

Article

# Tool-Wear Analysis Using Image Processing of the Tool Flank

Ovidiu Gheorghe Moldovan <sup>1,†,‡</sup>, Simona Dzitac <sup>2,‡</sup>, Ioan Moga <sup>1,‡</sup>, Tiberiu Vesselenyi <sup>1,‡</sup>  
and Ioan Dzitac <sup>3,4,\*,‡</sup>

<sup>1</sup> Department of Mechatronics, University of Oradea, 410087 Oradea, Romania; omoldovan@uoradea.ro (O.G.M.); imoga@uoradea.ro (I.M.); tvesselenyi@yahoo.co.uk (T.V.)

<sup>2</sup> Department of Energy Engineering, University of Oradea, 410087 Oradea, Romania; simona@dzitac.ro

<sup>3</sup> Department of Mathematics—Computer Science, Aurel Vlaicu University of Arad, 310025 Arad, Romania

<sup>4</sup> Department of Social Sciences, Agora University, 410526 Oradea, Romania

\* Correspondence: professor.ioan.dzitac@ieee.org; Tel.: +40-359-101-032

† Current address: Department of Mathematics—Computer Science, Aurel Vlaicu University of Arad, 310025 Arad, Romania.

‡ These authors contributed equally to this work.

Received: 5 November 2017; Accepted: 28 November 2017; Published: 30 November 2017

**Abstract:** Flexibility of manufacturing systems is an essential factor in maintaining the competitiveness of industrial production. Flexibility can be defined in several ways and according to several factors, but in order to obtain adequate results in implementing a flexible manufacturing system able to compete on the market, a high level of autonomy (free of human intervention) of the manufacturing system must be achieved. There are many factors that can disturb the production process and reduce the autonomy of the system, because of the need of human intervention to overcome these disturbances. One of these factors is tool wear. The aim of this paper is to present an experimental study on the possibility to determine the state of tool wear in a flexible manufacturing cell environment, using image acquisition and processing methods.

**Keywords:** image processing; flexible manufacturing; tool-flank-wear monitoring; artificial neural networks

## 1. Introduction

The assessment of tool wear is of major importance in a manufacturing system that aims for higher automation and flexibility. The automatic tool readjustment (ATR) function implemented in flexible manufacturing systems (FMSs) prepares a new set of tools in the storage unit of the automatic tool changer (ATC) of the machine. The basic implementation of the ATR function in a FMS is based on a tool list. Each individual machine in the FMS transfers the tool list of its ATC magazine to a central control system; the system also has a list of workpieces to be manufactured that includes a tool list needed for each workpiece to be manufactured. The goal of the ATR is to transfer the required tools to the ATC in “hidden time”, meaning that the machine is still working while the tools for the new task are transferred, so the machine will have all the tools needed for each workpiece. Although the implementation of the ATR function significantly decreases the down time of the machines as a result of tool replacement in the ATC magazine, it has no effect in case of tool-life management (TLM). In order to manage the tool life with the ATR function, the system must monitor the tool wear and, on the basis of the standard life-time of the tools, include the tool determined to be close to its usage life in the ATR list of tools that need replacement. This system can improve the autonomy of the manufacturing system and the quality of the pieces. This kind of TLM, although bringing a significant improvement, has its own disadvantages. The efficiency of the system strongly depends on the precision of the

tool-life calculation. Depending on the true situation of cutting conditions (tool-material quality and its accordance with the standard, exact composition and homogeneity of the part material, and efficiency of the cooling system), a tool can present early signs of wear, causing a decrease in the quality of manufactured workpieces (the true life-time is shorter than the standard), or a tool can be removed on the basis of a standard life-time even if the quality of the cutting process is satisfactory and the tool is performing with acceptable parameters (the true life-time is longer than the standard). There are considerably many studies regarding different approaches to tool management. A tool-management approach enabling an autonomous cooperation of tools and machine tools within a batch production system is presented in [1]. Considering the type of parameter of the cutting process that is monitored, these range from surface roughness and cutting force to vibration and chip shape. In [2], the authors analyzed whether cutting parameters (feed rate, and spindle speed) have an effect on tool wear and surface roughness. Surface roughness of processed parts is also studied as a parameter that can predict the state of the tool wear. Prediction of the surface roughness of a workpiece by using adaptive neuro-fuzzy inference system (ANFIS) modeling for the monitoring of unmanned production systems with tool-life management is presented in [3]. In other studies, machined surface images are analyzed on the basis of a support vector machine using as input the features extracted from the gray-level co-occurrence matrix [4], and in-process surface-roughness monitoring system for an end-milling operation is analyzed using neural-fuzzy methods in [5]. Another parameter monitored in order to identify tool wear is the cutting force. In [6], force-based tool-condition monitoring for a turning process using support vector regression analysis is used to establish the flank wear of the cutting tool. Cutting force signals are also used in [7] to estimate the tool wear and the surface quality, and in [8], a partial least-square regression method is presented to make the tool-wear prediction also on the basis of the force signal. Related to cutting-force measurements are measurements of the current amplitude of the main drive of the machine tool, presented in [9,10], where it is shown that this parameter can also be linked to tool-wear development. Vibrations and machine tool dynamics are studied in order to find their relation with tool wear [11]. Some studies combine signals from different sensors, such as, for example, in [12], where tool-wear prediction in milling is analyzed using the simultaneous detection of acceleration and spindle drive current. Additionally, chip morphologies are analyzed in order to evaluate tool-flank wear and its effects on surface roughness [13]. More direct methods are focused on measuring spatial tool wear using a three-dimensional (3D) laser profile sensor [14]. Regarding processing and analysis techniques used to identify tool wear, there are also a large number of methods employed: machine learning and computer vision techniques [15], wavelet extreme learning machine models [16] support vector regression [17], analytical mathematical models [18], empirical models [19] and co-evolutionary particle swarm optimization-based selective neural network ensembles [20]. Tool wear is also studied experimentally; researchers aim to develop different models to predict tool wear from experimental data [21–23]. In this paper, we analyzed a system on the basis of image acquisition and processing, which can provide useful information regarding the cutting tool usage on the basis of the cutting edge wear. Our system is intended to be used in conjunction with methods to detect tool breakage, such as, for example, measuring the main drive current or torque, which are already built into the computer numerical control (CNC) of modern machine tools. Regarding the image processing methods, we describe one artificial neural network (ANN) applied to classify the image features obtained by processing the images with classical image processing methods (filtering, edge detection and morphological operations) and two ANNs applied directly on the images without their preprocessing (one single-hidden-layer ANN and one two-hidden-layer autoencoder). In order to compare the results obtained for each ANN, we describe in detail the training and testing of the ANNs. We also studied the Training Success Rate (TSR) for these ANNs for a large range of nodes in each layer. The integration of such a system with the ATR function of a FMS could increase the autonomy of the system and the quality of manufactured parts, as well as ensure a rational tool usage.

