

## Article

# A CNN Prediction Method for Belt Grinding Tool Wear in a Polishing Process Utilizing 3-Axes Force and Vibration Data

Wahyu Caesarendra <sup>1,\*</sup>, Triwiyanto Triwiyanto <sup>2</sup>, Vigneashwara Pandiyan <sup>3</sup>, Adam Glowacz <sup>4,\*</sup>,  
Silvester Dian Handy Permana <sup>5</sup> and Tegoeh Tjahjowidodo <sup>6,7,\*</sup>

<sup>1</sup> Faculty of Integrated Technologies, Universiti Brunei Darussalam, Jalan Tungku Link, Gadong BE1410, Brunei

<sup>2</sup> Department of Electromedical Engineering, Health Polytechnic Ministry of Health, Pucang Jajar Timur No. 10, Surabaya 60282, Indonesia; triwiyanto@ieee.org

<sup>3</sup> Laboratory for Advanced Materials Processing, Empa—Swiss Federal Laboratories for Materials Science & Technology, Feuerwerkerstrasse 39, 3602 Thun, Switzerland; Vigneashwara.SolaiRajaPandiyan@empa.ch

<sup>4</sup> Department of Automatic Control and Robotics, Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering, AGH University of Science and Technology, al. A. Mickiewicza 30, 30-059 Kraków, Poland

<sup>5</sup> Department of Informatics, Faculty of Creative Industry and Telematics, Universitas Trilogi, Jakarta 12760, Indonesia; handy@trilogi.ac.id

<sup>6</sup> School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639815, Singapore

<sup>7</sup> Department of Mechanical Engineering, De Nayer Campus, KU Leuven, Jan Pieter de Nayerlaan 5, 2860 Sint-Katelijne-Waver, Belgium

\* Correspondence: wahyu.caesarendra@ubd.edu.bn (W.C.); adglow@agh.edu.pl (A.G.); tegoeh.tjahjowidodo@kuleuven.be (T.T.)



**Citation:** Caesarendra, W.; Triwiyanto, T.; Pandiyan, V.; Glowacz, A.; Permana, S.D.H.; Tjahjowidodo, T. A CNN Prediction Method for Belt Grinding Tool Wear in a Polishing Process Utilizing 3-Axes Force and Vibration Data. *Electronics* **2021**, *10*, 1429. <https://doi.org/10.3390/electronics10121429>

Academic Editors: Zehui Mao, Xinggang Yan, Hamed Badihi and Dimitris Apostolou

Received: 3 May 2021  
Accepted: 9 June 2021  
Published: 14 June 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Abstract:** This paper presents a tool wear monitoring methodology on the abrasive belt grinding process using vibration and force signatures on a convolutional neural network (CNN). A belt tool typically has a random orientation of abrasive grains and grit size variation for coarse or fine material removal. Degradation of the belt condition is a critical phenomenon that affects the workpiece quality during grinding. This work focuses on the identification and the study of force and vibrational signals taken from sensors along an axis or combination of axes that carry important information of the contact conditions, i.e., belt wear. Three axes of the two sensors are aligned and labelled as X-axis (parallel to the direction of the tool during the abrasive process), Y-axis (perpendicular to the direction of the tool during the abrasive process) and Z-axis (parallel to the direction of the tool during the retract movement). The grinding process was performed using a customized abrasive belt grinder attached to a multi-axis robot on a mild-steel workpiece. The vibration and force signals along three axes (X, Y and Z) were acquired for four discrete sequential belt wear conditions: brand-new, 5-min cycle time, 15-min cycle time, and worn-out. The raw signals that correspond to the sensor measurement along the different axes were used to supervisedly train a 10-Layer CNN architecture to distinguish the belt wear states. Different possible combinations within the three axes of the sensors (X, Y, Z, XY, XZ, YZ and XYZ) were fed as inputs to the CNN model to sort the axis (or combination of axes) in the order of distinct representation of the belt wear state. The CNN classification results revealed that the combination of the XZ-axes and YZ-axes of the accelerometer sensor provides more accurate predictions than other combinations, indicating that the information from the Z-axis of the accelerometer is significant compared to the other two axes. In addition, the CNN accuracy of the XY-axes combination of dynamometer outperformed that of other combinations.

**Keywords:** convolutional neural network; manufacturing; prediction; grinding process; vibration and force signal

## 1. Introduction

In manufacturing industries, the requirements for high-quality precision parts with complex geometries have been increasing rapidly [1–3]. A workpiece generally goes through various stages such as machining, finishing and polishing to attain prescribed design specification and tolerance [4]. The development of science technology, modern control theory, and advanced machining techniques using multi-axis arm robot has enabled automated finishing and polishing for curved surfaces to achieve uniform quality.

The quality of surface finish for a complex geometry part in manufacturing processes, e.g., grinding and polishing, depends primarily on two main variables, i.e., (1) the condition of belt grinding tool and (2) the combination of operating parameters such as cutting speed, force, feed rate, polymer wheel hardness and grit size. Monitoring the condition of the belt grinding tool is necessary because if the tool deterioration is undetected, it will affect the material removal mechanism and ultimately affect the workpiece's surface quality. In addition, the polishing process parameters are also an important aspect as they must be adaptable during the manufacturing process depending on certain scenarios or condition such as (1) the curvature shape of the workpiece, (2) the changes in abrasive tool wear condition, (3) the hardness of the workpiece, which is generally not fully uniform and (4) the area of the workpiece being manufacture, e.g., at the edges or toward the part center. This study focuses on predicting belt grinding tool condition from brand-new to wear with certain manufacturing parameters.

Several studies on tool condition monitoring found in manufacturing process literature mainly concentrate on the cutting tool and the milling process [5–9], while those on the belt grinding tool condition and prediction are still limited [10]. The abrasive belt grinding is a modification of the traditional rigid grinding process. However, the advantage of belt grinding over traditional grinding lies in its ease to uniformly machine any intricate geometries for workpieces [11,12]. The polymer wheel backing of the belt grinding tool on which the belt embedded with abrasive grains rests enables conformability with the intricate surfaces [13]. Due to the nature of the polymer backing's compliant, the process is highly nonlinear and dependent on the process parameters [11,14]. The belt grinding tool consists of three primary parts, namely the driving mechanism, the polymer wheel and the belt itself. The driving mechanism generally consists of at least two wheels in which the belt rotates. The compliance depends primarily on the type and contour of the backing material, and finally, the belt grit size and grain composition define the longevity and the material removal characteristics. Apart from the three primary components, the operating style, such as the applied force, the depth of cut, feeding direction, lubrication, the angle of feed etc., also affects the process dynamics [15].

Out of all the process parameters, only the belt degradation over the cycle time cannot be controlled. The degradation of the belt happens with the abrasive grains withering out or by the grains degenerating resulting in loss of performance of the tool [16]. Several approaches have been proposed on monitoring tool states in the grinding process using the sensor data acquired during the process and state of the art machine learning (ML) algorithm [17–20]. Apart from the tool wear, Deep Learning (DL) based auto-encoder architecture has been used to perform pixel-level classification of weld seam/bead states [20]. Spectrograms computed from the sound signals with DL method have been used to classify wear states of the abrasive belt [10]. A multi-sensor fusion method of vision and sound have been used along with a light gradient boosting machine (LightGBM) algorithm to monitor in-process grinding material removal rate (MRR) [21,22].

The advancement of the manufacturing process equipped with an intelligent method opens the possibilities to answer the existing challenges. An alternate monitoring strategy using CNN and signals input from the accelerometer and force sensor is proposed in this study for monitoring the belt states. To apply ML and DL methods in tool condition monitoring, the inputs representing the dynamic condition between the abrasive grinding tool and the workpiece play an important role. It has been studied that the appearance of vibration, force and torque variable in the polishing process can be determined as the

indicator of the tool wear condition [17]. To date, most of tool wear condition monitoring methods in abrasive processes, such as grinding and polishing, rely on human-interference inspection and is typically conducted as an offline measurement exercise. The offline measurements consequently will interrupt the entire grinding and polishing process due to dismounting and re-mounting procedures of the working coupon to its reference point. This interruption leads to disruptions in the production line. This paper aims to open the possibility of using the DL method for belt tool condition prediction and the potential implementation towards online monitoring systems.

CNN is one of the DL algorithms based on the convolution sliding kernel approach [23]. CNN is usually implemented for classification and prediction methods to avoid unnecessary intermediate step of computing sparse representation such as time, frequency and time-frequency features as they can handle raw data with minimum pre-processing. The CNN algorithm has also been applied in 1-dimensional (1-D) signal, e.g., audio and vibration signal, and 2-dimensional (2-D) data, e.g., images. More details of CNN algorithm is described in Section 2. Specific to the application of CNN in the manufacturing area, especially in tool condition monitoring, CNN has been applied in tool condition monitoring such as for end milling [7]. The previous work of the authors in [13] presents the application of analysis of variance (ANOVA) combined with an adaptive neuro-fuzzy inference system (ANFIS) to model material removal. The combined ANOVA and ANFIS method is used to obtain the optimum configuration between the process parameters/variables of abrasive belt grinding such as RPM, feed, force, rubber hardness and grit size and the stock material removal rate. In the present work, the CNN method is applied to the vibration and force signal from a three-axis accelerometer and a three-axis dynamometer collected during the polishing process to monitor the belt degradation. A new belt grinding tool was used in the polishing process of the mild-steel workpiece with certain manufacturing parameters until wear. The workpiece's polishing process was performed using a customized abrasive belt grinding attached to a multi-axis arm robot. The vibration and force signals are acquired from the 4 discrete sequential conditions of the belt grinding tool from new to wear (brand-new, 5-min cycle time, 15-min cycle time, worn-out) during the polishing process of the mild-steel workpiece. The '5-min cycle time' and '15-min cycle time' mean that the belt is prepared by polishing continuously for 5 and 15 minutes, respectively. This study also aims to correlate the vibration and force sensor axes directions with the belt grinding condition.

The paper is organized into 5 Sections. Section 1 briefly reviews the belt degradation, its influence on surface quality, and the real-time monitoring strategies. Section 2 gives a brief overview of the CNN architecture used for this work. Section 3 introduces the robotized belt grinding experimental setup, processing parameters and data acquisition setup. Section 4 presents and discusses the CNN classification results on the sensor data. Finally, Section 5 summarizes this investigation's findings and future works.

## 2. Materials and Methods

### 2.1. Convolutional Neural Network (CNN)

CNN is a feed-forward neural network inspired by the brain's visual cortex and specializes in processing data with a grid or a sequential structure [24]. CNN is specifically designed for handling multiple arrays as the input. This initial input process of CNN is analogous to imitating the eyes to identify images, followed by a training process for further recognition of the scene [25]. However, to predict well on tricky scenarios, in many cases CNN must be designed with a more complicated architecture. Even though CNN takes higher computational time and hardware resources compared to traditional ML methods (support vector machine (SVM), artificial neural network (ANN) and adaptive neuro-fuzzy inference system (ANFIS) etc.), they are very efficient in processing raw data with minimum pre-processing.

CNN's are well known as a first applied method for tool wear prediction compared to other DL methods such as recurrent neural network (RNN), gated neural network (GNN) and long short-term memory (LSTM) [10,11,21,26]. CNN can have different architectures,

and it depends based on the problem at hand. The model might have different layers such as VGG-16 (16 layers), LeNet, googleNet (22 layers), AlexNet (8 layers), ResNet (152 layers), etc., and also can use different transformation such as ReLu (rectified linear unit), Dropout and batch normalization. The parameters used for the model training are primarily chosen in such a way as to reduce computational cost and hardware utilization. CNN training parameters are chosen based on exhaustive search or on trial and error [27,28].

The first modern CNN used by Tavakkoli et al. [29] has a 7-layer structure (excluding the input layer), namely LeNet-5, which has the following structures C1, S2, C3, S4, C5 and F6 output. In addition, Zhang et al. [27] reported that several layers in a CNN model are to be considered for better performance. The following layers on CNN considered in this work following [27], are as follows:

- The sub-sampling layer is usually used to reduce the dimensions coming from the input. The sub-sampling layer’s goal is to reduce the number of trainable parameters needed on CNN, which can speed up the performance of CNN. The speed means computational time. By optimizing the CNN architecture, the accuracy and computational time cost could be obtained as fast as possible. The speed also means using fewer resources in identifying images.
- The convolutional layer works by imitating the properties of the visual cortex of the brain for studying the features from input images. Filters or kernels trained at this layer can identify specific shapes. For example, if it is used to study an image, the filter might learn for edge detection. The following is the equation in the convolution layer:

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \tag{1}$$

Equation (1) is a formula to calculate the image filter for each pixel. A detailed illustration is presented in Figures 1–5, and a summary of all convolution process is shown in Figure 6.

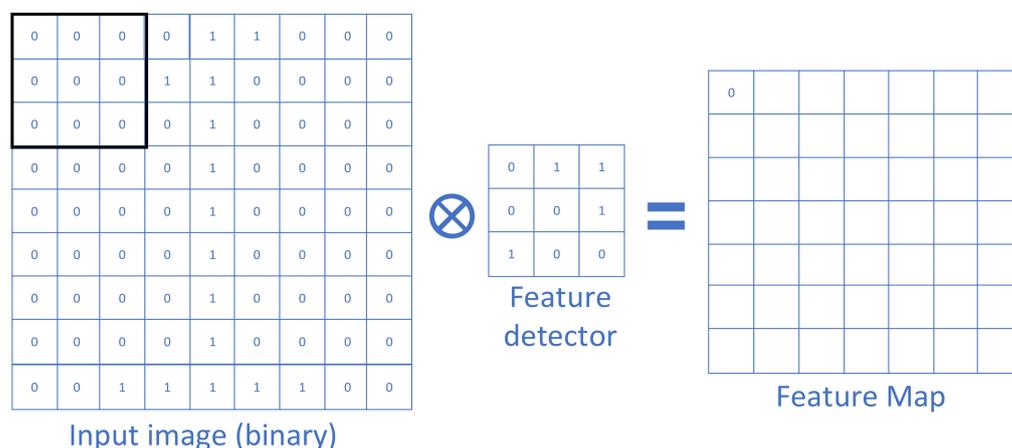


Figure 1. Binary conversion of feature map 1.

- The loss layer is the output of CNN, which is an important part of the neural network. This function calculates loss that corresponds to the difference between the model output and the original ground truth. The magnitude of the loss value determines the rate of gradient change and weights updation during backpropagation.
- The fully connected layer is found in most neural networks. Fully connected layers use matrix multiplication to get the output from that layer.
- ReLU layer is used for thresholding or similar to the activation function on neural network. They bring in non-linearity to the model. The following is the equation in the ReLU layer  $\max(0,x)$  function:

$$g(x) = \text{ReLU}(x) = \begin{cases} x, & x \geq 0 \\ 0, & x \leq 0 \end{cases} \tag{2}$$

$$g'(x) = \text{ReLU}'(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x \leq 0 \end{cases} \tag{3}$$

At this stage, the map feature results will be fed into the search function for the highest value between  $x$  and 0. The architecture of the ReLU layer process is presented in Figure 7.

