retention, N/mm	VebThuringerIndustriewerk SP 10
Tensile strength, MPa	VebThuringerIndustriewerk SP 10
Brinell hardness, HB	CV Instruments CV-3000LP8
Swelling 24 h, %	calipers
Water absorption 24h, %	calipers
Density profile	GreCon density Analyzer X-ray

Table 2. Selected mechanical and physical parameters of chipboard (Pfleiderer, DecoBoard P2 model)

Physical parameter	Value
Density, kg/m^3	670
Tensile strength, MPa	0.39
Swelling 24 h, %	61.8
Elasticity modulus, MPa	2950
Static bending strength, MPa	15.35

Obtained results were saved as a class label for the performed experiments. The registration was made at various degrees of wear:

- 4 times for the green state,
- 2 times for the yellow state,
- 3 times for the red state.

Two separate cards were used for acquisition, with different sampling speeds: National Instrument PCI-6111 and NIPCI-6034E. Tests were performed in accordance with CEN EN 310 (1994) and CEN EN 1534 (2020), using an Brinell CV 3000LDB tester (CV Instruments, Surrey, UK) as well as Instron 3382 testing machine (Norwood, MA, USA) respectively.

PC computer was used for the recording process, with National Instruments software, i.e. Lab ViewTM environment (National Instruments Corporation, ver. 2015 SP1, Austin, Texas, USA) using the NI PCI - 6034E and NI PCI – 6111 (Austin, Texas, USA) data acquisition cards. Two cards were used due to presence of signals with different frequency. To adequately record AE signal, card with high sampling frequency was required (2 MHz, measuring window of 0.3 s). Remaining signals were recorded at a frequency of 50 kHz, with 1.1s measuring window. The signals were connected to the cards separately for each frequency range, using BNC-2110 connection boxes.

Table 3. List of standards used in testing the properties of wood-based materials

Property	Norm
Static bending strength, MPa	PN-EN 310
Elasticity modulus, MPa	PN-EN 310
Screw retention, N/mm	PN-EN 320
Tensile strength, MPa	PN-EN 319
Swelling 24h, %	PN-EN 317
Water absorption 24h, %	PN-D-04234, PN-D-04213, PN-D-04213:1964
Sand content, %	ISO 3340 (PN-76/D-04245)

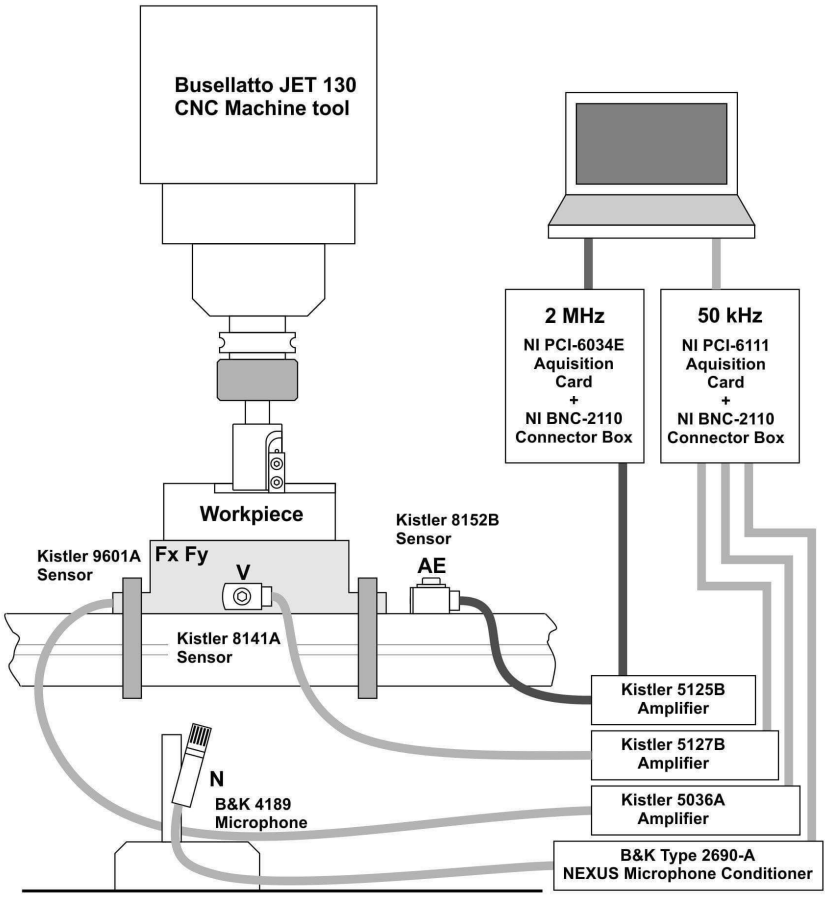


Figure 2. Outline of the measuring track used for the data collection during the machining process

In order to ensure minimal noise and changes in recorded signals, all sensors were kept in the same position in relation to the cutting zone and work-piece, through the entire measurement process. Outline of the measuring track is presented in Figure 2. The overall data structure of the obtained set is presented in Table 4.

Table 4. Data structure in the obtained dataset after recording all signals

Data set	Variable	Length of one trial	Sampling frequency [Hz]	Measure time [s]
DataHigh	acoustic emission	27,999,960	5,000,000	5.59
DataLow	force X	700,000	200,000	3.50
DataLow	force Y	700,000	200,000	3.50
DataLow	noise	700,000	200,000	3.50
DataLow	vibration	700,000	200,000	3.50
DataCurrent	device current	30,000	50,000	0.60
DataCurrent	device voltage	30,000	50,000	0.60
DataCurrent	head current	30,000	50,000	0.60
DataCurrent	head voltage	30,000	50,000	0.60
DataCurrent	servo current	30,000	50,000	0.60
DataCurrent	servo voltage	30,000	50,000	0.60

3. Data preparation

3.1. Downsampling based on reinterpolation

In the presented approach, the chosen input files were scalograms derived from the initial signals, represented as time series data. Due to large size of the more precisely measured signals, further usage can pose some problems. It was important to initially measure those elements with high accuracy, since using lower frequencies might have resulted in some errors and changes in the overall shape of signal curve. Therefore it was decided, that instead of lowering the initial quality of measured data, it will be reinterpolated before presenting it as an input for the multiple input CNN.