## 2. Tool-Flank-Wear Monitoring System

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results and their interpretation, as well as the experimental conclusions that can be drawn.

This area is relatively protected from chips and cooling liquid used during the processing, which could substantially impede the acquisition of the images. Thus, after every instance that the tool has been used and placed in the magazine, a tool-flank image can be acquired and processed. In this approach, it is important that one of the tool flanks is in the area covered by the camera. We assume that all the teeth of the tool are more or less equally affected by wear, so that it is enough to acquire the image of the flank of only one tooth. One of the conditions for image acquisition is to have one tooth of the tool oriented toward the camera, so that the tool has to be positioned in the tool holder accordingly. In order to acquire more accurate images, the camera is placed on an positioning device, which moves the camera in the appropriate position for each tool (Figure 1). The coordinates for each tool are stored in the tool database on the FMS controller. This controller also transmits the coordinates to the camera positioning controller. The acquisitioned images are processed on a separate computer (tool wear identification in Figure 1), which decides whether the tool is worn or not. If the system decides that the tool-flank wear exceeds the acceptable limit, the information is transmitted to the FMS, which in turn replaces the tool in the magazine or updates the information in the tool list so that a replacement tool is used from that time on, if such a replacement tool exists in the magazine. Flank wear of tools can be detected as a result of changes in the angle of the worn surface relative to the unworn flank surface. The worn surface will have a different orientation, causing the light to be reflected at a different angle and then observed on the acquired image (Figure 2).

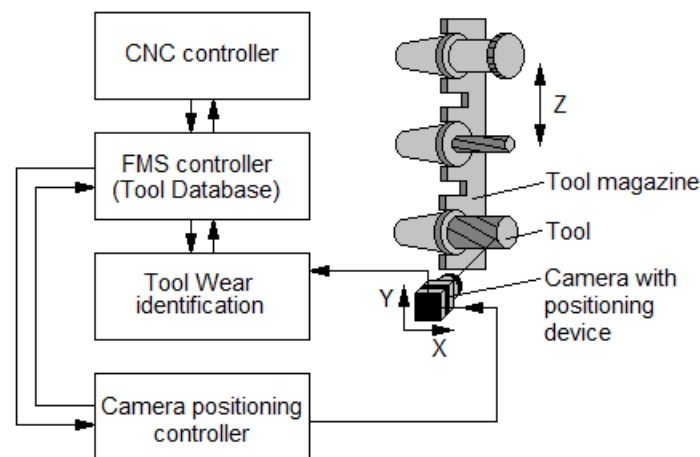


Figure 1. Tool-wear monitoring system.

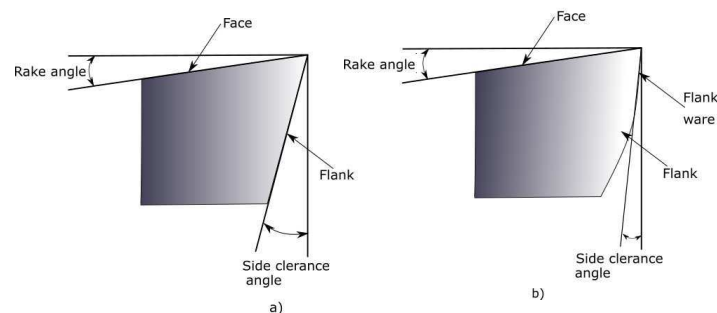


Figure 2. Tool-wear monitoring system. (a) unworn flank; (b) worn flank.

In practice, these surfaces are not ideal planes but are packed with irregularities. The new tool flank has marks as a consequence of sharpening, and the worn surface is much more irregular because of particle displacement by friction with the surface of the workpiece and chips. Thus, in the resulting images, the two surfaces (worn and unworn) are not simple to separate.

### 3. Experimental Setup for Tool-Flank-Wear Detection

The experimental system was developed with the goal to obtain consecutive images of the tool flank, which then can be used to test and analyze different image processing methods. Extensive study of flank wear for different tools, tool and workpiece materials or different processing parameters is beyond the scope of our study.

The system presented in Figure 3 is composed of a computer system (1), National Instruments CCD camera (2), optical microscope with lighting system (3), cutting tool (4), tool holder (5), and National Instruments camera source (6).



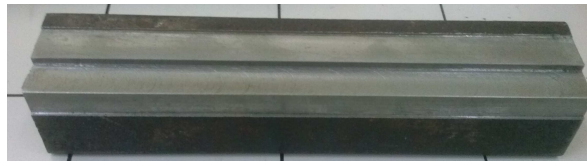
**Figure 3.** Image acquisition system for the automatic determination of tool wear.

For the experimental tests, a high-speed steel end mill HS18-0-1 (AISI T1) was used, with two helical teeth and with a diameter of 14 mm (the tool is presented in Figure 4).



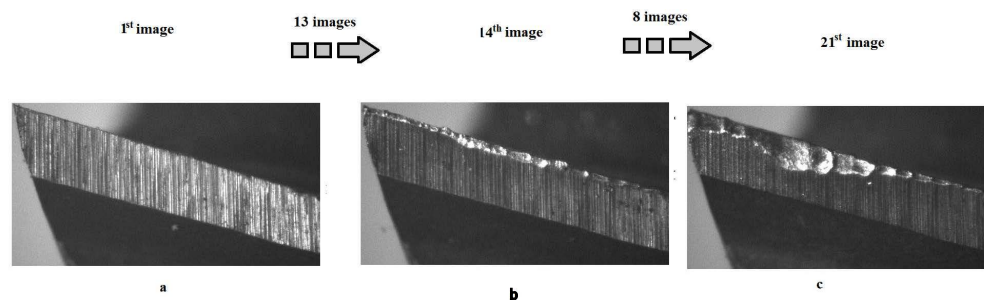
**Figure 4.** Tool used for the experimental test.