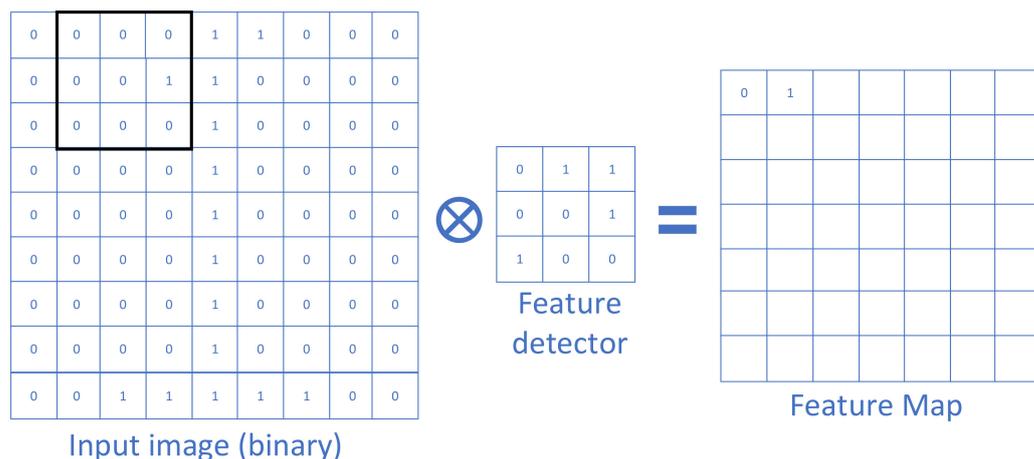


Figure 2. Binary conversion of feature map 2.

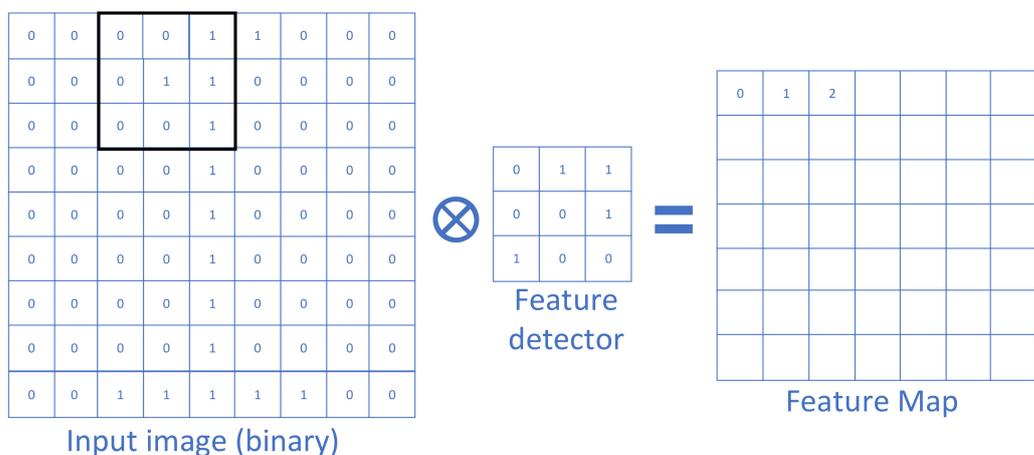


Figure 3. Binary conversion of feature map 3.

- The pooling layer is very similar to the convolutional layer. This operation in the pooling layer is essential for reducing the spatial size of the feature map. To reduce computational cost, a dimension reduction process is necessary. This layer is useful for extracting domain features that are rotational and invariant so that the model training process becomes more effective. The pooling operations are of many types based on model performance, namely maximum, average, minimum pooling, etc. An example of the max-pooling process of feature map 1 to feature map 9 is presented in Figure 8.
- Figure 9 shows how a 2D input image transforms when it passes through series of a convolutional layer to produce a feature map. The pooling layer is generally used to reduce the feature map size between convolutions. The pooling method that is often used in CNN is the max pooling method. In very rare cases averaging pooling is also used.

- The flattening layer changes a matrix data form (2D and 3D) into a one-dimensional array to enter the next layer, typically before the output layer or SoftMax activation. This layer places all pixel data in one row and makes a connection with the last layer. An illustration of a  $3 \times 3$  matrix conversion to a one-dimensional matrix is illustrated in Figure 10.

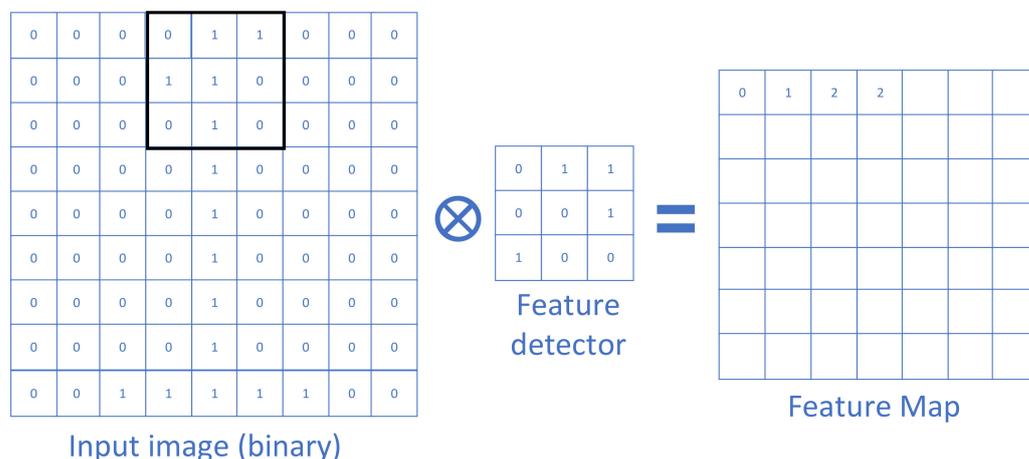


Figure 4. Binary conversion of feature map 4.

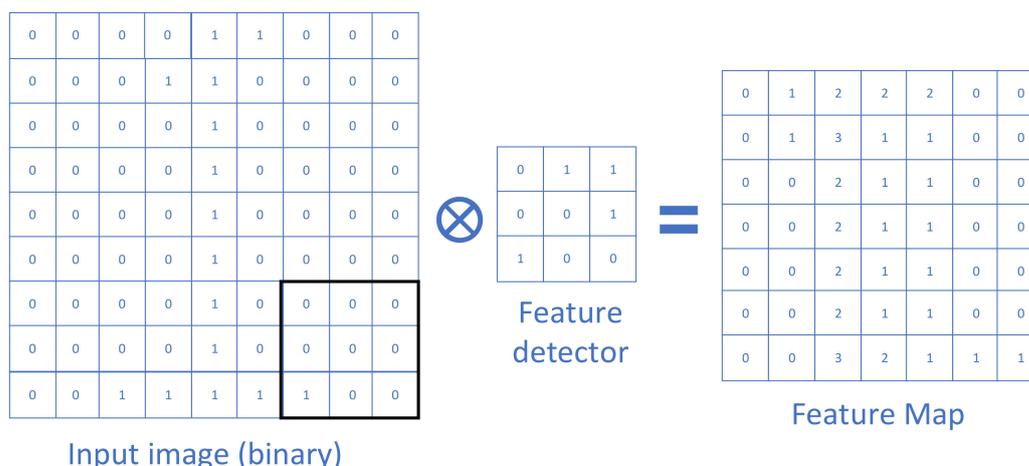


Figure 5. Binary conversion of feature map 25.

The CNN architecture discussed in this section is suitable for solving image classification, object detection, object location and image segmentation problems. Examples of case studies in image recognition using CNN typically have three stages: the input, CNN and output stages. A detailed explanation of the stages of CNN is presented in [30–32]. Figure 11 shows the stages carried out by CNN from the input image, convolutional layer, pooling layer and flattening layer.

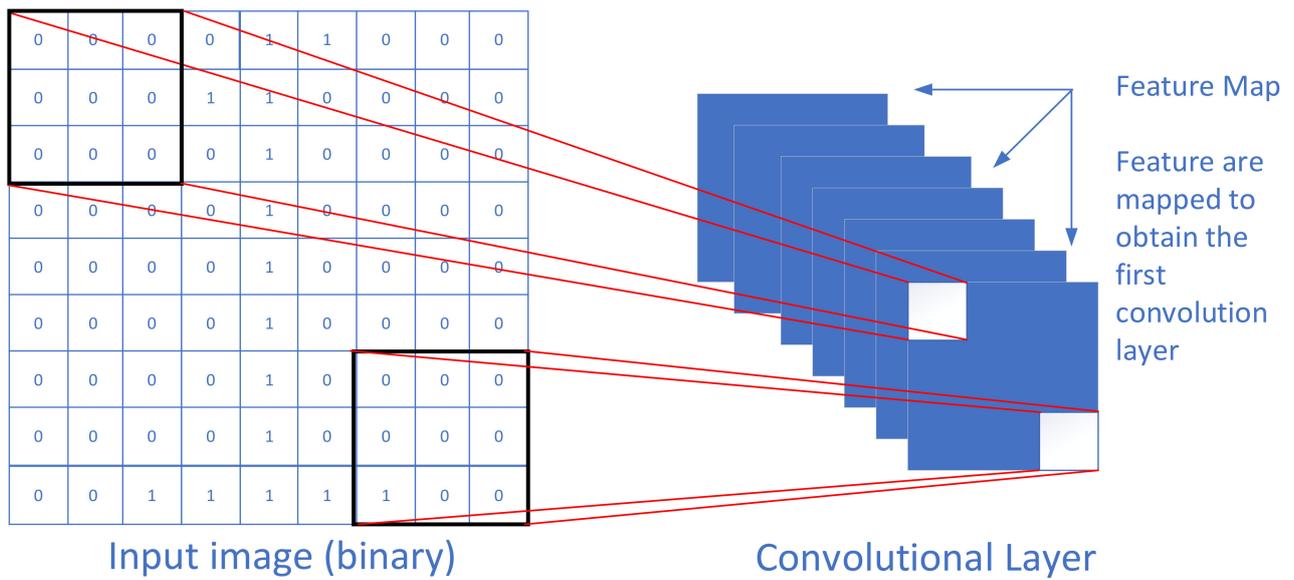


Figure 6. The architecture of convolutional layer (binary conversion) from feature map 1 to 25.

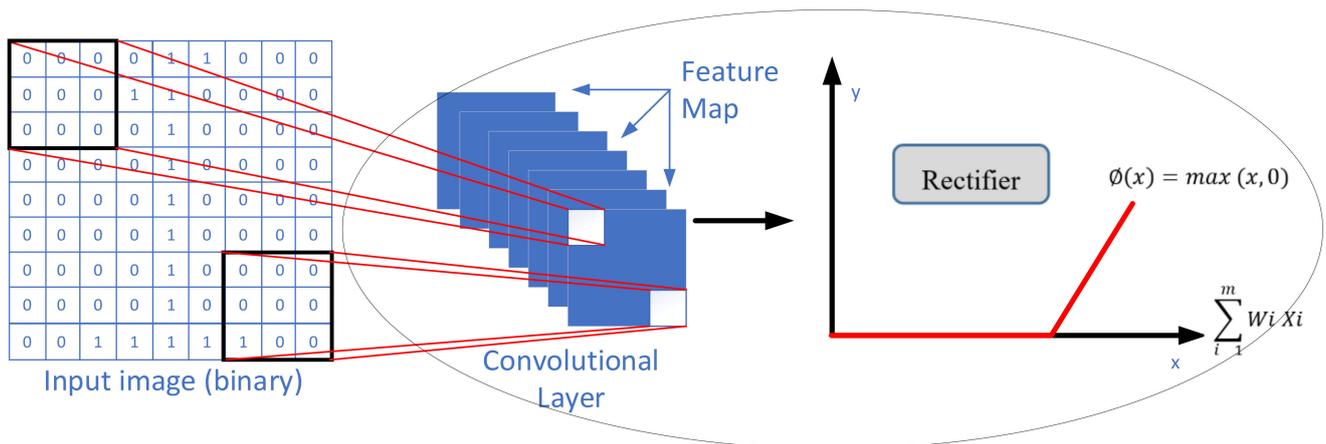


Figure 7. ReLU layer function applied in convolutional layer process.

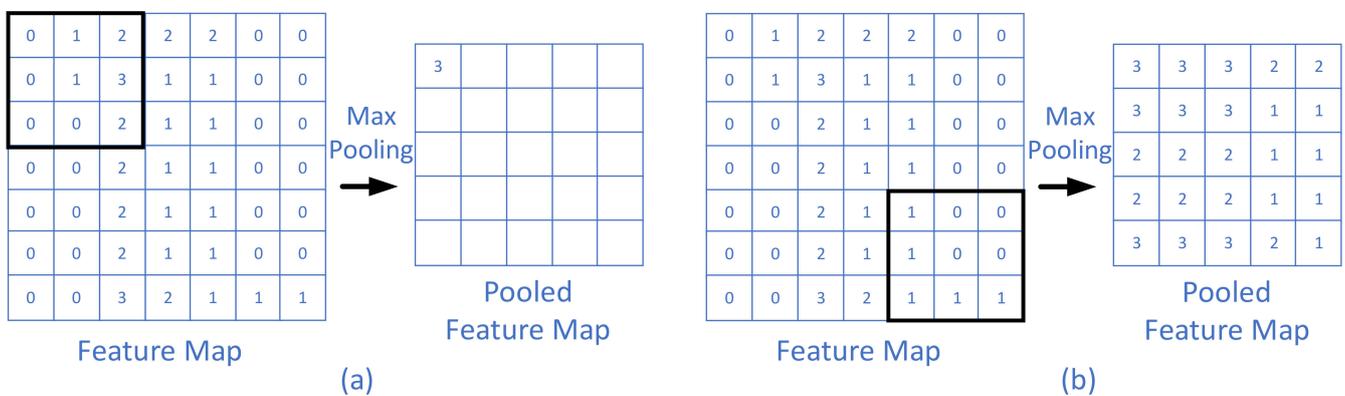


Figure 8. Max pooling: (a) feature map 1; (b) feature map 9.

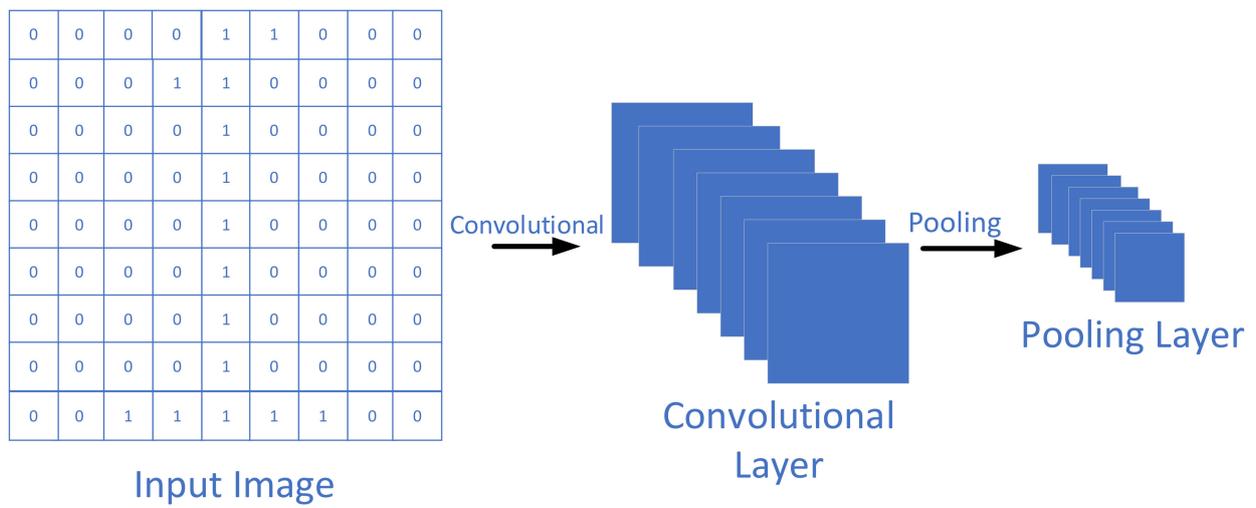


Figure 9. The architecture of the max-pooling layer.