In order to retain the original signal shape, the entire length of it was divided into sections, and interpolated to feet in the given range. Since in presented experiments few signals of different lengths were recorded, they were all downsampled to the size of shortest signal (30,000). Figure 3 shows an example of originally registered signal, and its shape after the reinterpolation procedure. As can be seen, by using this method the general signal shape is retained, while the overall complexity is significantly reduced. Table 5 shows all original signal lengths, and final values after the downsampling procedure. Overview of the procedure used for this operation is presented in Algorithm 1.

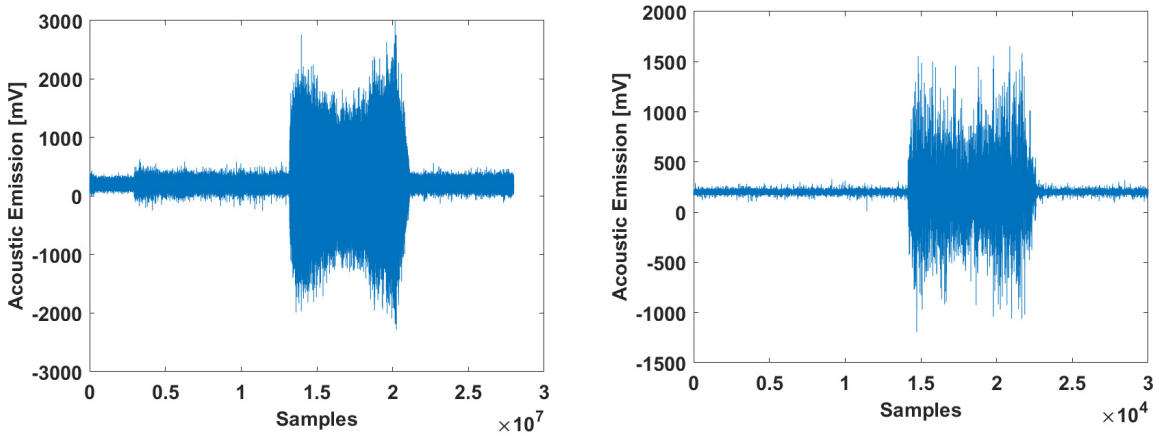
Input: $n > 1$ `/* desired length of the downsampled signal */`
Input: signal `/* original signal */`
$oL \leftarrow length(signal)$ `/* length of the original signal */`
$k \leftarrow linearspace(1, oL, n)$ `/* generate new resolution for the downsampled signal */`
$downsampled \leftarrow reinterpolation(1 : oL, signal, k)$ `/* output - downsampled signal */`

Algorithm 1. Finding the largest element

In the presented method linear space generates n points where spacing between the points equals $(oL-1)/(n-1)$. Reinterpolation function used is linear interpolation method defined as follows:

$$V_q = \text{reinterpolation}\ (X,\ V,\ X_q) \tag{1}$$

where: V_q – interpolated points used to find the underlying function $V = F(X)$ at the query points X_q, X – is a vector, V – is a vector with the same size as X, X_q is the same size as set of query points V_q.



a) **b)**

Figure 3. Original signal obtained during data acquisition (a), and a downsampled data used for scalogram generation (b); the general shape is retained

Table 5. The structure of the data variables in data sets

Data set	Signal	Length of one trial	Length of one trial after reinterpolation
DataHigh	acoustic emission	27,999,960	30,000
DataLow1	force X	700,000	30,000
DataLow2	force Y	700,000	30,000
DataLow3	noise	700,000	30,000
DataLow4	vibration	700,000	30,000
DataCurrent1	device current	30,000	30,000
DataCurrent2	device voltage	30,000	30,000
DataCurrent3	head current	30,000	30,000
DataCurrent4	head voltage	30,000	30,000
DataCurrent5	servo current	30,000	30,000
DataCurrent6	servo voltage	30,000	30,000

3.2. Scalogram generation

After the data reinterpolation process, all of the 75 data samples (each represented by 11 signals total), the signals are converted to scalograms. The images are then used as an input for the CNN network — initial layer has total of 11 inputs — one for each of the signals. Using scalograms images is a significant improvement, since it allows automatic feature extraction to take place. This removes the necessity for hand-crafting them.

A scalogram is a two-dimensional visual representation of a signal that provides information about its frequency and variation over time. It is obtained by applying the CWT to the signal in question. The

scalogram serves as a powerful tool for the analysis of non-stationary signals, enabling researchers to identify and characterize time-varying frequency patterns in the data.

In a scalogram, the X-axis represents time, and the Y-axis represents the scale (which is inversely proportional to frequency). The colour or intensity at each point in the plot signifies the magnitude of the wavelet coefficients, providing a measure of the signal's energy distribution across different time scales and frequencies. By examining the scalogram, one can discern localized features such as transient events or frequency modulations, which may not be readily apparent in traditional time-domain or frequency-domain representations.

From the tool condition monitoring point of view it is an important factor. Changes in individual signals might not be noticeable enough or achieved class thresholds can differ between them, blurring the results clarity. Using scalograms not only provides a reasonable way for representing changes in time, but also can provide additional insight into them. For approaches such as CNN, it can lead to more accurate and stable results.

In order to generate scalograms as accurately as possible, CWT with filter bank is used. Default wavelet for this is set as analytic Morse (3, 60) wavelet. The Morse wavelet is a parametric family of continuous wavelets that are well-suited for the analysis of non-stationary signals due to their ability to adapt their time-frequency localization properties. The Morse wavelet is characterized by two parameters, the order (γ) and the symmetry (β). These parameters control the time-frequency localization, concentration of energy, and the number of oscillations in the wavelet. The time-bandwidth and symmetry parameters for the Morse wavelets can both be varied to tune it for the specific solution needs. Additionally, analytic Morlet (Gabor) wavelet (or bump wavelet) can be used.

One additional operation that is performed in order to optimize the solution is precomputing the filters. Using multiple input CNN, with set of 11 signals is a computationally costly operation, hence improving the process is necessary. CWT can use such previously prepared filters as input, reducing number of calculation that need to be performed on the fly. In turn, with filter bank used, the wavelets can be visualized in time and frequency domains. In order to gain additional insight, filter banks with specific frequency or period ranges, and measure with multiple 3dB bandwidths can also be created. To further improve obtained results, quality factor can be defined for the wavelets in the filter bank.