The workpiece was made of C45 carbon steel (AISI 1045) and is shown in Figure 5.



**Figure 5.** Workpiece used for the experimental test.

Using the experimental setup, images of the cutting tool were acquired. During the experiments, a total of eight complete steps of the cutting tool were made (the tool cut a total of eight entire lengths of the workpiece). Images of the tool flank were acquired after cutting every 200 mm in the workpiece. The processing parameters were set to a feed rate of 50 mm/min, a spindle speed of 500 rpm, a width of cut of 6 mm (representing 43% of the tool diameter) and a depth of cut of 2 mm. During the experiments, 21 images of the tool flank were acquired showing consecutive stages of flank-wear development. A larger number of images obtained for successive stages of wear would be generally similar for larger numbers of tools of the same type, dimension and material and under the same cutting conditions, having little effect on the results of an image processing algorithm. The reason for this is that the number and complexity of the features contained in the images of a tool flank is fairly low. The images have been analyzed and it was concluded that the wear begins to be visible at the 14th image and progresses forward from this, as can be seen in Figure 6. After the images had been acquired, three different algorithms were tested in order to find out which of these, if presented with an unlabeled image, could decide whether the tool had reached an unacceptable degree of wear.



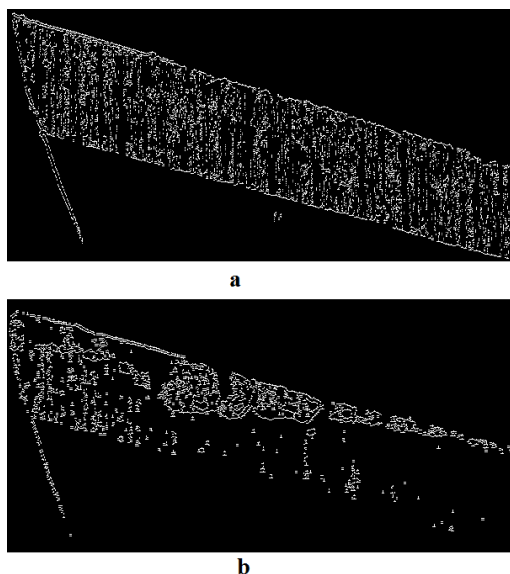
**Figure 6.** Images taken at successive times during processing showing the stages of tool wear: (a) new tool; (b) first stage when the wear became visible; and (c) last stage with massive wear.

#### 4. Image Processing for Tool-Flank-Wear Detection

##### 4.1. Image Classification Using One-Hidden-Layer ANN on Features Extracted from Image Data

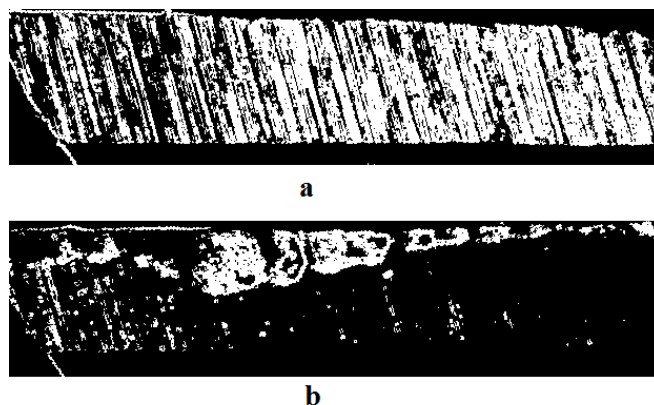
On the basis of a step-by-step analysis of the acquired images, we developed an algorithm in order to extract significant features that can help to distinguish worn and unworn tools. In the following, we describe the steps of the developed algorithm. First, the filtering of the original gray image is done with a range filter, for which each output pixel contains the maximum value minus the minimum value of a  $3 \times 3$  neighborhood of the filtered pixel. This is followed by transforming the gray-level image into a black and white (b/w) image using Otsu's method. This method establishes the threshold used to transform the image automatically from the image's gray-level histogram, so that no manual adjustment of the threshold is needed. The next step was to find edges in the b/w image using the Sobel edge detection method. Using this method, we obtained a new b/w image of the frontiers of the objects from the previous b/w image. Here, the "object" has to be understood as any white region separated from other similar regions, so that the word object has no physical meaning. The result of the above described steps is shown in Figure 7.





**Figure 7.** Binary black and white (b/w) image of (a) the new and (b) last stage of wear of the tool flank.

Starting from the observation that feature extraction can be more computationally efficient if the objects in the image are aligned to lines and columns of the internal computational representation (matrices) of the images, we computed the Hough transform of the image to find the angle of rotation of the tool edge; then we rotated the image with the angle identified by the Hough transform in order to position the edge of the tool horizontally in the image. This was followed by performing an image-opening morphological operation in order to enhance the objects in the image. The last step of the preprocessing was to apply a third-order one-dimensional median filter to the image. The result of these operations is presented in Figure 8.



**Figure 8.** Enhanced and “rotated-to-horizontal” images of (a) the flank of the new tool and (b) tool with wear.

Studying the structure of the enhanced and rotated images, we tried to use different morphological parameters to extract features that could distinguish between worn and unworn tool-flank images. One of these parameters was the Euler number (EN). The EN is the total number of objects (as defined above) in the image minus the total number of holes (dark regions surrounded by white regions).