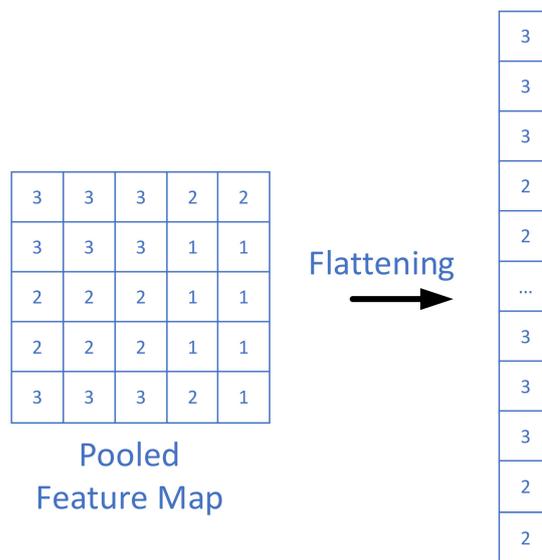


Figure 10. The architecture of flattening layer.

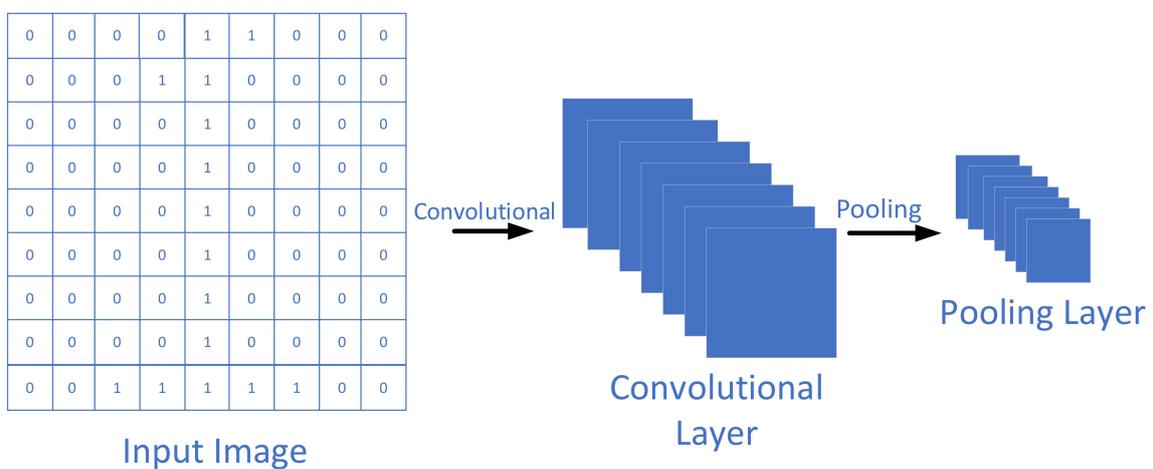
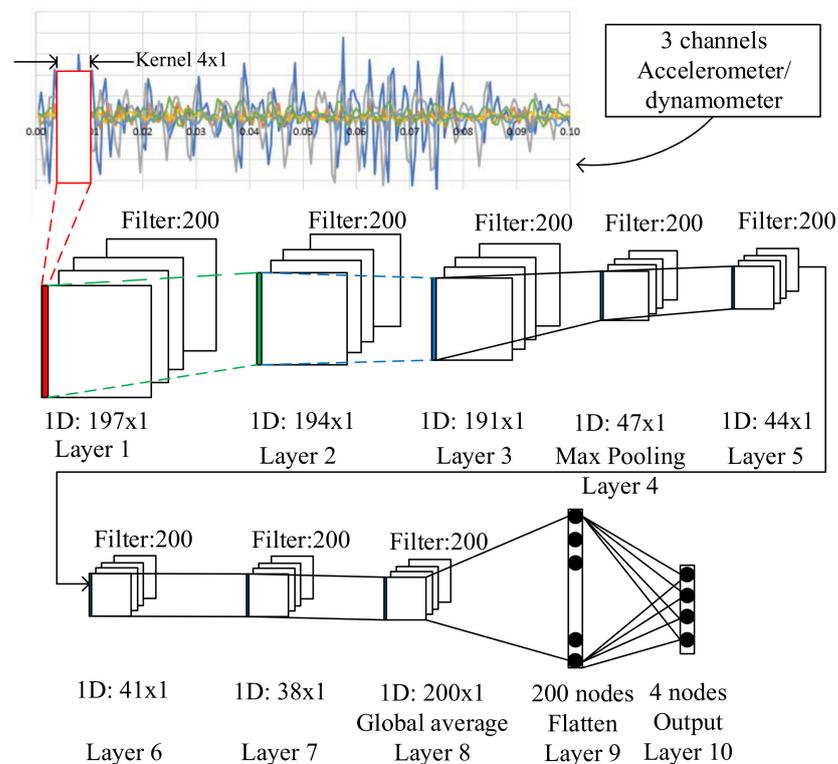


Figure 11. Step-by-step process of CNN from convolutional layer, pooling layer to flattening layer.

## 2.2. CNN Structure Used in the Present Study

The structure of CNN varies depending on the application, for example, the structure presented in [33]. The details of the CNN architecture to classify belt wear states is presented in Figure 12. The CNN architecture consists of 10 layers, namely 6 one dimensional (1D) convolution layers, 1 max-pooling layer, 1 global average pooling layer, 1 drop layer and 1 fully connected layer. Furthermore, each 1D convolution layer constitutes 200 filters, in which the filter weights have been initialized with random values before training. Additionally, the kernel length of size 4 is used in the convolution process between the signal and filter. At the first stage, the CNN receives the raw time-series 1D signals from a three-axis dynamometer sensor (X, Y and Z) and a three-axis accelerometer sensor (X, Y and Z). In the first layer (L1), each raw signal (X, Y and Z) from the dynamometer and accelerometer sensors are convolved with the filter kernels. Subsequently, the results of the convolutions at each layer are used as input to the next layers. In the fourth layer, to reduce the number of inputs, a pooling layer is applied. In this case, the max-pooling is taken. In the next three layers (L5, L6 and L7), the values from the max-pooling layer are convolved with the filter kernels. In the eighth layer (L8), in order to reduce the number of parameters, the global average pooling layer is applied. In this case, all of the values from each filter were averaged, which results in only one value and turns the total number of the parameter to 1200 (6 inputs  $\times$  200 filters). In the ninth layer, a dropping layer is applied to prevent overfitting in the training stage. Finally, all of the nodes from the drop layer is fully connected. In layer 4 (L4), the max-pooling layer was chosen because it helps to have a high contrast feature map of the resulting convolution in the middle of the feature engineering process. On the contrary, average pooling was chosen in the last convolution (L8) to smoothen the feature maps.



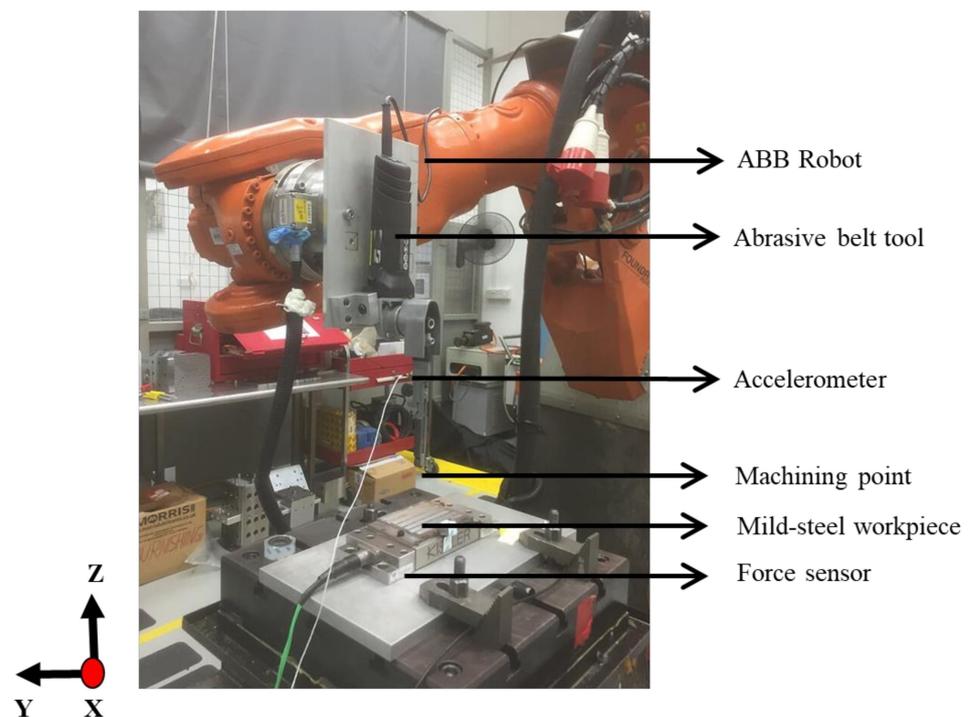
**Figure 12.** CNN model and architecture for 3-axis acceleration and 3-axis force data training and prediction.

## 3. Experimental Setup

This study employs a 3-axis accelerometer and 3-axis dynamometer sensor signature to predict the belt wear that affects the polishing quality. The following subsection presents in detail the experimental polishing setup and data acquisition procedure:

### 3.1. Polishing Process Equipment, Sensors and Data Acquisition

The experiments were performed using a multi-axis arm robot mounted with an abrasive belt grinding setup, as presented in Figure 13. The experimental setup consisted of ABB 6660-205-193 robot that was primarily used for imparting motion in the grinding direction. The belt grinder used in this work was electrically powered that can run at 11,000 RPM at no-load condition. The belt grinder could polish and grind with belts about 8" to 3/4" wide  $\times$  18" long. The belt grinder was coupled to a force control present in the end effector of the ABB robot with the help of a customized bracket. The force control was closed-looped into the robot controller to ensure the force input from the operator and imparted force in the normal direction (Z-axis) are equal. Normal direction is usually preferred to achieve uniform material removal. As far as the experimental trials, a force of 20 N was applied throughout this study. The experimental conditions used in this work are listed in Table 1. Four different belt condition representing four different classes of wear level in this study were prepared beforehand by performing actual grinding processes at different cycle times, as depicted in Table 2. During the polishing experiments, one variable output process parameter was the belt wear condition (four levels). Other process parameters such as feed rate, grinding speed and force control were maintained at a constant value, as presented in Table 1.



**Figure 13.** A customized abrasive belt-grinding experiment and sensors position (accelerometer and dynamometer).

**Table 1.** Experimental condition.

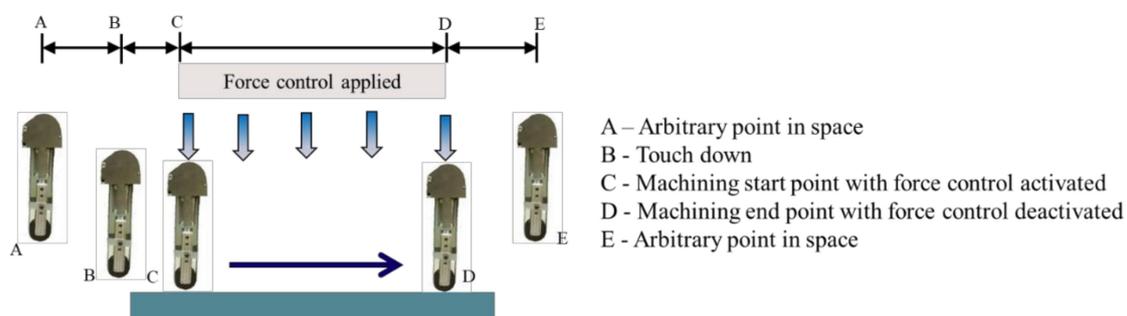
Parameters	Description
Belt type	Aluminium oxide with grit size 60
Force control	Constant force of 20 N using ATI force sensor
Grinding speed	Constant at feed rate of 50 mm/s
Workpiece	Mild steel with uniform surface roughness
Contact wheel	Shore A Hardness 80

**Table 2.** Four states of abrasive belt under monitoring.

Belt No.	Abrasive Belt Grinding Label	Usage Time (min)
1	Brand-new belt	0
2	5-min cycle time belt	5
3	15-min cycle time belt	15
4	Worn-out belt	30

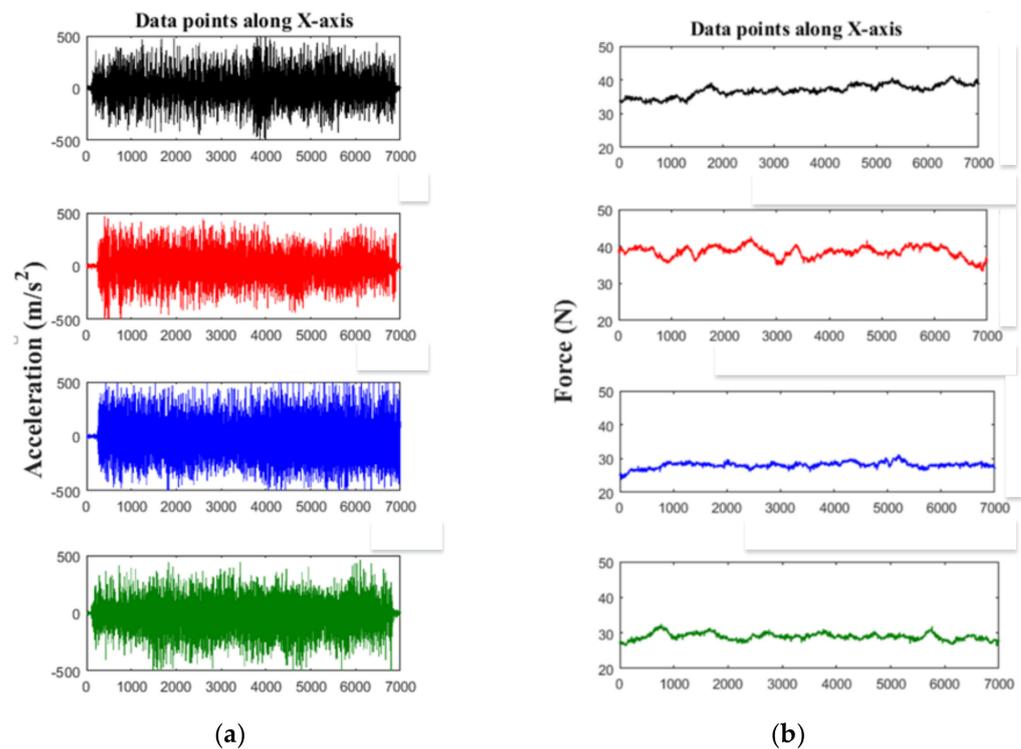
Kistler 8763A500 triaxial accelerometer sensor was attached near the tension arm of the electric belt grinder to obtain the vibration signal during polishing (i.e., contact between belt tool and workpiece), as presented in Figure 13. In addition, Kistler 9254 three-component dynamometer was placed below the mild-steel workpiece, as shown in Figure 13. NI data acquisition (DAQ) module and LabVIEW environment were used to acquire vibration signal in a digitized format with a sampling frequency of 2 kHz. In the case of the dynamometer, a DAQ device supported by the DEWESoft platform was used to acquire force signals across normal and tangential directions. The signals corresponding to the force and accelerometer sensors were synchronized offline.

ABB Robot studio was used to configure and was planned for the tool path. The tool path planning for the linear run performed in this work has five different zones, A–E, as presented in Figure 14. The region between A–B was used to align the tool head over the machining region. Zone B–C was used to attain the force required for grinding. C–D was the zone where force control is actively applied. Finally, Zone D–E was used to decelerate the force sensor and robot motion. The whole experiments performed in this study were carried out in a non-lubricated condition on a mild steel workpiece.