For the presented approach, the filters are normalized so that the peak magnitudes for all passbands are approximately equal to 2. The default filter bank is designed for a signal with 1024 samples and uses the analytic Morse (3, 60) wavelet. Additionally it uses the default scales: approximately 10 wavelet bandpass filters per octave (10 voices per octave). The highest-frequency passband is designed so that the magnitude falls to half of the peak value at the Nyquist frequency.

As implemented, the CWT uses L1 normalization. With L1 normalization, equal amplitude oscillatory components at different scales have equal magnitude in the CWT. It provides a more accurate representation of the signal. The amplitudes of the oscillatory components agree with the amplitudes of the corresponding wavelet coefficients. Making sure that the resulting signals are as close to the original as possible was one of the key points in the presented experiments. L1 normalization, along with the data preparation process made sure, that as much of the original information will be retained as possible.The general overview of the used scalogram generation method is presented in Algorithm 2.

Require: $f_s > 0$ ▷ sampling frequency of input signal
Require: signal ▷ original signal
$n \leftarrow length(signal)$ ▷ length of the input signal
$fb \leftarrow cwtfilterbank(n, fs)$ ▷ generate a continuous wavelet transform (CWT) filter bank
$[cfs, frq] \leftarrow wt(fb, signal)$ ▷ generate continuous wavelet transform with filter bank
▷ cfs - continuous wavelet transform (CWT) coefficients
▷ frq - frequencies corresponding to the scales of cfs
$t = (0 : n-1)/f_s$ ▷ calculate time range based on sampling frequency
$scalogram\_image \leftarrow pcolor(t, frq, abs(cfs))$ ▷ create scalogram in the image form.

Algorithm 2. Scalogram generation algorithm

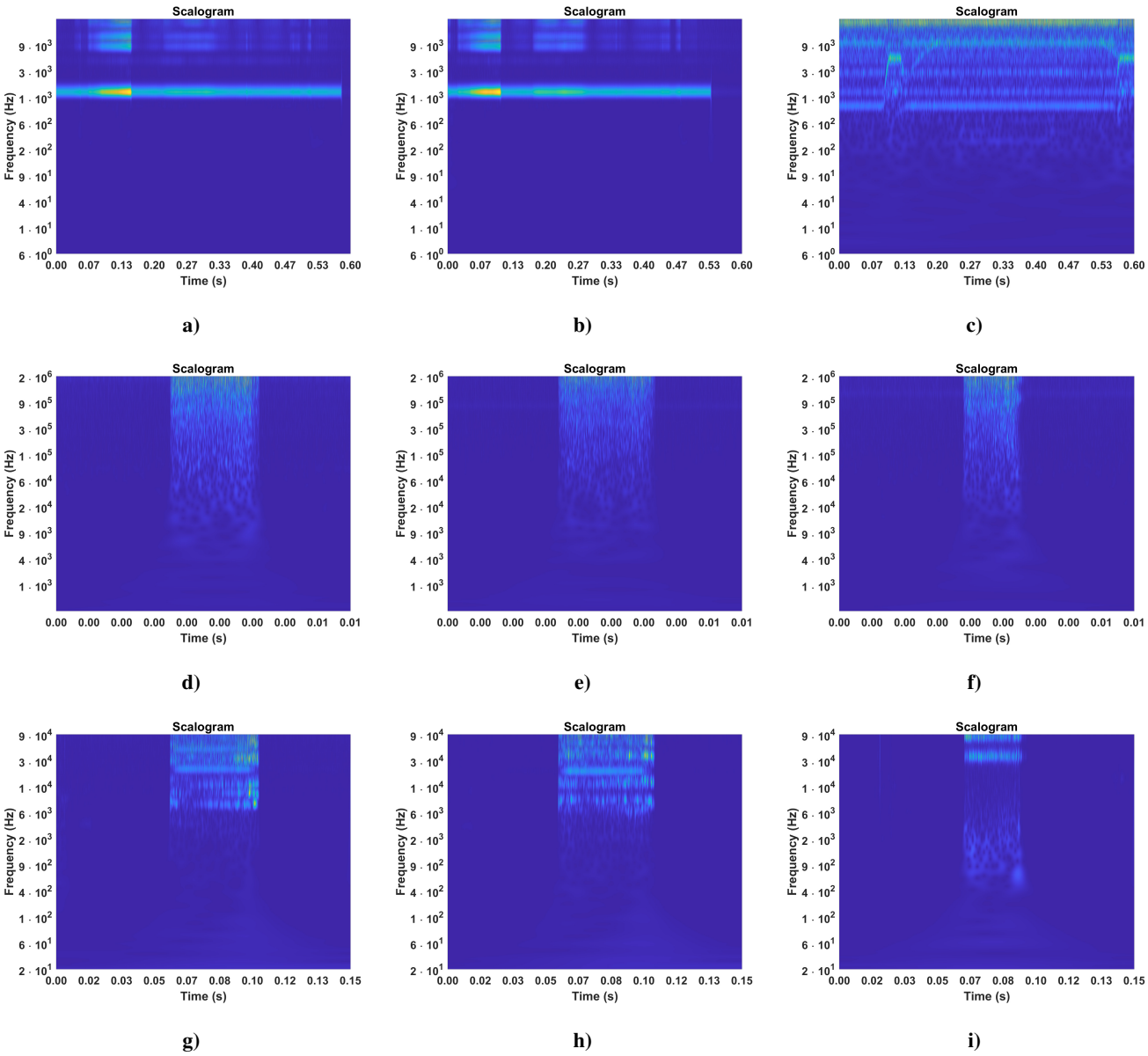


Figure 4. Exemplary scalograms generated for Curr1 (top row), High1 (middle row) and Low1 (bottom row) signals. For each scalogram, examples representing individual classes are shown in columns: a), d), g) – green, b), e), h) – yellow, c), f), i) – red

4. Methods

Previous approaches to tool wear classification show that even minimal adjustments can result in significant changes to the overall solution performance [14, 18, 19, 27]. Drawing from those experiences, the method proposed focuses on taking advantage of the strengths offered by different parts of the solution. For example, the use of signals ensures precise data, retaining information about the milling process and changing tool state. At the same time, this input in its unchanged form, introduces to much noise, blurring the borders between recognized classes. Using scalograms ensures, that as much of important information as possible will be retained in a form offering most advantages to the CNN. Finally, novel structure of the network in itself works well with the dataset prepared in this way.