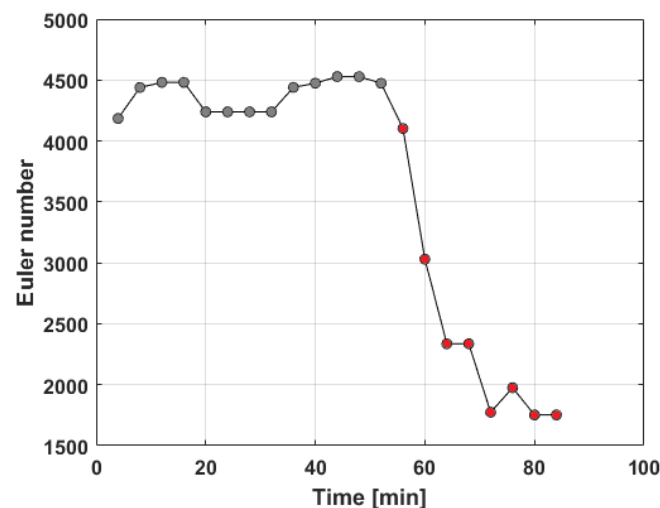
Computing the ENs for the whole set of images, we obtained the diagram in Figure 9. As can be seen in the diagram in Figure 9, the EN has a significant drop from the 14th image, which corresponds to the first image that has visible signs of wear. If we establish a threshold (in this case at about  $EN = 3500$ ), we can fairly discriminate between ENs corresponding to worn and unworn tool-flank images.

Although we obtained good results in this case, in order to consider the EN as a reliable feature, a large number of experiments with different tools should be made and thresholds for the EN diagram have to be established manually (by a human agent) for each tool, which impedes the practicality of the method. Another feature had been found by observing differences between the worn and unworn flank images. We computed the normalized sum of white pixels (denoted NSP), having the value of 1, on each horizontal line of the image. The NSP on each line is computed as the sum of pixels on that line divided by the number of pixels on the line with the maximum number of pixels. Clearly, the sum of the pixels in a line will be equal to the number of white pixels in that line, given that the black pixels have a value of 0. The NSP is defined by the expressions:

$$w_i = \sum_{j=1}^n v_{i,j} \quad (1)$$

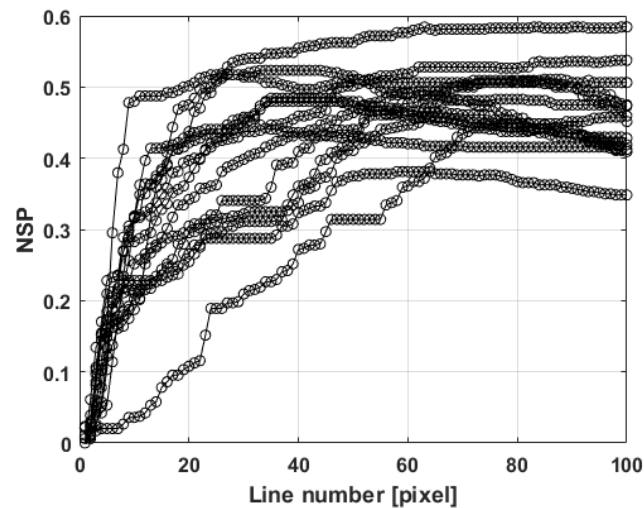
$$NSP_i = \frac{w_i}{\max(w_i)_{i=1 \dots m}} \quad (2)$$

where  $i$  is the current line with  $n$  as the number of lines in the image;  $j$  is the current column with  $n$  as the number of columns in the image;  $w_{i,j}$  is the value of the pixel intensity on line  $i$  and column  $j$ , which in this case is 0 for dark and 1 for white pixels;  $w_i$  is the sum of pixels on line  $i$ ; and  $NSP_i$  is the value of the NSP on line  $i$ .

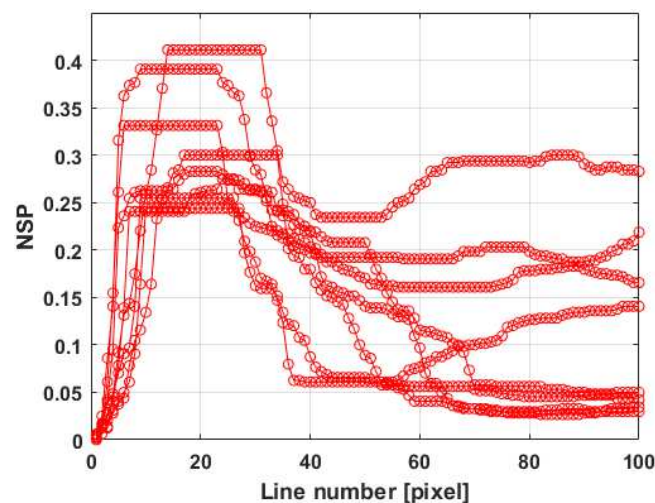


**Figure 9.** Euler number diagram for the experimental images set (points corresponding to worn tool flank are marked with red).

Computing the NSP for each image, we obtained the diagrams in Figures 10 and 11. It can be seen that the shape of the NSP diagram of worn and unworn tool flanks are quite different.



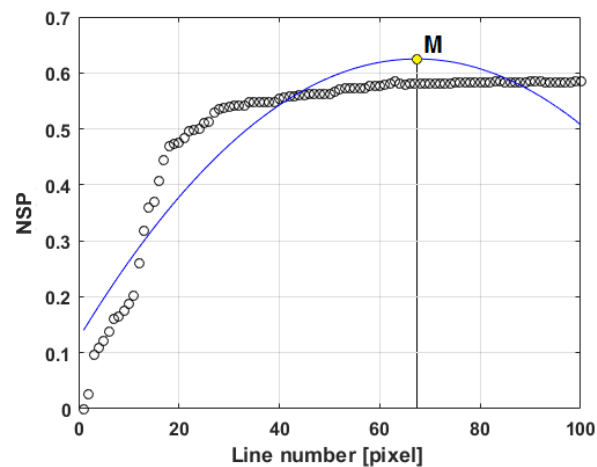
**Figure 10.** Normalized sum of white pixels (NSP) diagrams for unworn tool flank.



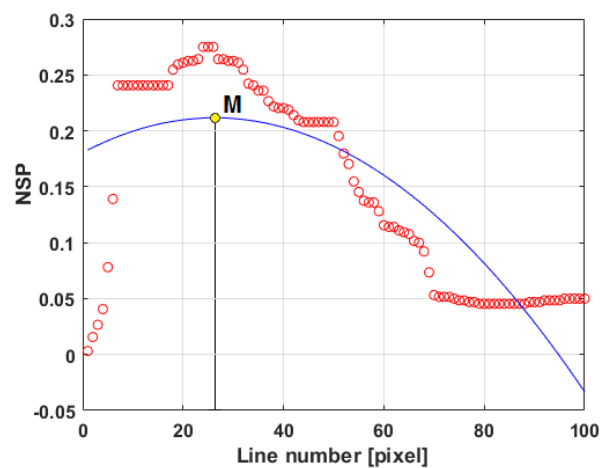
**Figure 11.** Normalized sum of white pixels (NSP) diagrams for worn tool flank.