**Figure 14.** Abrasive tool path on the mild-steel workpiece.

### 3.2. Vibration and Force Data Acquisition

This paper is an extension work of the previous study that applied ML methods to sensor signatures [17,18]. The difference to the work reported by Pandiyan et al. [17] is that they used short-time Fourier transform (STFT) on acoustic emissions signal to detect the change in contact mechanisms caused by tool wear in abrasive belt grinding. However, in the present study, CNN methods are developed to predict the wear level of the belt tool using 3-axes accelerometer and 3-axes dynamometer signatures instead of the acoustic data. All accelerometer and force sparse features from the 3 axes were used as input data in the ML method [18]. However, extracting features and feeding them into the ML algorithms is a two-step procedure and is computationally expensive. In this study, the CNN algorithm evaluates which axis/axes of the sensor signature(s) are more relevant for classifying the abrasive belt wear state using raw sensor signals. An example of raw vibration and force signal from the X-axis sensor direction for different belt tool states, i.e., brand-new, 5-min cycle time, 15-min cycle time and worn-out (from top to bottom) is presented in Figure 15a,b. For interested readers, a complete raw signals plot of Figure 15 is presented in [18].



**Figure 15.** Raw signals of x-axis sensor direction: (a) vibration and (b) force.

The raw vibration and force signals dataset were acquired with a sampling frequency of 2 kHz for 3.5 s, resulting in 7000 data points as presented in Figure 15. Prior to the convolution process, the raw datasets were segmented into a smaller data form. The raw datasets were reshaped to 200 data points with 10% overlaps. The new 200 data points were equivalent to a window of 0.1 sec of the sampling frequency of the polishing process. According to previous works [18], 200 data points can still capture the polishing process's dynamic behavior. The pre-processing of the raw dataset produced 64,000 new rows of datasets for each class and 256,000 new datasets for all four classes combined, i.e., brand-new, 5-min cycle time, 15-min cycle time and worn-out. These datasets were then divided into two sets for training and validation. The CNN model computation used 80% of 256,000 datasets (192,000 datasets) as the training set and another 20% of 256,000 datasets (64,000 datasets) for validation.

A sample of data input signals for the CNN algorithm from three directions (X, Y and Z-axes) of the accelerometer is presented in Figure 16. Based on the visualization of the vibration signals corresponding to four different belt conditions, it is evident that distinguishing them is challenging. An example of force signals from three directions (X, Y and Z-axes) of the dynamometer is presented in Figure 17. Synonymous to the vibration signals, differentiating belt conditions visually is also challenging. A detailed illustration of the data input preparation for the CNN method is presented in Figure 18. As presented in Figure 18, raw vibration and force data were collected from three axes. The raw data was then divided further into a smaller sized window for CNN training.

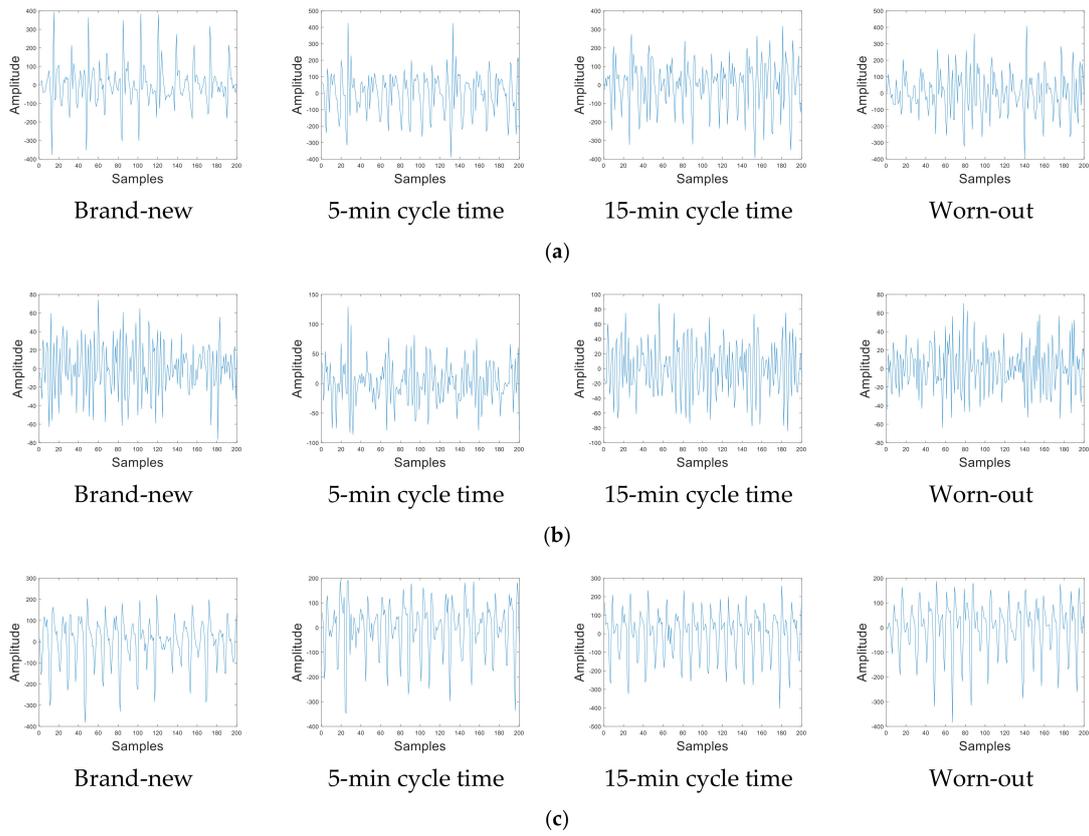


Figure 16. Vibration signals: (a) X-axis, (b) Y-axis and (c) Z-axis.

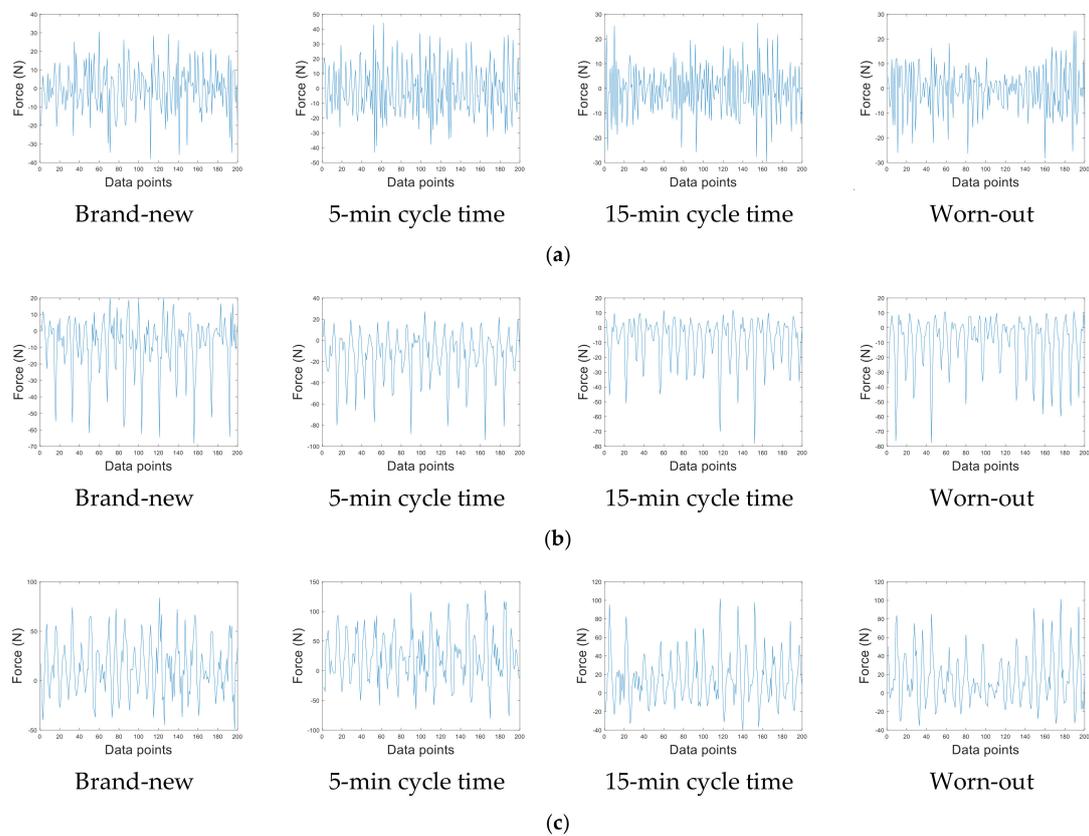
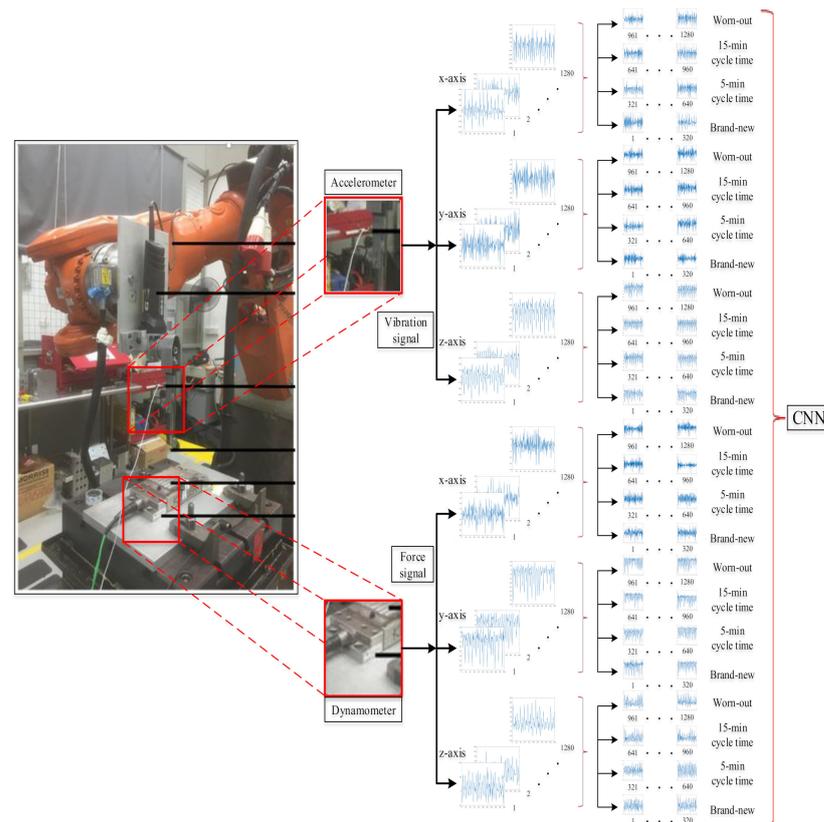


Figure 17. Force signals: (a) X-axis, (b) Y-axis and (c) Z-axis.



**Figure 18.** Data input preparation for CNN algorithm. The data consist of 3-axes vibration and 3-axes force signal.

## 4. Results and Discussion

### 4.1. CNN Results of Vibration Data

This sub-section provides the CNN classification results of vibration signals from three categories: (1) single-axis accelerometer (X, Y and Z), (2) double-axes accelerometer (XY, XZ and YZ) and (3) triple-axes (XYZ).

This study is based on the CNN design, as illustrated in Figure 12 on the accelerometer sensor data. To provide a better understanding for the classification labels in Table 3 confusion matrix results in Tables 4–6 and Figures 19 and 20, they are renamed as follows (as presented in Table 2):

- Brand-new as Belt No.1.
- 5-min cycle time as Belt No.2.
- 15-min cycle time as Belt No.3.
- Worn-out as Belt No.4.

#### 4.1.1. Vibration Signal of Single-Axis Accelerometer (X, Y and Z)

As depicted in Table 3a, the CNN classification result in the X-axis indicates that all 796 datasets of the “brand-new” class can be classified correctly without misclassification. In the “5-min cycle time” class, 796 datasets were classified correctly. However, 2 datasets were misclassified as “15-min cycle time”, and other 2 datasets were misclassified as “worn-out”. In the “15-min cycle time” class, all 798 datasets were also successfully classified. For the “worn-out” class, 793 datasets were classified correctly, and only 1 dataset was identified as a “15-min cycle time” class.

Additionally, Table 3b shows that 795 datasets of the “brand-new” class corresponding to the Y-axis can be classified correctly, and 1 dataset was misclassified as a “5-min cycle time” class. In the “5-min cycle time” class, 798 datasets were classified correctly. However, 2 datasets were extremely misclassified as “worn-out” class. In the “15-min cycle time” class,

717 datasets were classified correctly. However, 83 datasets were classified incorrectly as “worn-out” class. The worst classification occurs in the “worn-out” class, where 670 datasets were classified correctly, but 89 datasets were classified incorrectly as “brand-new” and 16 datasets were identified as “5-min cycle time” class, while other 19 datasets were misclassified to the “15-min cycle time” class.

**Table 3.** Training curves and confusion matrix of single-axis vibration data.

	Model Accuracy and Loss	Confusion Matrix																																	
(a) Vibration data of X-axis		<table border="1"> <tr> <td rowspan="4">True label</td> <td>Belt No. 1</td> <td>796</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Belt No. 2</td> <td>0</td> <td>796</td> <td>2</td> <td>2</td> </tr> <tr> <td>Belt No. 3</td> <td>0</td> <td>0</td> <td>798</td> <td>0</td> </tr> <tr> <td>Belt No. 4</td> <td>0</td> <td>0</td> <td>1</td> <td>793</td> </tr> <tr> <td></td> <td></td> <td>Belt No. 1</td> <td>Belt No. 2</td> <td>Belt No. 3</td> <td>Belt No. 4</td> </tr> <tr> <td colspan="6">Predicted label</td> </tr> </table>	True label	Belt No. 1	796	0	0	0	Belt No. 2	0	796	2	2	Belt No. 3	0	0	798	0	Belt No. 4	0	0	1	793			Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4	Predicted label					
True label	Belt No. 1	796		0	0	0																													
	Belt No. 2	0		796	2	2																													
	Belt No. 3	0		0	798	0																													
	Belt No. 4	0	0	1	793																														
		Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4																														
Predicted label																																			
(b) Vibration data of Y-axis		<table border="1"> <tr> <td rowspan="4">True label</td> <td>Belt No. 1</td> <td>795</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>Belt No. 2</td> <td>0</td> <td>798</td> <td>0</td> <td>2</td> </tr> <tr> <td>Belt No. 3</td> <td>0</td> <td>0</td> <td>717</td> <td>83</td> </tr> <tr> <td>Belt No. 4</td> <td>89</td> <td>16</td> <td>19</td> <td>670</td> </tr> <tr> <td></td> <td></td> <td>Belt No. 1</td> <td>Belt No. 2</td> <td>Belt No. 3</td> <td>Belt No. 4</td> </tr> <tr> <td colspan="6">Predicted label</td> </tr> </table>	True label	Belt No. 1	795	1	0	0	Belt No. 2	0	798	0	2	Belt No. 3	0	0	717	83	Belt No. 4	89	16	19	670			Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4	Predicted label					
True label	Belt No. 1	795		1	0	0																													
	Belt No. 2	0		798	0	2																													
	Belt No. 3	0		0	717	83																													
	Belt No. 4	89	16	19	670																														
		Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4																														
Predicted label																																			
(c) Vibration data of Z-axis		<table border="1"> <tr> <td rowspan="4">True label</td> <td>Belt No. 1</td> <td>790</td> <td>1</td> <td>0</td> <td>5</td> </tr> <tr> <td>Belt No. 2</td> <td>0</td> <td>791</td> <td>3</td> <td>6</td> </tr> <tr> <td>Belt No. 3</td> <td>0</td> <td>0</td> <td>748</td> <td>52</td> </tr> <tr> <td>Belt No. 4</td> <td>2</td> <td>47</td> <td>26</td> <td>719</td> </tr> <tr> <td></td> <td></td> <td>Belt No. 1</td> <td>Belt No. 2</td> <td>Belt No. 3</td> <td>Belt No. 4</td> </tr> <tr> <td colspan="6">Predicted label</td> </tr> </table>	True label	Belt No. 1	790	1	0	5	Belt No. 2	0	791	3	6	Belt No. 3	0	0	748	52	Belt No. 4	2	47	26	719			Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4	Predicted label					
True label	Belt No. 1	790		1	0	5																													
	Belt No. 2	0		791	3	6																													
	Belt No. 3	0		0	748	52																													
	Belt No. 4	2	47	26	719																														
		Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4																														
Predicted label																																			

Note: please see Table 2 for detailed information on the classification label.