4.1. Data sets for numerical experiments

The final dataset used for the network training consisted of 825 scalogram images, representing signals recorded during the milling process. The set was divided into training, validation and test sets, according to established practices in the artificial neural network training.

One important research goal presented in this paper was the incorporation of the overall manufacturer requirements in terms of solution performance. Key property in that case was reduction of critical errors between border classes:

- denoting example from green class as red, in which case a good tool will be discarded, resulting in unnecessary production downtime,
- classifying red instance as green, which can lead to poor product quality and financial loss associated with the need to discard such element.

Both cases are highly undesirable, and should be avoided.

4.2. Applying CNN for image classification

CNNs have emerged as a powerful tool for image classification tasks, with a profound impact in computer vision tasks. This class of deep learning models has demonstrated remarkable performance in classifying images by exploiting spatial, hierarchies and local connectivity patterns. CNNs employ a series of architectural components and techniques, to effectively learn and classify images. Such elements can include convolutional layers, activation functions, pooling layers, and fully connected layers.

The architecture of a CNN is typically composed of multiple layers arranged in a hierarchical manner. In this structure each layer performs a specific operation. The input image is processed sequentially through consecutive layers, ultimately resulting in a predicted class label. The main layers in a CNN are the convolutional layers, activation functions, pooling layers, and fully connected layers, which are interconnected to form a deep architecture. CNNs also employ batch normalization and dropout techniques to improve model stability and prevent overfitting.

A typical CNN architecture uses the following layers, arranged sequentially:

1. **Input layer.** The initial layer responsible for receiving raw image data in the form of a matrix with pixel values.

2. **Convolutional layers.** Layers performing the main computational operations by convolving the input image with learnable filters, detecting features such as edges, corners, and textures.
3. **Activation layers.** Following the convolutional layers, activation layers introduce non-linearity into the network by applying an activation function to the output of the convolution (such as the Rectified Linear Unit - ReLU).
4. **Pooling layers.** Layers performing the down-sampling operations to reduce the spatial dimensions of the feature maps, decreasing computational complexity and controlling overfitting.
5. **Fully connected layers.** Layers responsible for integrating high-level features extracted from the previous layers and making the final classification decision; they employ traditional feedforward neural network architecture and include an output layer with a Softmax activation function, generating class probabilities.

Additionally, key components of a CNN include:

- **Filters.** Also known as convolutional kernels; filters are learnable weight matrices that slide over the input image during the convolution operation; they are responsible for detecting specific patterns and features within the image.
- **Feature maps.** The output of the convolutional layers; feature maps represent the spatial arrangement of the detected features in the image.
- **Stride.** The step size by which filters slide over the input image during convolution, affecting the spatial dimensions of the resulting feature maps.
- **Padding.** The process of adding extra pixels around the input image before convolution, ensuring that the spatial dimensions of the feature maps are preserved.

Convolution used in CNN is a mathematical operation that combines the input image matrix and the filter matrix. It is performed by element-wise multiplication of the overlapping regions between the input and the filter, followed by summing up the results. This operation is repeated for each location in the input image, producing a feature map that highlights the presence of specific features.

During the training process, the CNN adjusts its filter weights to minimize the classification error. This is achieved through backpropagation - an algorithm that calculates the gradient of the loss function with respect to each weight by applying the chain rule. The gradients are then used to update the filter weights using optimization techniques. Used methods can include stochastic gradient descent (SGD) or more advanced methods like Adam.

To prevent overfitting and enhance generalization, CNNs employ regularization techniques such as L1 and L2 regularization, dropout, and batch normalization. Data augmentation is another approach to improving the model's performance. Such operations usually involve generation of new training samples by applying random transformations, such as rotation, scaling, or flipping, to the original images.

4.3. Multiple input CNN architecture

The multiple input CNN architecture proposed in this paper consist of 11 inputs. The layers are described as follows:

1. **ImageInputLayers** (11 inputs total). Architecture contains 11 ImageInputLayers, each accepting an input image of size 128×128×3, which means that each input image has a resolution of 128×128 pixels with 3 color channels (RGB).
2. **Convolution2DLayer** (11 layers). Each of the 11 ImageInputLayers is connected to its corresponding Convolution2DLayer with 64 filters of size 3×3×3. These layers perform the convolution operation on the input images, learning to extract relevant features from the input data.
3. **BatchNormalizationLayer** (11 layers). Each of the 11 Convolution2DLayer is connected to a BatchNormalizationLayer with 64 channels. These layers normalize the activations of the previous layer to stabilize the training process and improve convergence.
4. **ReLULayer** (11 layers). Each of the 11 BatchNormalizationLayers is connected to a ReLULayer. ReLU (Rectified Linear Unit) is an activation function that introduces non-linearity into the network by applying the function max(0, x) to the input, where x is the input value.
5. **MaxPooling2DLayer** (11 layers). Each of the 11 ReLULayers is connected to a MaxPooling2 DLayer with a stride of 1x1. Max-pooling reduces the spatial dimensions of the input by selecting the maximum value within a specified window size, which in this case is 1x1, meaning there is no reduction in spatial dimensions.
6. **DepthConcatenationLayer** (single layer). The 11 MaxPooling2DLayer outputs are concatenated along the depth dimension, forming a single tensor that is passed to the subsequent layers.
7. **Convolution2DLayer** (single layer). The DepthConcatenationLayer is connected to a Convolution2DLayer with 128 filters of size 3x3x704. This layer performs another round of feature extraction on the combined output from the previous layers.
8. **BatchNormalizationLayer** (single layer). This Convolution2DLayer is connected to a BatchNormalizationLayer with 128 channels, normalizing the activations before passing them to the next layer.