Analyzing these diagrams, we can conclude that the NSP can be regarded as a fairly reliable feature to be used as a flank-wear detection parameter. From this point on, the main goal is to find a method to discriminate between the shapes (patterns) of NSP curves representing unworn flank images and those representing worn flank images. In order to do so, we analyzed two methods: one based on approximations with second-degree polynomials and the other based on ANN pattern recognition. The approximation of the NSP curves is shown in Figures 12 and 13. Differentiating the two types of curves are the locations (abscissa or line number  $i$ ) of their maxima: point M.



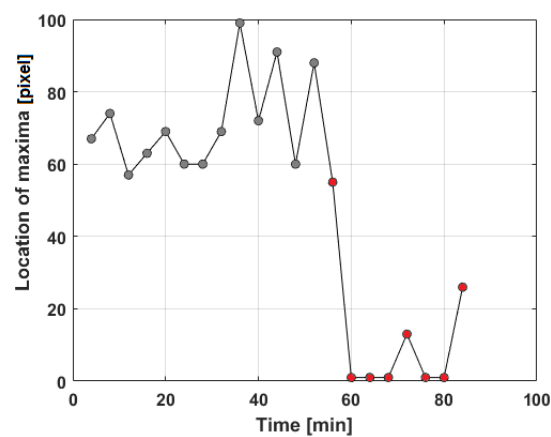


**Figure 12.** Normalized sum of white pixels (NSP) diagram for unworn tool flank and the second-order polynomial approximation (blue curve) with maxima at point M.



**Figure 13.** Normalized sum of white pixels (NSP) diagram for worn tool flank and the second-order polynomial approximation (blue curve) with maxima at point M.

Plotting the location of maxima versus the processing time until the image was acquired gives us the diagram in Figure 14.



**Figure 14.** Normalized sum of white pixels (NSP) diagram for worn tool flank and the second-order polynomial approximation (blue curve) with maxima at point M.

As in the case of the EN parameter, the location of maxima of the second-degree polynomial approximation of the NSP curve shows a good separation of worn and unworn flank images, but the manual adjustment of the threshold for the locations of maxima still remains an issue.

Another approach to classify the NSP curves is to use ANN pattern-recognition methods. For this purpose, we used the MATLAB nprtool module of the Neural Network Toolbox. The toolbox employs a two-layer feedforward network, with a sigmoid transfer function in the hidden layer and a softmax transfer function in the output layer [24]. The learning process is based on a scaled conjugate gradient backpropagation algorithm. The input data for the ANN consisted of 21 samples of 100 data points on each NSP curve, from which 13 were extracted from unworn flank images and 8 from worn flank images. These had been further divided into two groups: 13 samples (representing 8 unworn and 5 worn flanks) had been used for training and 8 samples had been used for testing (representing 5 unworn and 3 worn flanks). The input layer consisted of 100 neurons corresponding to the 100 data points of the NSP curve (Figures 10 and 11). The number of output neurons must be equal to the number of elements in the target vector, which is the number of categories of the classification process. In our case, there were two categories: worn and unworn tool-flank images. Essentially, the target vector represented the labeling of the NSP dataset with the dimension of  $2 \times 13$  for training and  $2 \times 8$  for testing.

For successful training, the training performance of the network is represented in Figure 15, which shows that, in this case, a very small error had been achieved in a short run of just 23 epochs. We can see the result of a successful testing session presented in Figure 16 in the form of a confusion matrix. In this representation, each column of the matrix represents a predicted class, while each row represents a true class. The green squares represent the correctly classified and the red squares represent the incorrectly classified samples. As can be seen in the confusion matrix, each sample, for this case, was correctly classified.

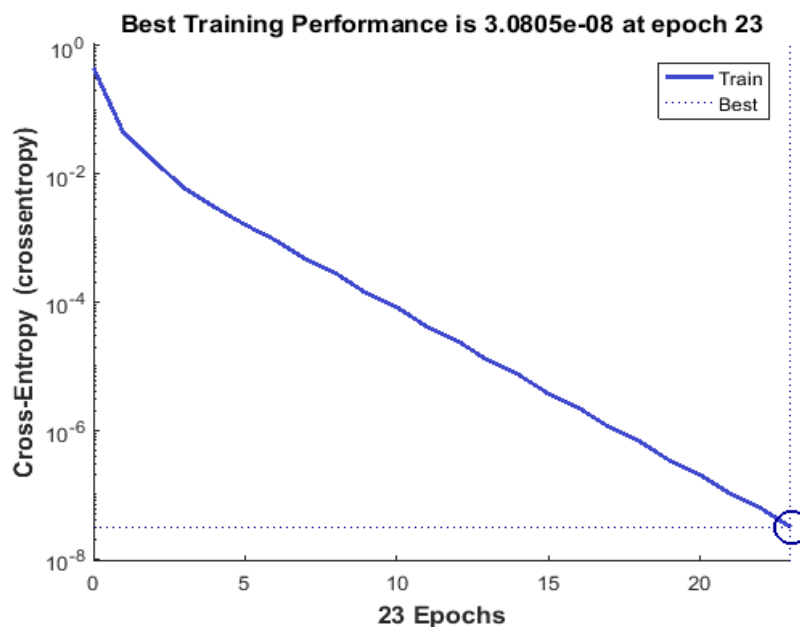
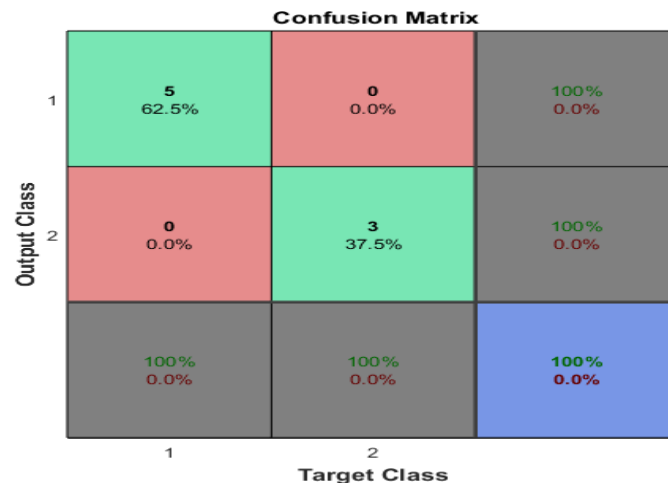


Figure 15. Training performance diagram for the artificial neural network (ANN).