Table 3c shows CNN results on Z-axis vibrational signals. 790 datasets of “brand new” class were classified correctly, 1 dataset was incorrectly classified as “5-min cycle time” class, and 5 datasets were misclassified as “worn-out” class. In the “5-min cycle time” class, 791 datasets were classified correctly, 3 datasets were incorrectly classified as “15-min cycle time” class, and 6 datasets were incorrectly classified as “worn-out” class. For the “15-min cycle time” class, 748 datasets were classified correctly, and there were 52 datasets classified incorrectly as “worn-out” class. Similar to the Table 3b result, the poor classification result occurred in the “worn-out” class, where class 2 datasets were misclassified as “brand-new” class, 47 datasets were incorrectly identified as “5-min cycle time”, and 26 datasets were misclassified as “15-min cycle time” class.

Table 3 shows the accuracy and loss curves during CNN training using the accelerometer sensor data from X, Y or Z axes. In Table 3, it can be seen that processing 7672 data for 50 epoch takes 17 to 20 s. When trained with sensors derived from the X-axis of the accelerometer, the accuracy of the CNN model results in higher accuracy (accuracy: 95.78%) than the other axes (Y-axis, accuracy: 87.47%; Z-axis, accuracy: 89.43%).

**Table 4.** Training curves and confusion matrix of double-axis vibration data.

	Model Accuracy and Loss	Confusion Matrix																																	
(a) Vibration data of XY-axis		<table border="1"> <tr> <td rowspan="4">True label</td> <td>Belt No. 1</td> <td>796</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Belt No. 2</td> <td>1</td> <td>797</td> <td>2</td> <td>0</td> </tr> <tr> <td>Belt No. 3</td> <td>0</td> <td>0</td> <td>800</td> <td>0</td> </tr> <tr> <td>Belt No. 4</td> <td>0</td> <td>0</td> <td>6</td> <td>788</td> </tr> <tr> <td></td> <td></td> <td>Belt No. 1</td> <td>Belt No. 2</td> <td>Belt No. 3</td> <td>Belt No. 4</td> </tr> <tr> <td></td> <td></td> <td colspan="4">Predicted label</td> </tr> </table>	True label	Belt No. 1	796	0	0	0	Belt No. 2	1	797	2	0	Belt No. 3	0	0	800	0	Belt No. 4	0	0	6	788			Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4			Predicted label			
True label	Belt No. 1	796		0	0	0																													
	Belt No. 2	1		797	2	0																													
	Belt No. 3	0		0	800	0																													
	Belt No. 4	0	0	6	788																														
		Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4																														
		Predicted label																																	
(b) Vibration data of XZ-axis		<table border="1"> <tr> <td rowspan="4">True label</td> <td>Belt No. 1</td> <td>796</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Belt No. 2</td> <td>0</td> <td>798</td> <td>0</td> <td>0</td> </tr> <tr> <td>Belt No. 3</td> <td>0</td> <td>0</td> <td>800</td> <td>0</td> </tr> <tr> <td>Belt No. 4</td> <td>0</td> <td>0</td> <td>0</td> <td>794</td> </tr> <tr> <td></td> <td></td> <td>Belt No. 1</td> <td>Belt No. 2</td> <td>Belt No. 3</td> <td>Belt No. 4</td> </tr> <tr> <td></td> <td></td> <td colspan="4">Predicted label</td> </tr> </table>	True label	Belt No. 1	796	0	0	0	Belt No. 2	0	798	0	0	Belt No. 3	0	0	800	0	Belt No. 4	0	0	0	794			Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4			Predicted label			
True label	Belt No. 1	796		0	0	0																													
	Belt No. 2	0		798	0	0																													
	Belt No. 3	0		0	800	0																													
	Belt No. 4	0	0	0	794																														
		Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4																														
		Predicted label																																	
(c) Vibration data of YZ-axis		<table border="1"> <tr> <td rowspan="4">True label</td> <td>Belt No. 1</td> <td>790</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Belt No. 2</td> <td>0</td> <td>791</td> <td>1</td> <td>1</td> </tr> <tr> <td>Belt No. 3</td> <td>0</td> <td>0</td> <td>800</td> <td>0</td> </tr> <tr> <td>Belt No. 4</td> <td>0</td> <td>0</td> <td>0</td> <td>794</td> </tr> <tr> <td></td> <td></td> <td>Belt No. 1</td> <td>Belt No. 2</td> <td>Belt No. 3</td> <td>Belt No. 4</td> </tr> <tr> <td></td> <td></td> <td colspan="4">Predicted label</td> </tr> </table>	True label	Belt No. 1	790	0	0	0	Belt No. 2	0	791	1	1	Belt No. 3	0	0	800	0	Belt No. 4	0	0	0	794			Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4			Predicted label			
True label	Belt No. 1	790		0	0	0																													
	Belt No. 2	0		791	1	1																													
	Belt No. 3	0		0	800	0																													
	Belt No. 4	0	0	0	794																														
		Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4																														
		Predicted label																																	

Note: please see Table 2 for detailed information on the classification label.

#### 4.1.2. Vibration Signal of Double-Axis Accelerometer (XY, XZ and YZ)

The sensor data from two axes were combined to examine the dynamic behavior of the abrasive process. This combined signal is inputted into the CNN algorithm. A confusion matrix from the CNN model that used two-axis (XY, XZ and YZ) from the accelerometer sensor is presented in Table 4. Table 4b indicates that the result across XZ-axis shows higher accuracy with the CNN compared to other combination. Table 4a indicates that all 796 datasets of “brand-new” can be classified appropriately. Additionally, in the “5-min cycle time” class, 797 datasets were classified correctly, 1 dataset was misclassified as “brand-new”, and 2 datasets were misclassified as “15-min cycle time”. For the “15-min cycle time” class, all 800 datasets were classified correctly. In the “worn-out” class,

788 datasets were classified correctly, and another 6 datasets were shifted to the “15-min cycle time” class.

Table 4 shows the accuracy and loss curves during CNN training using two inputs from sensors with a combination of XY, XZ or YZ inputs. In Table 4, it can be seen that processing 7672 data with 50 epoch takes 17 to 20 s. When trained with sensors derived from the XZ-axes of the accelerometer, the accuracy of the CNN model results in higher accuracy (accuracy: 96.40%) than the other axes (XY-axes, accuracy: 95.06%; YZ-axes, accuracy: 94.92%).

**Table 5.** Model accuracy and loss and confusion matrix of single-axis force data.

	Model Accuracy and Loss	Confusion Matrix																																	
(a) Force data of X-axis		<table border="1"> <tr> <td rowspan="4">True label</td> <td>Belt No. 1</td> <td><b>793</b></td> <td>3</td> <td>0</td> <td>0</td> </tr> <tr> <td>Belt No. 2</td> <td>3</td> <td><b>797</b></td> <td>0</td> <td>0</td> </tr> <tr> <td>Belt No. 3</td> <td>1</td> <td>0</td> <td><b>798</b></td> <td>0</td> </tr> <tr> <td>Belt No. 4</td> <td>0</td> <td>0</td> <td>47</td> <td><b>747</b></td> </tr> <tr> <td></td> <td></td> <td>Belt No. 1</td> <td>Belt No. 2</td> <td>Belt No. 3</td> <td>Belt No. 4</td> </tr> <tr> <td></td> <td></td> <td colspan="4">Predicted label</td> </tr> </table>	True label	Belt No. 1	<b>793</b>	3	0	0	Belt No. 2	3	<b>797</b>	0	0	Belt No. 3	1	0	<b>798</b>	0	Belt No. 4	0	0	47	<b>747</b>			Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4			Predicted label			
True label	Belt No. 1	<b>793</b>		3	0	0																													
	Belt No. 2	3		<b>797</b>	0	0																													
	Belt No. 3	1		0	<b>798</b>	0																													
	Belt No. 4	0	0	47	<b>747</b>																														
		Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4																														
		Predicted label																																	
(b) Force data of Y-axis		<table border="1"> <tr> <td rowspan="4">True label</td> <td>Belt No. 1</td> <td><b>794</b></td> <td>2</td> <td>0</td> <td>0</td> </tr> <tr> <td>Belt No. 2</td> <td>0</td> <td><b>796</b></td> <td>0</td> <td>4</td> </tr> <tr> <td>Belt No. 3</td> <td>0</td> <td>0</td> <td><b>771</b></td> <td>29</td> </tr> <tr> <td>Belt No. 4</td> <td>0</td> <td>0</td> <td>30</td> <td><b>764</b></td> </tr> <tr> <td></td> <td></td> <td>Belt No. 1</td> <td>Belt No. 2</td> <td>Belt No. 3</td> <td>Belt No. 4</td> </tr> <tr> <td></td> <td></td> <td colspan="4">Predicted label</td> </tr> </table>	True label	Belt No. 1	<b>794</b>	2	0	0	Belt No. 2	0	<b>796</b>	0	4	Belt No. 3	0	0	<b>771</b>	29	Belt No. 4	0	0	30	<b>764</b>			Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4			Predicted label			
True label	Belt No. 1	<b>794</b>		2	0	0																													
	Belt No. 2	0		<b>796</b>	0	4																													
	Belt No. 3	0		0	<b>771</b>	29																													
	Belt No. 4	0	0	30	<b>764</b>																														
		Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4																														
		Predicted label																																	
(c) Force data of Z-axis		<table border="1"> <tr> <td rowspan="4">True label</td> <td>Belt No. 1</td> <td><b>794</b></td> <td>2</td> <td>0</td> <td>0</td> </tr> <tr> <td>Belt No. 2</td> <td>0</td> <td><b>798</b></td> <td>2</td> <td>0</td> </tr> <tr> <td>Belt No. 3</td> <td>0</td> <td>0</td> <td><b>788</b></td> <td>12</td> </tr> <tr> <td>Belt No. 4</td> <td>0</td> <td>0</td> <td>9</td> <td><b>785</b></td> </tr> <tr> <td></td> <td></td> <td>Belt No. 1</td> <td>Belt No. 2</td> <td>Belt No. 3</td> <td>Belt No. 4</td> </tr> <tr> <td></td> <td></td> <td colspan="4">Predicted label</td> </tr> </table>	True label	Belt No. 1	<b>794</b>	2	0	0	Belt No. 2	0	<b>798</b>	2	0	Belt No. 3	0	0	<b>788</b>	12	Belt No. 4	0	0	9	<b>785</b>			Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4			Predicted label			
True label	Belt No. 1	<b>794</b>		2	0	0																													
	Belt No. 2	0		<b>798</b>	2	0																													
	Belt No. 3	0		0	<b>788</b>	12																													
	Belt No. 4	0	0	9	<b>785</b>																														
		Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4																														
		Predicted label																																	

Note: please see Table 2 for detailed information on the classification label.

#### 4.1.3. Vibration Signal of Triple-Axis Accelerometer (XYZ)

Figure 19 shows the confusion matrix of the CNN model that used the data from the three axes of the accelerometer sensor. Figure 19 indicates that 795 datasets of the “brand-new” class can be classified appropriately, and 1 dataset was misclassified as a “5-min cycle time” class. For the “5-min cycle time” class, 794 datasets were classified correctly, 6 datasets were misclassified to the “15-min cycle time” class. In the “15-min cycle time” class, all 800 datasets were classified correctly. In contrast to the “worn-out”

class, 765 datasets were classified correctly, and 29 datasets were misclassified as “15-min cycle time” class.

Figure 19 shows the accuracy and loss model during CNN training using all accelerometer sensor inputs. Figure 8 shows that processing 7672 data with the 50 epochs take 18.2 s. When trained with sensors derived from the XYZ-axis accelerometer, the accuracy of the CNN model yields an accuracy of 94.23%.

**Table 6.** Training curves and confusion matrix of double-axis force data.

	Model Accuracy and Loss	Confusion Matrix																																	
(a) Force data of XY-axis		<table border="1"> <tr> <td rowspan="4">True label</td> <td>Belt No. 1</td> <td><b>796</b></td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Belt No. 2</td> <td>1</td> <td><b>797</b></td> <td>2</td> <td>0</td> </tr> <tr> <td>Belt No. 3</td> <td>0</td> <td>0</td> <td><b>800</b></td> <td>0</td> </tr> <tr> <td>Belt No. 4</td> <td>0</td> <td>0</td> <td>6</td> <td><b>788</b></td> </tr> <tr> <td></td> <td></td> <td>Belt No. 1</td> <td>Belt No. 2</td> <td>Belt No. 3</td> <td>Belt No. 4</td> </tr> <tr> <td></td> <td></td> <td colspan="4">Predicted label</td> </tr> </table>	True label	Belt No. 1	<b>796</b>	0	0	0	Belt No. 2	1	<b>797</b>	2	0	Belt No. 3	0	0	<b>800</b>	0	Belt No. 4	0	0	6	<b>788</b>			Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4			Predicted label			
True label	Belt No. 1	<b>796</b>		0	0	0																													
	Belt No. 2	1		<b>797</b>	2	0																													
	Belt No. 3	0		0	<b>800</b>	0																													
	Belt No. 4	0	0	6	<b>788</b>																														
		Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4																														
		Predicted label																																	
(b) Force data of XZ-axis		<table border="1"> <tr> <td rowspan="4">True label</td> <td>Belt No. 1</td> <td><b>794</b></td> <td>2</td> <td>0</td> <td>0</td> </tr> <tr> <td>Belt No. 2</td> <td>2</td> <td><b>797</b></td> <td>1</td> <td>0</td> </tr> <tr> <td>Belt No. 3</td> <td>0</td> <td>0</td> <td><b>793</b></td> <td>7</td> </tr> <tr> <td>Belt No. 4</td> <td>0</td> <td>0</td> <td>0</td> <td><b>794</b></td> </tr> <tr> <td></td> <td></td> <td>Belt No. 1</td> <td>Belt No. 2</td> <td>Belt No. 3</td> <td>Belt No. 4</td> </tr> <tr> <td></td> <td></td> <td colspan="4">Predicted label</td> </tr> </table>	True label	Belt No. 1	<b>794</b>	2	0	0	Belt No. 2	2	<b>797</b>	1	0	Belt No. 3	0	0	<b>793</b>	7	Belt No. 4	0	0	0	<b>794</b>			Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4			Predicted label			
True label	Belt No. 1	<b>794</b>		2	0	0																													
	Belt No. 2	2		<b>797</b>	1	0																													
	Belt No. 3	0		0	<b>793</b>	7																													
	Belt No. 4	0	0	0	<b>794</b>																														
		Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4																														
		Predicted label																																	
(c) Force data of YZ-axis		<table border="1"> <tr> <td rowspan="4">True label</td> <td>Belt No. 1</td> <td><b>794</b></td> <td>2</td> <td>0</td> <td>0</td> </tr> <tr> <td>Belt No. 2</td> <td>2</td> <td><b>797</b></td> <td>1</td> <td>0</td> </tr> <tr> <td>Belt No. 3</td> <td>0</td> <td>0</td> <td><b>793</b></td> <td>7</td> </tr> <tr> <td>Belt No. 4</td> <td>0</td> <td>0</td> <td>0</td> <td><b>794</b></td> </tr> <tr> <td></td> <td></td> <td>Belt No. 1</td> <td>Belt No. 2</td> <td>Belt No. 3</td> <td>Belt No. 4</td> </tr> <tr> <td></td> <td></td> <td colspan="4">Predicted label</td> </tr> </table>	True label	Belt No. 1	<b>794</b>	2	0	0	Belt No. 2	2	<b>797</b>	1	0	Belt No. 3	0	0	<b>793</b>	7	Belt No. 4	0	0	0	<b>794</b>			Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4			Predicted label			
True label	Belt No. 1	<b>794</b>		2	0	0																													
	Belt No. 2	2		<b>797</b>	1	0																													
	Belt No. 3	0		0	<b>793</b>	7																													
	Belt No. 4	0	0	0	<b>794</b>																														
		Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4																														
		Predicted label																																	

Note: please see Table 2 for detailed information on the classification label.