Table 6. Architecture of multiple inputs CNN model designed from scratch

No.	Layer	In	Out	Image size	Convolution	Stride padding	Batch norm
1	ImageInputLayer	0	1	128×128×3			
2	Convolution2DLayer	1	1		64 3×3×3	1×1/same	
3	BatchNormalizationLayer	1	1				64
4	ReLULayer	1	1				
5	MaxPooling2DLayer	1	1			1×1/same	
6	ImageInputLayer	0	1	128×128×3			
7	Convolution2DLayer	1	1		64 3×3×3	1×1/same	
8	BatchNormalizationLayer	1	1				64
9	ReLULayer	1	1				
10	MaxPooling2DLayer	1	1			1×1/same	
11	ImageInputLayer	0	1	128×128×3			
12	Convolution2DLayer	1	1		64 3×3×3	1×1/same	
13	BatchNormalizationLayer	1	1				64
14	ReLULayer	1	1				
15	MaxPooling2DLayer	1	1			1×1/same	
16	ImageInputLayer	0	1	128×128×3			

No.	Layer	In	Out	Image size	Convolution	Stride padding	Batch norm
17	Convolution2DLayer	1	1		64 3×3×3	1×1/same	
18	BatchNormalizationLayer	1	1				64
19	ReLULayer	1	1				
20	MaxPooling2DLayer	1	1			1×1/same	
21	ImageInputLayer	0	1	128×128×3			
22	Convolution2DLayer	1	1		64 3×3×3	1×1/same	
23	BatchNormalizationLayer	1	1				64
24	ReLULayer	1	1				
25	MaxPooling2DLayer	1	1			1×1/same	
26	ImageInputLayer	0	1	128×128×3			
27	Convolution2DLayer	1	1		64 3×3×3	1×1/same	
28	BatchNormalizationLayer	1	1				64
29	ReLULayer	1	1				
30	MaxPooling2DLayer	1	1			1×1/same	
31	ImageInputLayer	0	1	128×128×3			
32	Convolution2DLayer	1	1		64 3×3×3	1×1/same	
33	BatchNormalizationLayer	1	1				64
34	ReLULayer	1	1				
35	MaxPooling2DLayer	1	1			1×1/same	
36	ImageInputLayer	0	1	128×128×3			
37	Convolution2DLayer	1	1		64 3×3×3	1×1/same	
38	BatchNormalizationLayer	1	1				64
39	ReLULayer	1	1				
40	MaxPooling2DLayer	1	1			1×1/same	
41	ImageInputLayer	0	1	128×128×3			
42	Convolution2DLayer	1	1		64 3×3×3	1×1/same	
43	BatchNormalizationLayer	1	1				64
44	ReLULayer	1	1				
45	MaxPooling2DLayer	1	1			1×1/same	
46	ImageInputLayer	0	1	128×128×3			
47	Convolution2DLayer	1	1		64 3×3×3	1×1/same	
48	BatchNormalizationLayer	1	1				64
49	ReLULayer	1	1				
50	MaxPooling2DLayer	1	1			1×1/same	
51	ImageInputLayer	0	1	128×128×3			
52	Convolution2DLayer	1	1		64 3×3×3	1×1/same	
53	BatchNormalizationLayer	1	1				64
54	ReLULayer	1	1				
55	MaxPooling2DLayer	1	1			1×1/same	
56	DepthConcatenationLayer	11	1				
57	Convolution2DLayer	1	1		128 3×3×704	1×1/same	
58	BatchNormalizationLayer	1	1				128
59	ReLULayer	1	1				
60	FullyConnectedLayer	1	1	2097152××			
61	ReLULayer	1	1				
62	FullyConnectedLayer	1	1	128××			

No.	Layer	In	Out	Image size	Convolution	Stride padding	Batch norm
63	ReLULayer	1	1				
64	FullyConnectedLayer	1	1	64××			
65	ReLULayer	1	1				
66	FullyConnectedLayer	1	1	32××			
67	SoftmaxLayer	1	1				
68	ClassificationOutputLayer	1	0				

9. **ReLULayer** (single layer). This BatchNormalizationLayer is connected to a ReLULayer, introducing non-linearity into the network.
10. **FullyConnectedLayers** and ReLULayers (3 pairs). The architecture has three pairs of FullyConnectedLayers and ReLULayers with 128, 64, and 32 units, respectively. The FullyConnectedLayers enable the network to learn higher-level features and representations from the extracted features, while the ReLULayers introduce non-linearity.
11. **SoftmaxLayer**. The last FullyConnectedLayer is connected to a SoftmaxLayer with 3 classes. The SoftmaxLayer normalizes the input into a probability distribution over the 3 classes.
12. **ClassificationOutputLayer** (single layer). Finally, the architecture ends with a ClassificationOutputLayer, which computes the categorical cross-entropy loss for training and provides the final class predictions for the input images.

In summary, presented multiple input CNN architecture is a deep learning model with 11 input branches, each containing a series of Convolutional, Batch Normalization, ReLU, and MaxPooling layers. These branches are then combined using a DepthConcatenationLayer and followed by additional Convolutional, Batch Normalization, ReLU, FullyConnected, and Softmax layers to produce a final classification output. Overall network structure is visualized in Figure 5, while full outline of the used layers is presented in Table 6.

4.4. Improved approach for drill wear classification

As shown in previous research [17], connecting different classifiers into ensemble and using voting method to obtain final classification can improved overall results. Similar approach was used for final solution presented in this paper. While initial experiments obtained some acceptable results, there were still not satisfactory, and additional work to improve overall score was required. The evaluation of a machine learning model involves the partitioning of a dataset containing 75 samples into three distinct subsets: the training set, the validation set, and the test set. In this instance, the dataset is divided such that there are 60 samples for training, 10 samples for validation, and 5 samples for testing. Additionally, a k-fold cross-validation technique is employed, where $k = 15$.

The purpose of dividing the dataset into these three subsets is to ensure a rigorous and unbiased assessment of the model's performance. The training set is utilized for fitting the model, allowing it to learn patterns and relationships within the data. The validation set is used to tune the model's hyperparameters and to gauge its performance during the training process. This enables the identification of potential issues such as overfitting or underfitting, which can then be addressed before the final evaluation. Lastly, the test set serves as an independent subset of data that is employed to evaluate the model's performance,

Figure 5. Multiple input CNN architecture with 11 input image layers.

providing a reliable estimation of its generalization capabilities when applied to previously unseen data. The application of k-fold cross-validation is intended to further enhance the robustness of the evaluation process. This technique involves dividing the dataset into k equally sized partitions, where $k = 15$ in this case. The model is then trained and validated k times, with each partition serving as the validation set exactly once, while the remaining $k - 1$ partitions are combined to form the training set. The average performance across all k iterations is calculated to yield a more reliable and stable performance metric. This process helps to minimize potential biases arising from the initial partitioning of the dataset and provides a more accurate assessment of the model's true performance.