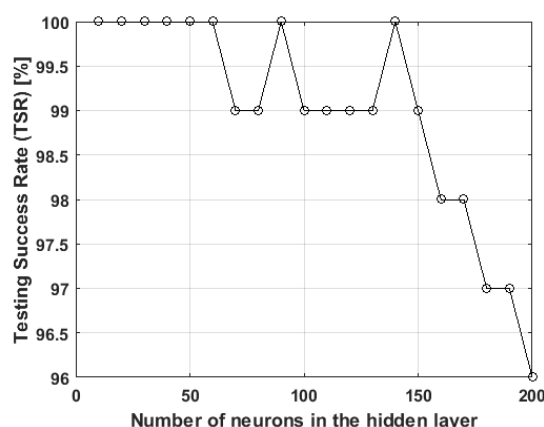
**Figure 16.** Confusion matrix for the test set.

Although working with ANN software—in our case using the MATLAB Neural Network Toolbox, but generally with other ANN software products as well—is quite straightforward, there are some specific issues which have to be dealt with:

1. The number of neurons in the hidden layer must be established on a somehow empirical basis. The number of neurons in the hidden layers may be important in order to extract the meaningful features of the image.
2. Every training session can produce different results because of the fact that the initial weights and biases of each neuron are set randomly. Training the network with the same number of neurons on the same input datasets can produce different results when tested with unlabeled data.
3. The number of training epochs has to be well established in order to avoid overfitting. If overfitting occurs, the network will be less successful in classifying unlabeled data.

In order to find the influence of these parameters on successfully training the networks, we ran training sessions multiple times with different number of neurons in the hidden layer. Every trained network was tested on the test sample set of five unworn and three worn flank images that were not used in the training sessions. For the network type described in this paragraph, we trained networks with a number of neurons in the hidden layer in the range of 10 to 200 with a step of 10 (10, 20, 30, ..., 200 neurons). A way to reduce the influence of the randomly set initial weights is to run training sessions a large number of times for the same number of neurons. For each network with a specific neuron number, we trained the network 100 times. After each training, the network was tested, and we counted the number of times the testing was successful (successfully classified every test image from the test set). We defined the TSR as the percentage of trainings that produced successful testing results from the whole number of trainings (in this case, 100). The TSR is a good indicator of the influence of neuron numbers, as the influence of initial weights is diminished by the high number of trainings. The results are presented in the diagram in Figure 17.

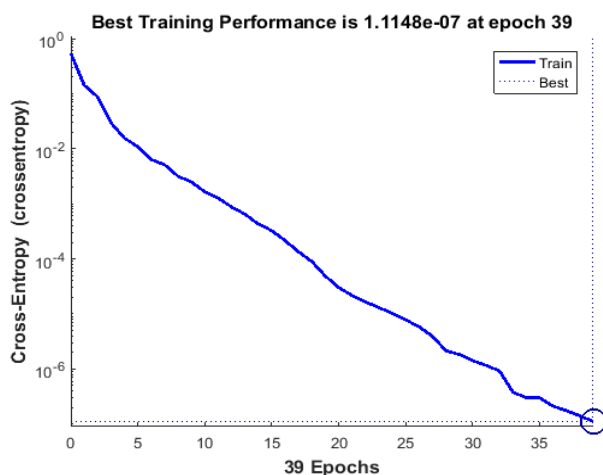
As can be seen, the number of successful trainings decreases with the increase of the number of neurons in the hidden layer. This led us to the conclusion that for this type of classification, it is better to use a small number of neurons (from 10 to 60). Using a small number of neurons is also recommended because the network will use less memory, which is important if we have a large number of types of tools which have to be classified.



**Figure 17.** Training success rate (TSR) for networks with 10 to 200 neurons in the hidden layer (100 training sessions for each network; each point in the diagram).

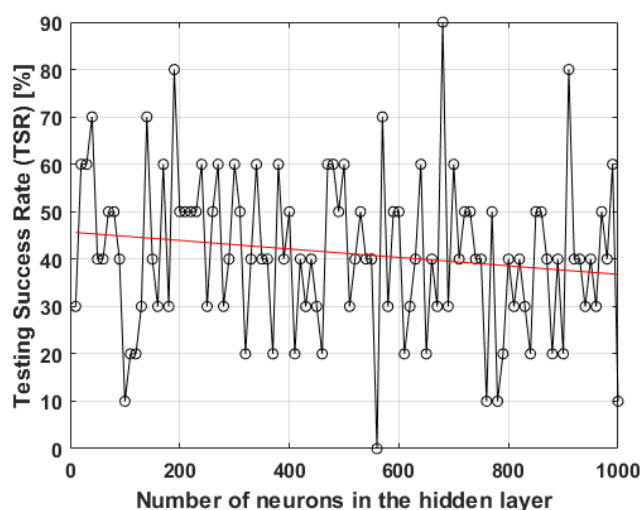
#### 4.2. Image Classification Using One-Hidden-Layer ANN on Image Data

In the above paragraph, we described the application of the ANN pattern-recognition method to discriminate between NSP features extracted from worn and unworn flank images. To avoid lengthy computations to extract features (e.g., EN or NSP) from the images, we tried to apply the pattern-recognition method directly to the images. The original images had a size of  $640 \times 480$  pixels; the input for the ANN should be a vector, which would have resulted in a size of 307,200 elements. In order to reduce the number of elements for the input vector, we resized the original image to a size of  $126 \times 94$  pixels, resulting in an input vector of 11,844 elements. The samples were divided in training and testing datasets in the same manner as described in the previous paragraph. The target vector and output layer were the same as in the previous paragraph. An example of a successful training performance for 30 neurons in the hidden layer is presented in Figure 18.