#### 4.2. CNN Results of Force Signals

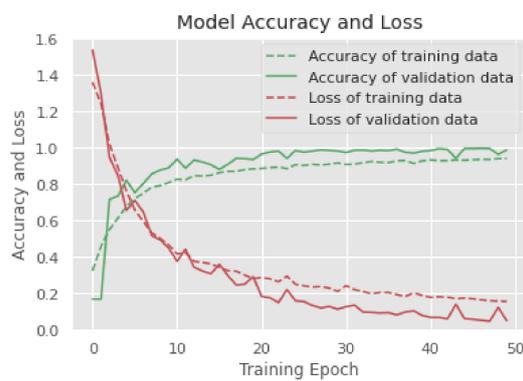
This sub-section provides the CNN classification results of force data from three categories: (1) single-axis (X, Y and Z), (2) double-axis (XY, XZ and YZ) and (3) triple-axis (XYZ).

##### 4.2.1. Force Signal of Single-Axis Dynamometer (X, Y and Z)

This study also utilized the CNN method for the three-axis force data, as presented in Table 5. The CNN result in the X-axis (Table 5a) indicates that 793 datasets of the “brand-new” class can be classified appropriately, and 3 datasets were misclassified to the “5-min cycle time” class. In the “5-min cycle time” class, 797 datasets were classified

correctly, 3 datasets were identified as “brand-new” class. In the “15-min cycle time” class, 798 datasets were classified correctly, 1 dataset was classified incorrectly to “brand-new”, and 1 dataset was classified incorrectly as “worn-out”. In the “worn-out” class, 747 datasets were classified correctly. However, 47 datasets were classified incorrectly to the “15-min cycle time” class.

Table 5b shows that 794 datasets of the “brand-new” class can be classified, and 2 datasets were incorrectly classified as “5-min cycle time” class. In the “5-min cycle time” class, 796 datasets were classified correctly, and 4 datasets were identified as “worn-out”. In the “15-min cycle time” class, 771 datasets were classified correctly, and 29 datasets were classified incorrectly to the “worn-out” class. Meanwhile, in the “worn-out” class, 764 datasets were classified correctly, and 30 datasets were classified incorrectly to the “15-min cycle time” class.

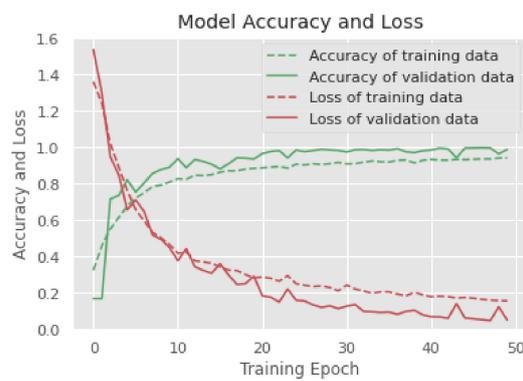


(a)

True label	Belt No. 1	795	0	0	0
	Belt No. 2	1	794	6	0
	Belt No. 3	0	0	800	0
	Belt No. 4	0	0	29	765
		Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4
		Predicted label			

(b)

Figure 19. (a) Model accuracy and loss curves; and (b) confusion matrix of triple-axis vibration data (please see Table 2 for detailed information on classification label).



(a)

True label	Belt No. 1	795	0	0	0
	Belt No. 2	1	794	6	0
	Belt No. 3	0	0	800	0
	Belt No. 4	0	0	29	765
		Belt No. 1	Belt No. 2	Belt No. 3	Belt No. 4
		Predicted label			

(b)

Figure 20. (a) Model accuracy and loss; and (b) Confusion matrix of triple-axis force data. (Please see Table 2 for detailed information on the classification label).

Table 5c shows that 794 datasets of the “brand-new” class can be classified correctly. However, 2 datasets were misclassified as “5-min cycle time” class. In the “5-min cycle time” class, 798 datasets were classified correctly, and 2 datasets were incorrectly classified as “15-min cycle time”. In the “15-min cycle time” class, 788 datasets were classified correctly. However, 12 datasets were classified incorrectly to the “worn-out” class. In the “worn-out” class, 785 datasets were classified correctly, and 9 datasets were misclassified as “15-min cycle time” class.

Table 5 shows the accuracy and loss curves during CNN training using force data input from X, Y or Z axes. In Table 5, it can be seen that processing 7672 data with 50 epochs took

nearly 17 to 20 s. When trained with sensors derived from the X-axis of the dynamometer, the accuracy of the CNN model resulted in higher accuracy (accuracy: 93.07%) than the other axes (Y-axis, accuracy: 92.14%; Z-axis, accuracy: 92.94%).

#### 4.2.2. Force Signal of Double-Axis Dynamometer (XY, XZ and YZ)

The force signature across two directions were combined to examine the dynamic behavior during the abrasive process. This combined signal was input into the CNN algorithm. The confusion matrices from the CNN model that used two-axis (XY, XZ and YZ) is presented in Table 6. According to the CNN results presented in Table 6, Table 6a indicates that the CNN model results show highest than other combinations. In detail, Table 6a indicates that all 796 datasets of the “brand-new” class can be classified perfectly. For the “5-min cycle time” class, 797 datasets were classified correctly, 1 dataset was identified as “brand-new” class, and 2 datasets were incorrectly classified as “15-min cycle time” class. In the “used” class, all 800 datasets were also classified perfectly. Furthermore, in the “worn-out” class, 788 datasets were classified correctly, and 6 datasets were incorrectly identified as “15-min cycle time” class.

Table 6 shows the accuracy and loss curves during CNN training with inputs from axes combinations such as of XY, XZ or YZ. In Table 6, it can be seen that processing 7672 data with the number of 50 epochs takes 17 to 20 s. When trained with sensors derived from the XY-axis of the dynamometer, the accuracy of the CNN model results in higher accuracy (accuracy: 99.80%) than the other axes (XZ-axes, accuracy: 95.53%; YZ-axes, accuracy: 96.47%).

#### 4.2.3. Force Signal of Triple-Axis Dynamometer (XYZ)

Model accuracy and the confusion matrix from the CNN model that used tree axis from dynamometer sensor is presented in Figure 20. Figure 20 indicates that 793 datasets of “brand-new” can be classified appropriately, and three datasets were misclassified to the “5-min cycle time” class. Additionally, in the “5-min cycle time” class, 797 datasets were classified correctly, 2 datasets were misclassified to “brand-new” class and 1 dataset was misclassified to “15-min cycle time” class. Moreover, in the “15-min cycle time” class, 798 datasets were classified correctly and 2 datasets were incorrectly classified to “worn-out” class. For the “worn-out” class, 790 datasets were classified correctly and 4 datasets were misclassified to the “15-min cycle time” class.

Figure 20 shows the accuracy and loss curves during CNN model training using all dynamometer sensor inputs. In Figure 9, it can be seen that processing 7672 data with the number of 50 epochs takes 18.2 s. When trained with sensors derived from the XYZ-axis dynamometer, the accuracy of the CNN model yields an accuracy of 96.34%.

### 4.3. Statistical Measurement of CNN Prediction and Classification

#### 4.3.1. Accelerometer (Vibration) Data

This section provides a statistical summary of CNN prediction and classification of abrasive belt grinding condition from brand-new to worn-out condition. The statistical summary of metrics such as training time, testing time, accuracy on test data and loss on test data.

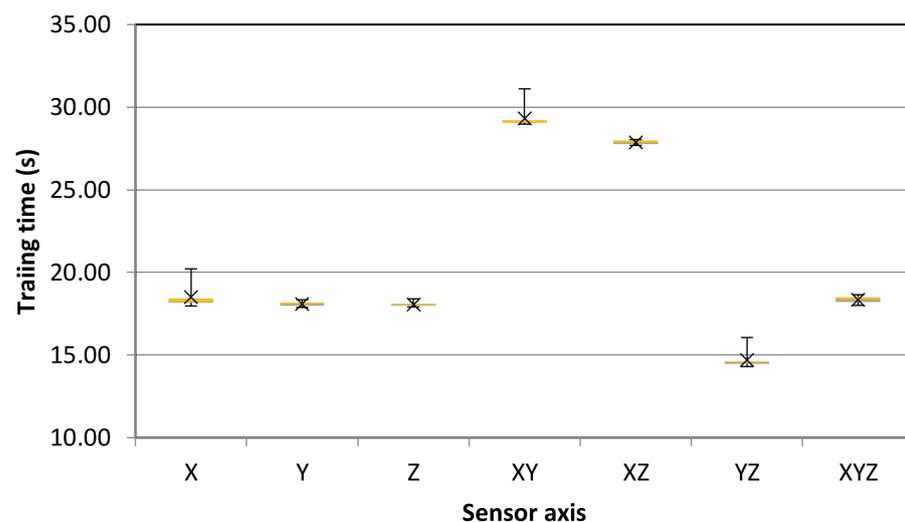
##### Training Time

Table 7 shows the summary of ‘training time’ for the CNN model. Each combination of three-axis accelerometer sensor showed different time required to train the model. Table 7 indicates that a combination of the XY axes of the accelerometer sensor shows less ‘training time’, i.e.,  $14.71 \pm 0.51$  s than other axis combination.

**Table 7.** CNN summary of ‘training time’ (sec).

Data No.#	X	Y	Z	XY	XZ	YZ	XYZ
1	20.21	18.29	17.98	31.11	27.91	16.06	18.20
2	18.05	18.10	17.91	29.11	27.99	14.60	18.00
3	18.18	18.06	18.16	29.09	27.97	14.57	18.57
4	18.27	17.97	18.02	29.64	27.82	15.03	18.37
5	19.23	18.12	18.09	29.00	28.00	14.64	18.43
6	18.31	18.08	18.10	29.07	28.04	14.30	18.14
7	18.45	18.19	18.40	29.06	27.69	14.46	18.23
8	18.16	17.94	18.03	29.26	27.69	14.43	18.65
9	17.97	18.35	17.91	29.07	27.81	14.47	18.35
10	18.21	17.88	18.03	28.97	27.76	14.56	18.53

A box plot of the ‘training time’ of the CNN model for a different accelerometer axis direction is presented in Figure 21. All of the model’s combination input showed that the standard deviation value is small (Q3-Med), as presented in Table 7. This is indicated that the proposed architecture model was able to train the CNN model. The highest time required to train the model is shown by the XY combination (29.64 s). In training the model, using a single input axis (X, Y or Z), it is shown that the time required is almost similar. However, by using more input (all axis), the time needed to train the model was not reduced.

**Figure 21.** Box plot of ‘training time’ for all sensor axis of the accelerometer.

The boxplot is divided into four quartiles, namely 1st, 2nd, 3rd, and 4th quartiles. The 1st, 2nd, 3rd, and 4th quartiles show a range data distribution of 0–25%, 25–50%, 50–75%, and 75–100%, respectively. The grey and yellow color in the boxplot of Figures 21–28. indicates the second, and third quartile, respectively. A wider box in the boxplot shows more distribution of data.

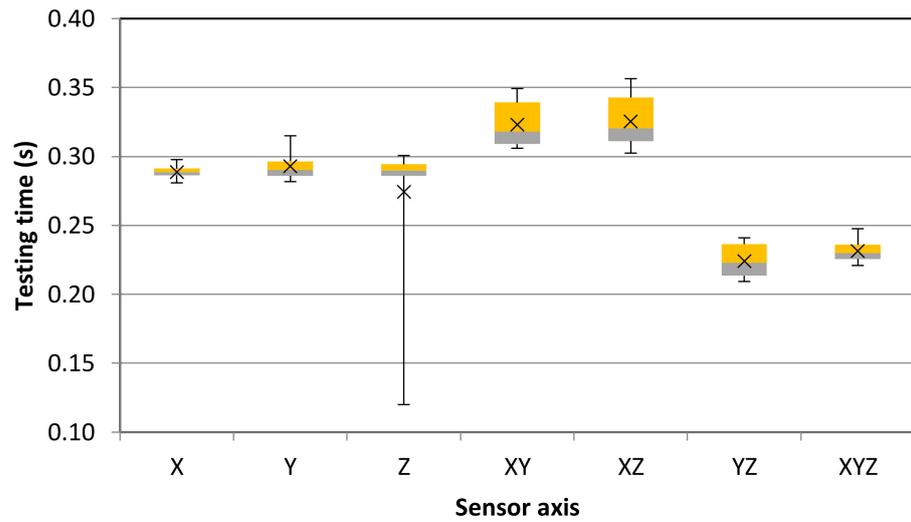


Figure 22. Box plot of 'testing time' for all sensor axis of the accelerometer.

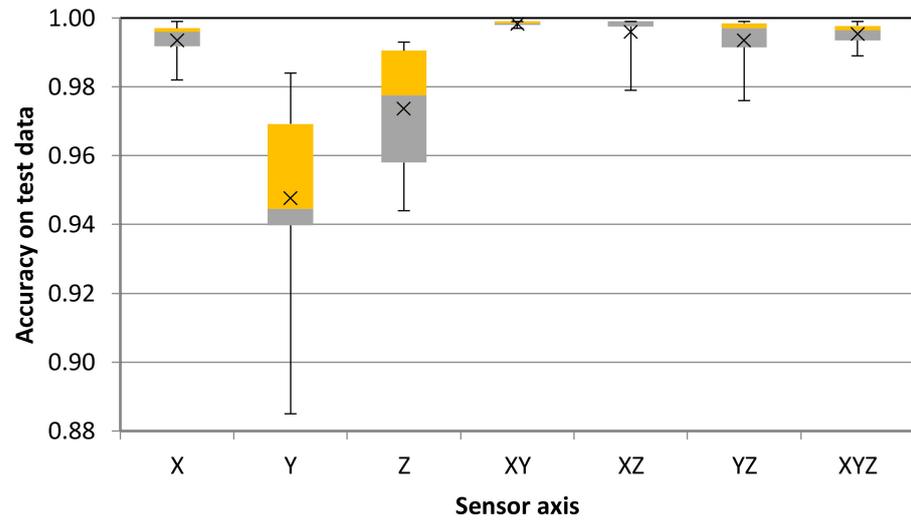


Figure 23. Box plot of 'accuracy on test data' for all sensor axis of the accelerometer.

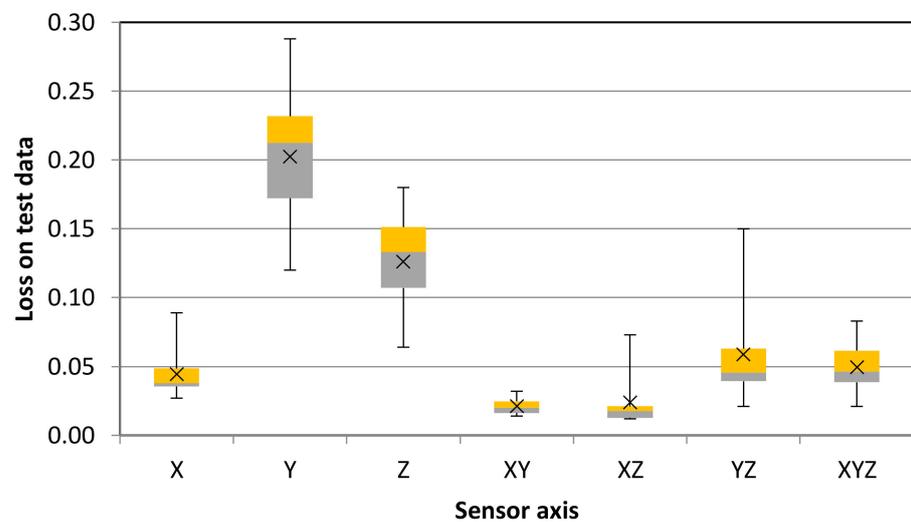


Figure 24. Box plot of 'loss on test data' for all sensor axis of the accelerometer.