To transition from the k-fold ($k = 15$) cross-validation to a confusion matrix, you must first complete the k-fold cross-validation process and then aggregate the predictions for each fold. The steps involved in that process are as follows:

1. Perform k-fold cross-validation:
   A. Divide the dataset into k equally sized partitions (folds).
   B. For each fold, train the model on the remaining $k - 1$ folds combined as the training set, and validate the model on the current fold as the validation set.
   C. Store the predictions and true labels for each validation set.
2. Aggregate predictions:

A. Combine the predictions and true labels from all k validation sets to form a single set of predictions and true labels.

3. Generate the confusion matrix:
   A. Compare the aggregated predictions to the true labels and create a matrix that represents the number of occurrences for each possible combination of predicted and true classes.
   B. The rows of the matrix represent the true classes, while the columns represent the predicted classes. Each cell in the matrix contains the count of instances where the model predicted a particular class (column) when the true class was another (row).
4. Interpret the confusion matrix:
   A. Diagonal elements represent correct predictions (true positives and true negatives), while off-diagonal elements represent incorrect predictions (false positives and false negatives).
   B. Analyse the confusion matrix to determine the model's performance metrics, such as accuracy, precision, recall, and F1-score.

5. Results and discussion

The results for the proposed approach were compared with previously prepared solutions, including one based on gradient boosting, extreme gradient boosting and random forest algorithms. The outline of the training progress of our CNN model over 200 iterations is presented in Figure 6. The top panel displays the model's accuracy as a percentage, while the bottom panel shows the loss. In the accuracy graph, the solid blue line indicates the mean accuracy across the training batches, with the shaded area representing one standard deviation from the mean. The dashed black line shows the validation accuracy. Notably, the validation accuracy closely follows the training accuracy, suggesting a good generalization of the model. The black dots indicate the epochs where the model achieved a new peak in validation accuracy. The final accuracy achieved by the model is denoted by the 'Final' marker at the end of the training process. The loss graph illustrates the decline in both training (orange line) and validation (black line) loss over time, which is a typical behavior of a converging model. The initial sharp decrease indicates rapid learning, which gradually stabilizes as the model optimizes. The black dots represent points of minimum validation loss, coinciding with the peaks in validation accuracy.

Obtained accuracy results for all methods, and final solution presented in this paper are presented in Table 7. Confusion matrix outlining the overall class predictions, with associated classification errors is shown in Figure 7. As can be seen, the presented approach performed well both in terms of overall accuracy – achieving highest score from all of the methods – and overall classification. Total of 3 instances were misclassified: 2 times example from red class was classified as yellow, and once yellow class was classified as red. It is important to note, that there are no instanced of red-green or green-red misclassifications (the most influential ones in terms of tool condition monitoring).

The overall performance of the multiple inputs CNN model is summarized in the Table 8 The model achieved an impressive overall accuracy of 96.00%. This high accuracy indicates that the model is highly effective in classifying the tool condition into the correct categories (green, yellow, red) based on the input scalogram images derived from the milling process signals. Furthermore, the precision (macro average)

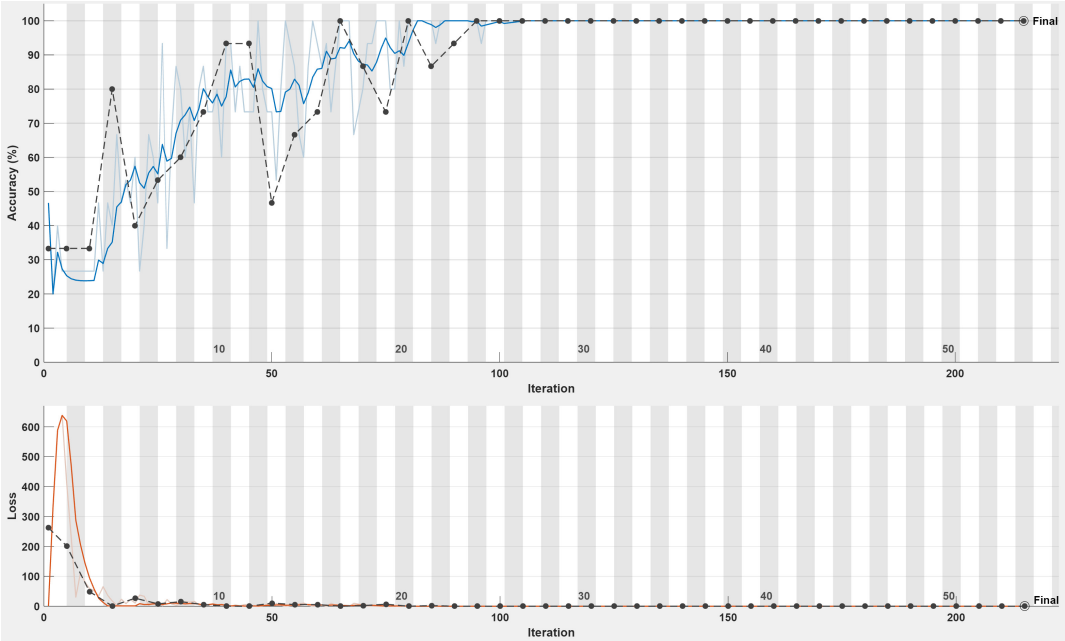


Figure 6. Outline of the training process for the proposed multiple input CNN architecture

Table 7. Final classification results with previously prepared approaches

Mode	Parameters	Accuracy [%]
Multiple inputs CNN	269.2M total learnables	96.00
Extreme gradient boosting	learning_rate = 0.1 n_estimators = 100	93.33
Random forest	n_estimators = 100 min_samples_split = 2 min_samples_leaf = 1	86.66
Gradient boosting	min_samples_split = 2 min_samples_leaf = 1	86.66