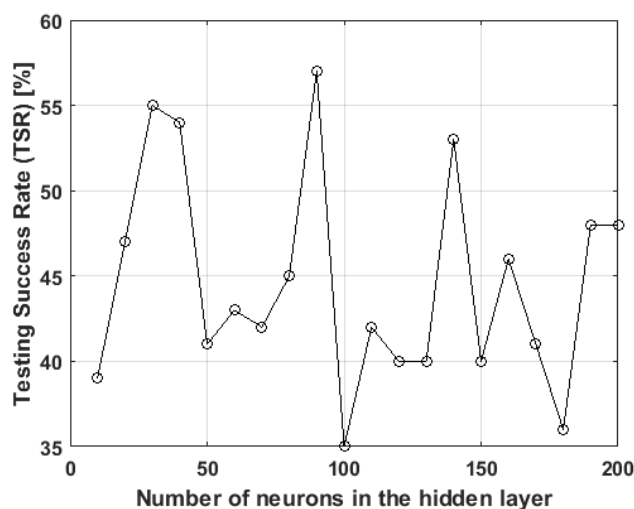
**Figure 18.** Performance diagram for the artificial neural network (ANN).

The confusion matrix in this case was similar to that in Figure 16. We also tested the TSR for this type of network. At first, we trained networks with a number of neurons from 10 to 1000 with 10 training sessions for each case, to see the overall trend of the TSR as a function of neuron numbers. The results are presented in the diagram in Figure 19. The trend has been established using a first-order (linear) approximation of the set of TSRs (the red line in Figure 19), which shows that the TSR decreases slightly with the increase in the number of neurons in the hidden layer.



**Figure 19.** Training success rate (TSR) for networks with 10 to 1000 neurons in the hidden layer (10 training sessions for each network; each point in the diagram).

Secondly, we raised the number of training sessions to 100 for the range of 10 to 200 neurons. The TSR obtained is shown in the diagram in Figure 20.



**Figure 20.** Training success rate (TSR) for networks with 10 to 200 neurons in the hidden layer (100 training sessions for each network; each point in the diagram).

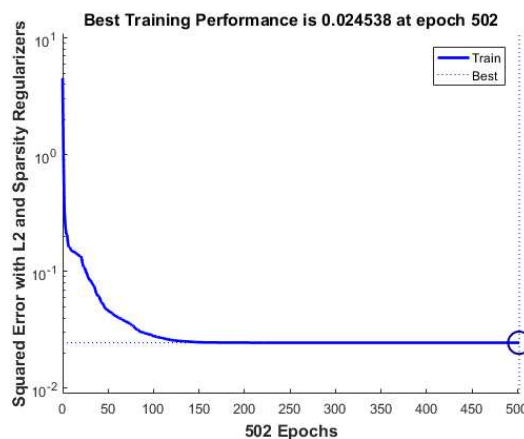
It can be observed from Figure 20 that the TSR was much less stable in this case than the TSR referenced in the previous paragraph (Figure 17). This means that there was a smaller chance to successfully train a network directly on the image data (e.g., 55% for 30 neurons) than on NSP data (for 30 neurons is 100%). In the studied range, the best TSR was for 90 neurons, at 57%.

#### 4.3. Image Classification with Autoencoders on Image Data

Deep learning is a new approach that has been introduced in the field of ANNs in the last decades by a number of researchers (e.g., [25,26]). There are a relatively large number of methods classified as deep learning, such as deep belief networks, restricted Boltzmann machines, deep autoencoders, and others. Deep learning algorithms, which use more than one hidden layer, have been successfully applied to image classification. In order to find out whether this type of network would be better

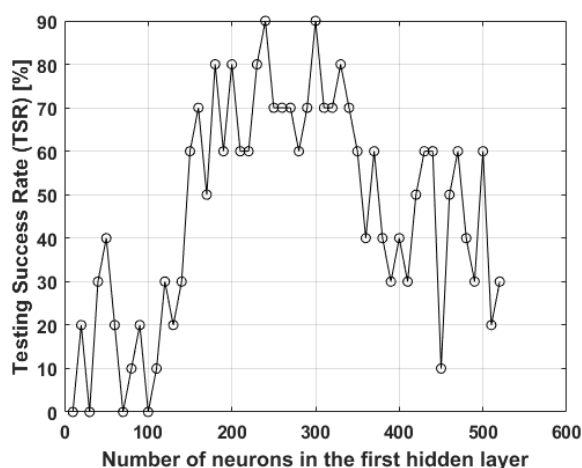


than the above presented networks, we used the MATLAB Neural Network Toolbox's Deep Learning section, which contains a module for autoencoder networks [24]. Autoencoders use methods to separately train each layer; they then stack these together in a single network with multiple layers and train the final network as a whole. As input and target data, we used the same sets as in Section 4.2. The network we used was composed of two autoencoder layers and one softmax layer as output. An example of successful training performance for 30 neurons in the first hidden layer, 10 neurons in the second hidden layer and 2 neurons in the output layer is presented in Figure 21.



**Figure 21.** Performance diagram for the second layer of a two-hidden-layer autoencoder artificial neural network (ANN).

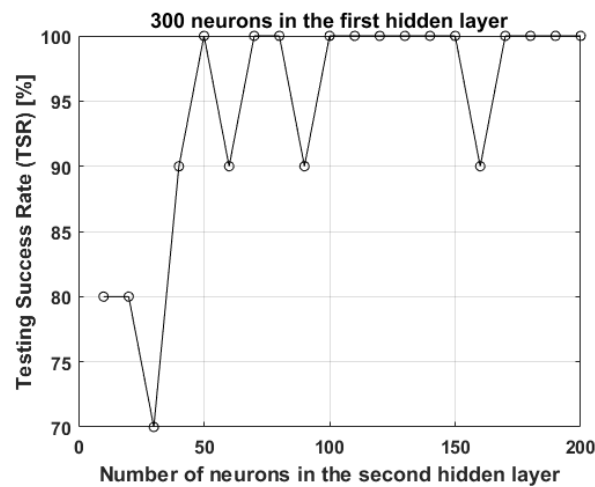
Regarding the TSR for this type of network, we trained the network for 10 to 520 neurons in the first hidden layer with a step of 10 (10, 20, 30, ..., 520) and a constant number of 10 neurons for the second layer. For each number of neurons, we trained the network 10 times, and we obtained the diagram in Figure 22.