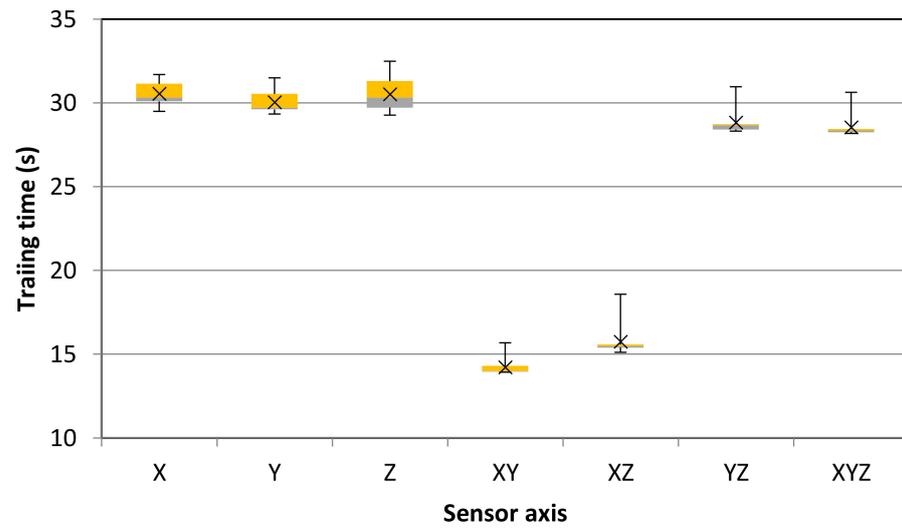


Figure 25. Box plot of 'training time' for all sensor axis of the dynamometer.

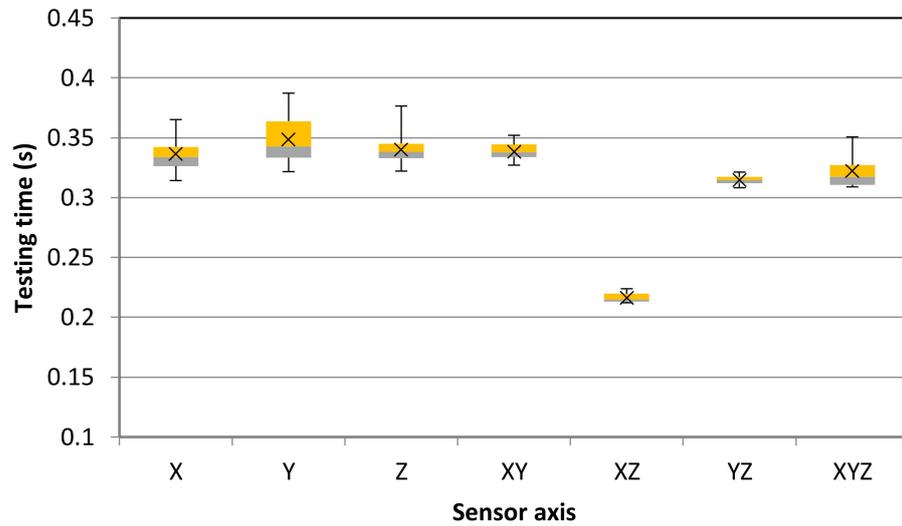


Figure 26. Box plot of 'testing time' for all sensor axis of the dynamometer.

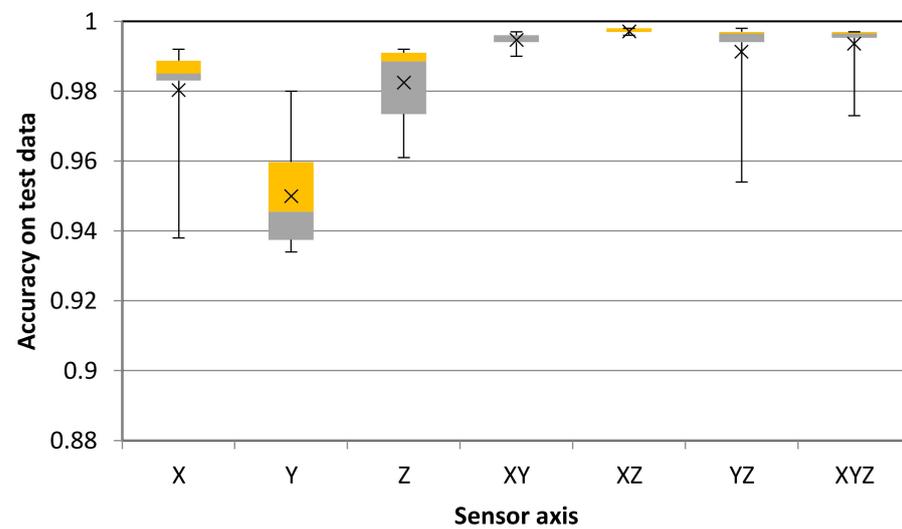


Figure 27. Box plot of 'accuracy on test data' for all sensor axis of the dynamometer.

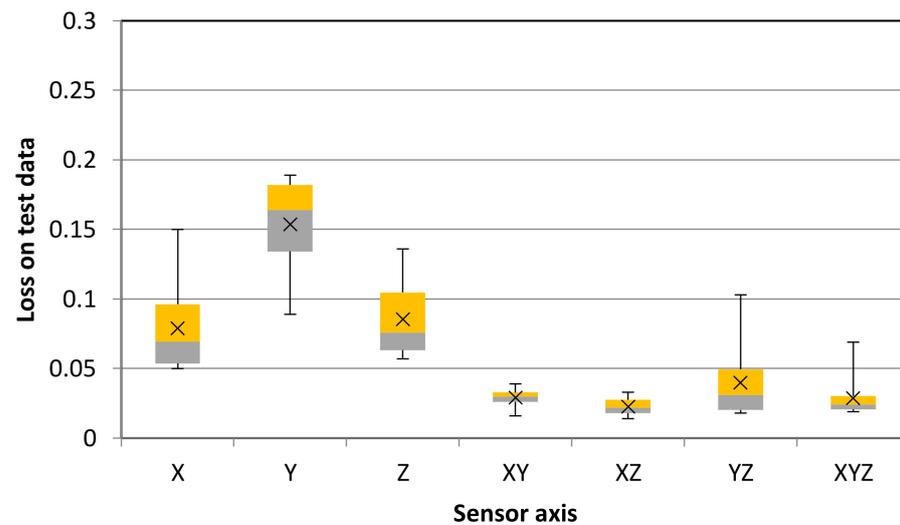


Figure 28. Box plot of ‘loss on test data’ for all sensor axis of the dynamometer.

#### Testing Time

Table 8 shows the summary of ‘testing time’ for the CNN model. Each combination of three-axis accelerometer sensor showed a different time to test the model. Table 8 indicates that a combination of the accelerometer sensor’s YZ-axes shows less time ( $0.22 \pm 0.01$  s) than other axis combination.

Table 8. CNN summary of ‘testing time’ (s).

Data No.	X	Y	Z	XY	XZ	YZ	XYZ
1	0.28	0.28	0.29	0.31	0.35	0.21	0.22
2	0.29	0.28	0.30	0.31	0.32	0.22	0.23
3	0.29	0.30	0.29	0.31	0.35	0.23	0.23
4	0.29	0.29	0.29	0.32	0.32	0.24	0.23
5	0.29	0.29	0.28	0.35	0.36	0.21	0.24
6	0.29	0.30	0.30	0.32	0.31	0.24	0.22
7	0.29	0.31	0.12	0.31	0.31	0.21	0.23
8	0.30	0.29	0.29	0.32	0.30	0.21	0.24
9	0.28	0.29	0.29	0.35	0.31	0.23	0.25
10	0.29	0.28	0.29	0.34	0.32	0.24	0.23

A box plot of ‘testing time’ of the CNN model for a different combination of accelerometer axis is presented in Figure 22. All of the model’s combination input showed that the standard deviation value is low (Q3-Med), as presented in Table 8. This indicated that the proposed architecture model was able to test the CNN model. The highest time required to test the model is shown by the XZ combination (0.33 s). In training the model, using a single input axis (X, Y or Z), it is shown that the required processing time is almost similar. However, by using more input (all axis), the time needed to train the model was not reduced. This issue was confirmed when we compared between YZ and XYZ combination.

#### Accuracy of Test Data

Table 9 shows the summary of the performance of the CNN model. Each combination of three-axis accelerometer sensor showed different accuracy. Table 9 indicates that a combination of the accelerometer sensor XY-axes shows better accuracy than that of other combinations ( $1.00 \pm 0.00$ ). The lowest accuracy was found when the model was trained using Y-axis ( $0.95 \pm 0.03$ ).

**Table 9.** CNN summary of ‘accuracy on test data’.

Data No.	X	Y	Z	XY	XZ	YZ	XYZ
1	1.00	0.93	0.99	1.00	0.99	1.00	0.99
2	0.99	0.96	0.96	1.00	0.98	0.99	1.00
3	1.00	0.98	0.95	1.00	1.00	1.00	1.00
4	1.00	0.94	0.99	1.00	1.00	0.98	1.00
5	1.00	0.94	0.97	1.00	1.00	1.00	1.00
6	0.98	0.95	0.99	1.00	1.00	1.00	0.99
7	0.99	0.98	0.98	1.00	1.00	1.00	0.99
8	0.99	0.89	0.97	1.00	1.00	0.99	1.00
9	1.00	0.94	0.94	1.00	1.00	1.00	1.00
10	1.00	0.97	0.99	1.00	1.00	0.99	1.00

A box plot of ‘accuracy on test data’ of the CNN model for a different combination of accelerometer axis is presented in Figure 23. The CNN model, which is trained using XY-axis datasets, showed the highest accuracy ( $1.00 \pm 0.00$ ). In contrast, The CNN model based on Y-axis indicated the lowest accuracy ( $0.89 \pm 0.02$ ). Additionally, the CNN which is trained using Y and Z datasets, showed wider deviation among the others axis combination.

#### Loss on Test Data

Table 10 shows the loss value of the CNN model. The loss value is important to indicate the summation of the error in the evaluation and testing datasets. Each combination of three-axis accelerometer sensor showed different loss value. Table 10 indicates that a combination of the accelerometer sensor’s XY-axes shows a better loss value than other combinations ( $0.02 \pm 0.01$ ). The worst loss value was found when the model was trained using Y-axis ( $0.20 \pm 0.05$ ).

**Table 10.** CNN summary of ‘loss on test data’.

Data No.	X	Y	Z	XY	XZ	YZ	XYZ
1	0.03	0.23	0.07	0.02	0.04	0.04	0.05
2	0.05	0.21	0.18	0.02	0.07	0.11	0.02
3	0.03	0.13	0.15	0.03	0.01	0.02	0.07
4	0.04	0.24	0.11	0.01	0.02	0.15	0.04
5	0.04	0.21	0.17	0.02	0.02	0.04	0.07
6	0.09	0.21	0.06	0.03	0.02	0.02	0.08
7	0.06	0.12	0.12	0.02	0.01	0.05	0.05
8	0.04	0.29	0.15	0.03	0.01	0.07	0.04
9	0.04	0.23	0.15	0.02	0.02	0.05	0.05
10	0.04	0.16	0.11	0.02	0.02	0.05	0.04

A box plot of ‘loss on test data’ of the CNN model for a different combination of accelerometer axis is presented in Figure 24. The CNN model, which is trained using XY-axis datasets, showed the lowest value ( $0.02 \pm 0.01$ ). In contrast, The CNN model based on the Y and Z axis indicated the higher loss and wider deviation value among others axis combination.

#### 4.3.2. Dynamometer (Force) Data

This section provides a statistical summary of CNN prediction and classification of abrasive belt grinding condition from brand-new to worn-out condition. The statistical summary includes ‘training time’, ‘testing time’, ‘accuracy on test data’ and ‘lost on test data’.

##### Training Time

Table 11 shows the summary of ‘training time’ for the CNN model. Each combination of three-axis dynamometer sensor showed different time required to train the model.

Table 11 indicates that a combination of the XY-axes of the dynamometer sensor shows less training time ( $14.23 \pm 0.54$  s) than other axis combinations.

**Table 11.** CNN summary of ‘training time’ (s).

No.	X	Y	Z	XY	XZ	YZ	XYZ
1	30.17	29.69	32.49	15.68	18.58	30.97	30.63
2	30.72	30.68	30.93	13.97	15.12	28.70	28.18
3	31.70	30.74	31.42	13.93	15.41	28.44	28.34
4	30.39	30.12	29.54	13.95	15.49	28.66	28.50
5	29.50	29.64	29.77	13.99	15.63	28.72	28.26
6	30.28	31.50	30.79	13.99	15.50	28.41	28.44
7	31.66	29.73	29.72	13.96	15.57	29.19	28.45
8	29.82	29.60	31.56	14.10	15.58	28.71	28.19
9	30.06	29.34	29.27	14.36	15.41	28.32	28.26
10	31.28	29.42	29.82	14.36	15.39	28.40	28.33

A box plot of the ‘training time’ of the CNN model for a different combination of dynamometer axis is presented in Figure 25. All of the combined input of the model showed that the value of the standard deviation is small, as presented in Table 11. This is indicated that the proposed architecture model was able to train the CNN model. The highest time required to train the model is presented on Z-axis (32.49 s). Using double and triple input axis in training, the model shows that the ‘training time’ required is less than the single-axis, especially XY with 14.23 s on average.

#### Testing Time

Table 12 shows the summary of ‘testing time’ for the CNN model. Each combination of three-axis accelerometer sensor showed a different time to test the model. Table 12 indicates that a combination of XZ-axes of accelerometer sensor shows less time ( $0.22 \pm 0.01$  s) than other axis combination.

**Table 12.** CNN summary of ‘testing time’ (s).

No.	X	Y	Z	XY	XZ	YZ	XYZ
1	0.33	0.36	0.34	0.33	0.22	0.31	0.31
2	0.31	0.37	0.34	0.35	0.21	0.31	0.31
3	0.34	0.34	0.38	0.34	0.21	0.31	0.33
4	0.33	0.39	0.35	0.35	0.21	0.32	0.31
5	0.32	0.33	0.34	0.35	0.21	0.32	0.34
6	0.34	0.36	0.32	0.33	0.22	0.32	0.31
7	0.33	0.34	0.33	0.33	0.22	0.31	0.32
8	0.33	0.34	0.35	0.34	0.22	0.32	0.32
9	0.36	0.32	0.33	0.34	0.22	0.31	0.35
10	0.37	0.33	0.33	0.33	0.21	0.32	0.31

A box plot of ‘testing time’ of the CNN model for a different combination of dynamometer axis is presented in Figure 26. All of the model’s combination input showed that the standard deviation value is small, as presented in Table 12. This is indicated that the proposed architecture model was able to test the CNN model. The highest time required to test the model is shown by the XZ combination (0.21 s) compared to the single input axis (X, Y and Z) and double input axis (XY and YZ). In addition, by using more inputs (XYZ) the time needed to test the model was not improved even better.

#### Accuracy on Test Data

Table 13 shows the summary of the performance of the CNN model. Each combination of three-axis dynamometer sensor showed different accuracy. Table 13 indicates that a combination of the XZ-axes of the dynamometer sensor shows better accuracy ( $1.00 \pm 0.00$ )

than other axis combinations. The lowest accuracy was found when the model was trained using Y-axis datasets, i.e.,  $0.95 \pm 0.02$ .