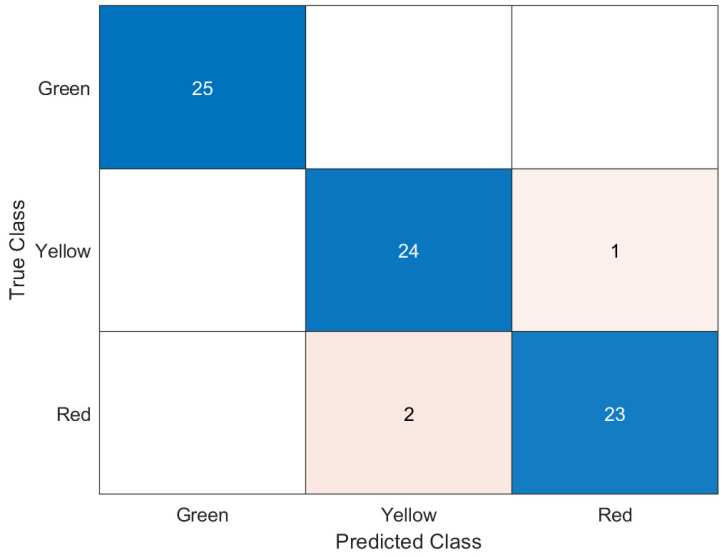


Figure 7. Confusion matrix outlining classification results for the proposed solution

of the model is 96.05%, suggesting that the model has a high level of reliability in its predictions. The recall (sensitivity, macro average) stands at 96.00%, which means that the model is capable of correctly identifying the majority of the relevant instances across all classes. The F1 score (macro average), which is a balance between precision and recall, is calculated to be 95.99%. This high F1 score underscores the model's balanced performance in both precision and sensitivity.

Table 8. Overall classification metrics for multiple inputs CNN model

Metric	Value [%]
Overall accuracy	96.00
Precision (macro average)	96.05
Recall (sensitivity, macro average)	96.00
F1 Score (macro average)	95.99

A more detailed insight into the model's performance is provided by the class-wise classification metrics, as presented in the Table 9. This analysis allows us to understand how the model performs for each specific class.

Table 9. Class-wise classification metrics for multiple inputs CNN model [%]

Class	Precision	Sensitivity	Specificity	F1 score
Green	100.00	100.00	100.00	100.00
Yellow	92.31	96.00	96.00	94.12
Red	95.83	92.00	98.00	93.88

Green class. For the green class, which represents tools in good condition, the model achieved 100% precision, sensitivity, and F1 score. This result indicates a perfect classification performance for this class, with no misclassifications.

Yellow class. The model showed a precision of 92.31% and a sensitivity of 96.00% for the yellow class, indicating tools in an intermediate state. The slightly lower precision suggests a few instances of over-predicting the yellow class, but a high sensitivity indicates a strong ability to correctly identify most of the yellow class instances. The F1 score for this class is 94.12%.

Red class. For the red class, indicating tools that need to be exchanged due to high wear, the model achieved a precision of 95.83% and a sensitivity of 92.00%. The high precision shows the model's ability to correctly identify the red class instances with minimal false positives, while the slightly lower sensitivity indicates some misses in identifying all the red class instances. The F1 score for the red class is 93.88%.

The results of the multiple inputs CNN model demonstrate its effectiveness in tool condition monitoring in a milling process. The high overall accuracy and balanced precision and recall across all classes indicate that the model is robust and reliable. Particularly noteworthy is the model's exceptional performance in classifying the green class without any errors, which is crucial for avoiding unnecessary tool changes and associated downtime. The slight variations in precision and sensitivity for the yellow and red classes suggest areas for further refinement. However, these results are still highly promising, indicating that the model can successfully differentiate between the varying degrees of tool wear, which is essential for effective tool maintenance and cost reduction.

In conclusion, the multiple inputs CNN model exhibits a strong potential for practical applications in tool condition monitoring. Future work may focus on further optimization of the model and expanding the dataset to enhance the model's generalization capabilities.

6. Conclusions

In this study, we introduced a multiple input CNN architecture, tailored for tool state recognition in milling processes. The development and implementation of this model have led to several significant findings and advancements in the field.

Foremost among these is the model's exceptional accuracy in classifying tool wear. It achieved an overall accuracy of 96.00%, a notable improvement over conventional methods. This high degree of accuracy not only demonstrates the model's robustness but also its reliability for practical applications in industrial settings.

The model's performance in classifying the state of tools was particularly remarkable. For tools in good condition (green class), it achieved perfect scores in precision, sensitivity, and F1 score. When identifying tools in an intermediate state (yellow class), the model showed a precision of 92.31% and a sensitivity of 96.00%. In the critical red class, indicative of tools requiring replacement, the model's precision and sensitivity were 95.83% and 92.00%, respectively. These results highlight the model's adeptness at distinguishing between various levels of tool wear with a high degree of accuracy.

Another key aspect of our study was the effective use of scalogram images derived from time-series signals. This approach allowed the CNN to extract detailed and nuanced features, facilitating the identification of complex patterns indicative of tool wear. Such a method is evidence of the significant potential of deep learning techniques in industrial and manufacturing applications.

The ability of the model to minimize errors in tool wear categorization is particularly beneficial for its application in real-world scenarios. This precision is vital in reducing unnecessary tool changes, optimizing production efficiency, and ensuring the quality of the final product.

However, our study also identified areas for potential improvement, especially in the yellow and red classes, where there were slight variations in precision and sensitivity. This finding provides a clear direction for future research and efforts to optimize the model.

In conclusion, the multiple input CNN model showcased in this study represents a significant step forward in the realm of tool condition monitoring. Its high accuracy and nuanced class-specific performance have the potential to revolutionize milling processes by enhancing tool maintenance, reducing operational costs, and ensuring product quality. Future research will aim to refine this model further, expand the dataset for more comprehensive training, and explore its application in various other industrial contexts.

References

[1] Bengio, Y. Learning deep architectures for AI. *Foundations and trends® in Machine Learning 2*, 1 (2009), 1–127.

[2] Choudhary, A., Mishra, R. K., Fatima, S., and Panigrahi, B. K. Multi-input CNN based vibro-acoustic fusion for accurate fault diagnosis of induction motor. *Engineering Applications of Artificial Intelligence 120* (2023), 105872.

[3] Demir, F., Turkoglu, M., Aslan, M., and Sengur, A. A new pyramidal concatenated CNN approach for environmental sound classification. *Applied Acoustics 170* (2020), 107520.