**Table 13.** CNN summary of ‘accuracy on test data’.

Data No.	X	Y	Z	XY	XZ	YZ	XYZ
1	0.98	0.98	0.98	0.99	1.00	1.00	1.00
2	0.98	0.94	0.99	1.00	1.00	0.99	0.97
3	0.99	0.93	0.99	1.00	1.00	0.99	1.00
4	0.99	0.96	0.99	1.00	1.00	1.00	0.99
5	0.99	0.94	0.97	0.99	1.00	1.00	1.00
6	0.97	0.94	0.97	0.99	1.00	1.00	1.00
7	0.99	0.96	0.99	0.99	1.00	1.00	1.00
8	0.94	0.96	0.96	1.00	1.00	0.99	1.00
9	0.99	0.94	0.99	1.00	1.00	0.95	1.00
10	0.98	0.95	0.99	1.00	1.00	1.00	1.00

A box plot of ‘accuracy on test data’ of the CNN model for a different combination of dynamometer axis is presented in Figure 27. The CNN model, which is trained using XZ-axis datasets, showed the highest accuracy ( $1.00 \pm 0.00$ ). In contrast, The CNN model based on Y-axis indicated the lowest accuracy ( $0.95 \pm 0.02$ ). Additionally, the CNN which is trained using Y and Z datasets showed wider deviation among the others axis combination.

#### Loss on Test Data

Table 14 shows the loss value of the CNN model. The loss value is important to indicate the summation of the error in the evaluation and testing datasets. Each combination of three-axis dynamometer sensor showed different loss value. Table 14 indicates that a combination of the XZ-axes of the accelerometer sensor shows a better loss value than other combinations ( $0.02 \pm 0.01$ ). The worst loss value was found when the model was trained using Y-axis ( $0.15 \pm 0.03$ ).

**Table 14.** CNN summary of ‘loss on test data’.

Data No.	X	Y	Z	XY	XZ	YZ	XYZ
1	0.10	0.09	0.10	0.04	0.03	0.02	0.02
2	0.08	0.18	0.06	0.02	0.02	0.04	0.07
3	0.06	0.18	0.06	0.02	0.02	0.05	0.02
4	0.05	0.13	0.06	0.03	0.03	0.02	0.03
5	0.06	0.18	0.12	0.03	0.02	0.03	0.02
6	0.09	0.19	0.11	0.04	0.02	0.02	0.02
7	0.05	0.11	0.07	0.03	0.02	0.02	0.02
8	0.15	0.14	0.14	0.03	0.02	0.07	0.02
9	0.05	0.18	0.08	0.03	0.03	0.10	0.03
10	0.10	0.15	0.06	0.03	0.01	0.03	0.03

A box plot of the “loss on test data” of the CNN model for a different combination of dynamometer axis is presented in Figure 28. The CNN model, which is trained using XZ-axis datasets, showed the lowest value ( $0.02 \pm 0.01$ ). In contrast, The CNN model based on the Y and Z axis indicated the higher loss and wider deviation value among others axis combination.

#### 4.3.3. Discussion

Based on the CNN results presented in Section 4.1 for vibration signals and Section 4.2 for force signals, it is found that the sensor data across different axes has significant variation in representing the dynamic behavior of the abrasive process, which is directly related to the belt grinding condition.

In vibration signals, the X-axis direction shows the highest classification accuracy among the single-axis sensors assessment compared to those from the other two single-axis sensors assessment compared to those from the other two (Y-axis and Z-axis). The result of the combining axes showed that the combination of XZ and YZ has higher accuracy than the XY combination. Moreover, the signal combination from the triple-axis (XYZ) does not improve the accuracy. The accuracy summary of the three-axis accelerometer signals assessment is presented in Table 15.

**Table 15.** Accuracy summary of CNN results for three-axis accelerometer and its combination.

Belt No.	X		Y		Z		XY		XZ		YZ		XYZ	
	C	I	C	I	C	I	C	I	C	I	C	I	C	I
1	796	0	795	1	790	6	796	0	796	0	796	0	795	1
2	796	4	798	2	791	9	797	3	798	2	798	2	794	6
3	798	2	717	83	748	52	800	0	800	0	800	0	800	0
4	793	1	670	124	719	75	788	6	794	0	794	0	765	29
Total	3183	7	2980	210	3048	142	3181	9	3188	2	3188	2	3154	36

Note: C is the correct prediction (classification), and I is the miss-classification. For 'Belt No.' information, please refer to Table 2. This table is extracted from all confusion matrix of CNN results in Section 4.1.

In force signals, all single-axis sensor assessment shows significant accuracy drop due to a lot of misclassified datasets. In addition, the result combining sensor data between two axes shows that higher accuracy is obtained from the combination of XY compared to the other combinations (XZ and YZ). Moreover, the triple-axis (XYZ) signal combination does not show better accuracy compared to the single-axis and double-axis assessment. The accuracy summary of the three-axis accelerometer signals assessment is presented in Table 16.

**Table 16.** Accuracy summary of CNN results for three-axis dynamometer and its combination.

Belt No.	X		Y		Z		XY		XZ		YZ		XYZ	
	C	I	C	I	C	I	C	I	C	I	C	I	C	I
1	793	3	794	2	794	2	796	0	794	2	794	2	795	1
2	797	3	796	4	798	2	797	3	797	3	798	2	794	6
3	798	2	771	29	788	12	800	0	793	7	794	6	800	0
4	747	47	764	30	785	9	788	6	794	0	793	17	765	29
Total	3135	55	3125	65	3165	25	3181	9	3178	12	3179	27	3154	36

Note: C is the correct prediction (classification), and I is the miss-classification. For 'Belt No.' information, please refer to Table 2. This table is extracted from all confusion matrix of CNN results in Section 4.2.

## 5. Conclusions

A methodology for abrasive belt grinding condition monitoring and worn state prediction is presented. The methodology is focused on understanding the effect of the directions of vibration and force signatures with respect to the belt wear rate. This proposed methodology also responds to the open question, i.e., opening the possibility of applying DL for tool condition monitoring and its potential application. From the presented result, the methodology revealed that the accelerometer and dynamometer's direction has a certain relation to the physical behavior of the belt grinding process, especially on the dynamics in the contact between the belt tool and workpiece. In vibration signal analysis, XZ and YZ direction combination have more accuracy than the others. In contrast, in force signal analysis, the combination of XY gives the most accurate results in comparison to the others. Another outcome of this study is also attributed to our future work that the DL model can be embedded into a monitoring measurement system to monitor the belt state during the abrasive process without interrupting the ongoing grinding process. Even though the DL method has potential application in tool wear prediction of mild steel, this study

needs further investigation if the proposed DL prediction can be generalized and translated across different materials during grinding, which is the research direction in progress.

**Author Contributions:** Conceptualization, W.C. and T.T. (Triwiyanto Triwiyanto); methodology, W.C. and T.T. (Triwiyanto Triwiyanto); software, T.T. (Triwiyanto Triwiyanto); validation, W.C. and T.T. (Triwiyanto Triwiyanto); formal analysis, W.C. and T.T. (Triwiyanto Triwiyanto); resources, V.P., and T.T. (Tegoeh Tjahjowidodo); data curation, V.P.; writing—original draft preparation, W.C., T.T. (Triwiyanto Triwiyanto), V.P. and S.D.H.P.; writing—review and editing, W.C., A.G. and T.T. (Tegoeh Tjahjowidodo); visualization, W.C., T.T. (Triwiyanto Triwiyanto) and V.P.; supervision, A.G. and T.T. (Tegoeh Tjahjowidodo); project administration, V.P.; funding acquisition, W.C., A.G. and T.T. (Tegoeh Tjahjowidodo). All authors have read and agreed to the published version of the manuscript.

**Funding:** The first author would like to thank to Universiti Brunei Darussalam for providing a research grant for this study through Research Grant No. UBD/RSCH/1.3/FICBF(b)/2019/007 and The APC was funded by Professor Dr. Adam Glowacz.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Zhan, J.; Yu, S. Study on error compensation of machining force in aspheric surfaces polishing by profile-adaptive hybrid movement–force control. *Int. J. Adv. Manuf. Technol.* **2011**, *54*, 879–885. [[CrossRef](#)]
- Hauth, S.; Linsen, L. Cycloids for polishing along doublespiral tool paths in configuration space. *Int. J. Adv. Manuf. Technol.* **2012**, *60*, 343–356. [[CrossRef](#)]
- Tian, Y.B.; Zhong, Z.W.; Lai, S.T.; Ang, Y.J. Development of fixed abrasive chemical mechanical polishing process for glass disk substrates. *J. Adv. Manuf. Technol.* **2013**, *68*, 993–1000. [[CrossRef](#)]
- Cheung, C.F.; Kong, L.B.; Ho, L.T.; To, S. Modelling and simulation of structure surface generation using computer controlled ultra-precision polishing. *Precis. Eng.* **2011**, *35*, 574–590. [[CrossRef](#)]
- Sun, H.; Liu, Y.; Pan, J.; Zhang, J.; Ji, W. Enhancing cutting tool sustainability based on remaining useful life prediction. *J. Clean. Prod.* **2020**, *244*, 118794. [[CrossRef](#)]
- Liu, C.; Li, Y.; Hua, J.; Lu, N.; Mou, W. Real-time cutting tool state recognition approach based on machining features in NC machining process of complex structural parts. *Int. J. Adv. Manuf. Technol.* **2018**, *97*, 229–241. [[CrossRef](#)]
- Kothuru, A.; Nooka, S.P.; Liu, R. Application of deep visualization in CNN-based tool condition monitoring for end milling. *Procedia Manuf.* **2019**, *34*, 995–1004. [[CrossRef](#)]
- Zhou, Y.; Xue, W. Review of tool condition monitoring methods in milling processes. *Int. J. Adv. Manuf. Technol.* **2018**, *96*, 2509–2523. [[CrossRef](#)]
- Mohanraj, T.; Shankar, S.; Rajasekar, R.; Sakthivel, N.; Pramanik, A. Tool condition monitoring techniques in milling process—A review. *J. Mater. Res. Technol.* **2020**, *9*, 1032–1042. [[CrossRef](#)]
- Cheng, C.; Li, J.; Liu, Y.; Nie, M.; Wang, W. Deep convolutional neural network-based in-process tool condition monitoring in abrasive belt grinding. *Comput. Ind.* **2019**, *106*, 1–13. [[CrossRef](#)]
- Pandiyan, V.; Shevchik, S.; Wasmer, K.; Castagne, S.; Tjahjowidodo, T. Modelling and monitoring of abrasive finishing processes using artificial intelligence techniques: A review. *J. Manuf. Process.* **2020**, *57*, 114–135. [[CrossRef](#)]
- Pandiyan, V. Modelling and In-Process Monitoring of Abrasive Belt Grinding Process. Ph.D. Thesis, Nanyang Technological University, Singapore, 2019.
- Pandiyan, V.; Caesarendra, W.; Tjahjowidodo, T.; Praveen, G. Predictive Modelling and Analysis of Process Parameters on Material Removal Characteristics in Abrasive Belt Grinding Process. *Appl. Sci.* **2017**, *7*, 363. [[CrossRef](#)]
- Pandiyan, V.; Tjahjowidodo, T. In-process endpoint detection of weld seam removal in robotic abrasive belt grinding process. *Int. J. Adv. Manuf. Technol.* **2017**, *93*, 1699–1714. [[CrossRef](#)]
- Zhang, X.; Kuhlentötter, B.; Kneupner, K. An efficient method for solving the Signorini problem in the simulation of free-form surfaces produced by belt grinding. *Int. J. Mach. Tools Manuf.* **2005**, *45*, 641–648. [[CrossRef](#)]
- Pandiyan, V.; Tjahjowidodo, T.; Samy, M.P. In-Process Surface Roughness Estimation Model for Compliant Abrasive Belt Machining Process. *Procedia CIRP* **2016**, *46*, 254–257. [[CrossRef](#)]
- Pandiyan, V.; Tjahjowidodo, T. Use of Acoustic Emissions to detect change in contact mechanisms caused by tool wear in abrasive belt grinding process. *Wear* **2019**, *436–437*, 203047. [[CrossRef](#)]
- Pandiyan, V.; Caesarendra, W.; Tjahjowidodo, T.; Tan, H.H. In-process tool condition monitoring in compliant abrasive belt grinding process using support vector machine and genetic algorithm. *J. Manuf. Process.* **2018**, *31*, 199–213. [[CrossRef](#)]
- Zhang, X.; Chen, H.; Xu, J.; Song, X.; Wang, J.; Chen, X. A novel sound-based belt condition monitoring method for robotic grinding using optimally pruned extreme learning machine. *J. Mater. Process. Technol.* **2018**, *260*, 9–19. [[CrossRef](#)]
- Pandiyan, V.; Murugan, P.; Tjahjowidodo, T.; Caesarendra, W.; Manyar, O.M.; Then, D.J.H. In-process virtual verification of weld seam removal in robotic abrasive belt grinding process using deep learning. *Robot. Comput. Manuf.* **2019**, *57*, 477–487. [[CrossRef](#)]

21. Wang, N.; Zhang, G.; Ren, L.; Pang, W.; Wang, Y. Vision and sound fusion-based material removal rate monitoring for abrasive belt grinding using improved LightGBM algorithm. *J. Manuf. Process.* **2021**, *66*, 281–292. [[CrossRef](#)]
22. Wang, N.; Zhang, G.; Pang, W.; Ren, L.; Wang, Y. Novel monitoring method for material removal rate considering quantitative wear of abrasive belts based on LightGBM learning algorithm. *Int. J. Adv. Manuf. Technol.* **2021**, *114*, 3241–3253. [[CrossRef](#)]
23. Triwiyanto, T.; Pawana, I.P.A.; Purnomo, M.H. An Improved Performance of Deep Learning Based on Convolution Neural Network to Classify the Hand Motion by Evaluating Hyper Parameter. *IEEE Trans. Neural Syst. Rehabilitation Eng.* **2020**, *28*, 1678–1688. [[CrossRef](#)]
24. Lindsay, G.W. Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future. *J. Cogn. Neurosci.* **2020**, 1–15. [[CrossRef](#)]
25. Demir, F.; Turkoglu, M.; Aslan, M.; Sengur, A. A new pyramidal concatenated CNN approach for environmental sound classification. *Appl. Acoust.* **2020**, *170*, 107520. [[CrossRef](#)]
26. Cao, X.-C.; Chen, B.-Q.; Yao, B.; He, W.-P. Combining translation-invariant wavelet frames and convolutional neural network for intelligent tool wear state identification. *Comput. Ind.* **2019**, *106*, 71–84. [[CrossRef](#)]
27. Zhang, Q.; Zhang, M.; Chen, T.; Sun, Z.; Ma, Y.; Yu, B. Recent advances in convolutional neural network acceleration. *Neurocomputing* **2019**, *323*, 37–51. [[CrossRef](#)]
28. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
29. Tavakkoli, V.; Mohsenzadegan, K.; Kyamakya, K. A Visual Sensing Concept for Robustly Classifying House Types through a Convolutional Neural Network Architecture Involving a Multi-Channel Features Extraction. *Sensors* **2020**, *20*, 5672. [[CrossRef](#)] [[PubMed](#)]
30. Zhao, Z.-Q.; Zheng, P.; Xu, S.-T.; Wu, X. Object Detection with Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [[CrossRef](#)] [[PubMed](#)]
31. Wang, Y.; Li, Y.; Song, Y.; Rong, X. The influence of the activation function in a convolution neural network model of facial expression recognition. *Appl. Sci.* **2020**, *10*, 1897. [[CrossRef](#)]
32. Shrestha, A.; Mahmood, A. Review of Deep Learning Algorithms and Architectures. *IEEE Access* **2019**, *7*, 53040–53065. [[CrossRef](#)]
33. Nguyen, P.T.; Ruscio, D.D.; Pierantonio, A.; Rocco, J.D.; Lovino, L. Convolutional neural networks for enhanced classification mechanisms of metamodels. *J. Syst. Softw.* **2021**, *172*, 110860. [[CrossRef](#)]