[4] Deng, L., and Yu, D. Deep learning: methods and applications. *Foundations and trends® in signal processing 7*, 3–4 (2014), 197–387.

[5] Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT Press, 2016.

[6] Hu, J., Song, W., Zhang, W., Zhao, Y., and Yilmaz, A. Deep learning for use in lumber classification tasks. *Wood Science and Technology 53*, 2 (2019), 505–517.

[7] Ibrahim, I., Khairuddin, A. S. M., Abu Talip, M. S., Arof, H., and Yusof, R. Tree species recognition system based on macroscopic image analysis. *Wood Science and Technology 51* (2017), 431–444.

[8] Iskra, P., and Hernández, R. E. 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 (2010), 523–535.

[9] Jegorowa, A., Górski, J., Kurek, J., and Kruk, M. Use of nearest neighbors (k-NN) algorithm in tool condition identification in the case of drilling in melamine faced particleboard. *Maderas: Ciencia y Tecnologia 22*, 2 (2020), 189 – 196.

[10] Jegorowa, A., Kurek, J., Antoniuk, I., Dołowa, W., Bukowski, M., and Czarniak, P. Deep learning methods for drill wear classification based on images of holes drilled in melamine faced chipboard. *Wood Science and Technology 55*, 1 (2021), 271–293.

[11] Jemielniak, K., Urbański, T., Kossakowska, J., and Bombiński, S. Tool condition monitoring based on numerous signal features. *The International Journal of Advanced Manufacturing Technology 59*, 1-4 (2012), 73–81.

[12] Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM 60*, 6 (2017), 84–90.

[13] Kuo, R. J. Multi-sensor integration for on-line tool wear estimation through artificial neural networks and fuzzy neural network. *Engineering Applications of Artificial Intelligence 13*, 3 (2000), 249–261.

[14] Kurek, J., Antoniuk, I., Górski, J., Jegorowa, A., Świderski, B., Kruk, M., Wieczorek, G., Pach, J., Orłowski, A., and Aleksiejuk-Gawron, J. Classifiers ensemble of transfer learning for improved drill wear classification using convolutional neural network. *Machine Graphics &Vision 28*, 1/4 (2019), 13–23.

[15] Kurek, J., Antoniuk, I., Górski, J., Jegorowa, A., Świderski, B., Kruk, M., Wieczorek, G., Pach, J., Orłowski, A., and Aleksiejuk-Gawron, J. Data augmentation techniques for transfer learning improvement in drill wear classification using convolutional neural network. *Machine Graphics &Vision 28*, 1/4 (2019), 3–12.

[16] Kurek, J., Kruk, Osowski, S., M., Hoser, P., Wieczorek, G., Jegorowa, Górski, J., A., Wilkowski, J., Śmietańska, K., and Kossakowska, J. Developing automatic recognition system of drill wear in standard laminated chipboard drilling process. *Bulletin of the Polish Academy of Sciences: Technical Sciences 64*, 3 (2016), 633–640.

[17] Kurek, J., Krupa, A., Antoniuk, I., Akhmet, A., Abdiomar, U., Bukowski, M., and Szymanowski, K. Improved drill state recognition during milling process using artificial intelligence. *Sensors 23*, 1 (2023), 448.

[18] Kurek, J., Świderski, B., Jegorowa, A., Kruk, M., and Osowski, S. 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) 29-31 October 2016, Tokyo, Japan (2017)* (Bellingham, 2017), T. Pham, V.Vozenilek and Z. Zeng, Eds., vol. 10225, SPIE, pp. 375–381.

[19] Kurek, J., Wieczorek, G., Świderski, B., Kruk, M., Jegorowa, A., and Osowski, S. Transfer learning in recognition of drill wear using convolutional neural network. In *2017 18th International Conference on Computational Problems of Electrical Engineering (CPEE) 11-13 September 2017, Kutna Hora, Czech Republic* (2017), IEEE, pp. 1–4.

[20] Lemaster, R. L., Lu, L., and Jackson, S. The use of process monitoring techniques on a CNC wood. Part 1. Sensor selection. *Forest Products Journal 50*, 7/8 (2000), 31–38.

[21] Lemaster, R. L., Lu, L., and Jackson, S. The use of process monitoring techniques on a CNC wood router. Part 2. Use of a vibration accelerometer to monitor tool wear and workpiece quality. *Forest Products Journal 50*, 9 (2000), 59-64.

[22] Lin, K. K.-Y. (2020) Github repository for AlexNet model (accessed on 5 August 2023).

[23] Panda, S. S., Singh, A. K., Chakraborty, D., and Pal, S. K. Drill wear monitoring using back propagation neural network. *Journal of Materials Processing Technology 172*, 2 (2006), 283–290.

[24] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg A. C. and Fei-Fei L. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision 115*, 3 (2015), 211–252.

[25] Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks 61* (2015), 85–117.

[26] Stanford Vision Lab, Stanford University, Princton University (2020) ImageNet web page (accessed on 5 August 2023).

[27] Świderski, B., Antoniuk, I., Kurek, J., Bukowski, M., Górski, J., and Jegorowa, A. Tool condition monitoring for the chipboard drilling process using automatic, signal-based tool state evaluation. *BioResources 17*, 3 (2022), 5349–5371.

[28] Świderski, B., Kurek, J., Osowski, S., Kruk, M., and Jegorowa, A. Diagnostic system of drill condition in laminated chipboard drilling process. In *21st International Conference on Circuits, Systems, Communications and Computers (CSCC 2017) Agia Pelagia Beach, Heraklion, Crete, Greece, July 14-17, 2017)*, N. Mastorakis, V. Mladenov and A. Bulucea, Eds., vol. 125 of *MATEC Web of Conferences*, EDP Sciences, 04002.

[29] Szwajka, K., and Trzepieciński, T. Effect of tool material on tool wear and delamination during machining of particleboard. *Journal of Wood Science 62*, 4 (2016), 305–315.

[30] Wei, W., Li, Y., Xue, T., Tao, S., Mei, C., Zhou, W., Wang, J., and Wang, T. The research progress of machining mechanisms in milling wood-based materials. *BioResources 13*, 1 (2018), 2139–2149.

[31] Wilkowski, J., and Górski, J. Vibro-acoustic signals as a source of information about tool wear during laminated chipboard milling. *Wood Research 56*, 1 (2011), 57–66.