**Figure 22.** Training success rate (TSR) for networks with 10 to 520 neurons in the first hidden layer and 10 neurons in the second layer (10 training sessions for each network; each point in the diagram).

As we can see in the diagram in Figure 22, the TSR rises abruptly at 150 neurons in the first hidden layer (increasing by more than 60%), then falls again to a mean value of about 40%. This shows that the probability to successfully train an autoencoder network, for the types of images we studied, is higher for a range of 150 to 350 neurons in the first layer. For a second set of experiments, we chose to have a fixed number of 300 neurons in the first hidden layer, which was a maximal value of 90%

TSR (Figure 22), and we ran the trainings for a range from 10 to 200 neurons in the second layer, with a step of 10, each 10 times. The results are presented in Figure 23.



**Figure 23.** Training success rate (TSR) for networks with 300 neurons in the first hidden layer and 10 to 200 neurons in the second layer (10 training sessions for each network; each point in the diagram).

The results show that for 300 neurons in the first layer and a range of 100 to 150 neurons in the second layer, we could obtain a TSR of 100%.

## 5. Conclusions

Our goal in this study was to analyze a tool-flank-wear monitoring system according to the paradigm described in Section 2 (Figure 1). The hardware needed to develop such a system is relatively simple to implement; there are a large number of suppliers for machine vision systems and positioning devices. The tool database can be easily extended with the specific parameters for tool-wear monitoring, with the observation that for a large database, the number of parameters has to be kept as low as possible. The main questions we focused on were the following:

1. What kind of image processing and classification method would be successful?
2. What are the costs for such a system to be implemented?

Regarding the first question, we developed and tested three methods, which used ANNs to classify the tool-flank images: one based on image-processing feature extraction followed by ANN classification, and the other two methods applying ANNs directly on the image data. A comparison of the three methods' performances is presented in Table 1.

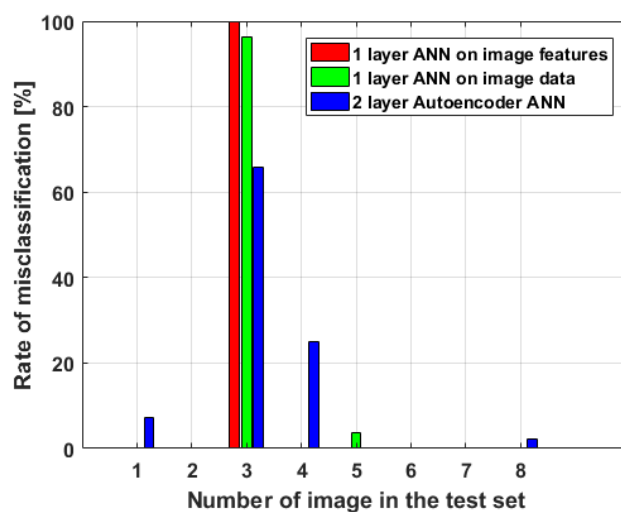
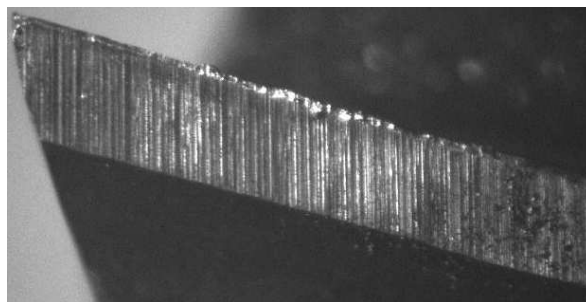
For method A, the image processing time to extract features was 4.2 s for all images of the image set (21 images). The computational times for network training are given in Table 1. For method A, the preprocessing time had to be added (4.2 s). As our focus was to compare only the presented algorithms, these are the only important computational times. Method A has the highest TSR, which suggests that this is the most reliable of the three methods tested. It also has the smallest training time and the smallest number of epochs needed for training. Method B is less attractive mainly because of the smaller TSR, which means that there may be a greater number of training sessions needed to develop a successful network. Even so, its advantage is that it is simpler to implement and requires a small number of neurons. Comparing the TSR for methods B and C, both working directly on the image data, we can conclude that autoencoders perform better than single-hidden-layer networks for large numbers of neurons. Method C requires the largest number of neurons, the largest number of epochs and the longest time to train.

**Table 1.** Comparative network training parameters.

Type of Network	No. of Neurons	Average No. of Training Epochs	Average Training Success Rate	Average Training Time (s)
A. Single hidden layer on image features	10	15–20	100	0.20
	200	25–30	96	0.25
B. Single hidden layer on image data	10	30–40	46	0.75
	200	30–40	42	12
C. Two autoencoder hidden layers (L1, L2) on image data	L1, L2	L1, L2	L1 + L2	L1 + L2
	10 to 140, 10	1000, 1000	15	280
	150 to 200, 10	1000, 1000	70	1900
	300, 100 to 150	1000, 1000	100	1900

Our experiments show that the TSR increases with the increase in the number of neurons.

During the training of the analyzed networks, we counted the percentage of misclassifications for each image in the testing set from the total number of misclassifications occurring for each of the three network types, with the same number of training sessions as in the case of the TSR. The result is presented in Figure 24. It can be observed that image number 3 from the training set (Figure 25) had the highest number of misclassifications regardless of the network type or number of neurons, which means that greater care has to be taken in the selection of the training set in order to produce a good result during operation.

**Figure 24.** Number of misclassifications of tool-flank images from the total number of misclassifications.**Figure 25.** Image of sample number 3 from the test set labeled as unworn tool flank but misclassified, in some cases, as worn tool flank.

Regarding the second question, our assessment is that the major parts of the costs to implement this tool-wear monitoring solution are the cost of image acquisition, the cost of developing the software, and training the decision-making system to discriminate between the unworn and worn tools. The image-acquisition process costs increase with the number of acquisitioned images. To reduce these costs, the method that is successful when trained with a small number of images is recommended. Although it seems that method A gives the best results, an image-processing specialist may be needed to develop the feature extraction algorithm, as there is a large amount of work spent on the manual adjustment of parameters. Additionally, the feature extraction algorithm may be different for different types of tools, further increasing the development costs.

For methods B and C, no image processing is needed before applying the ANN classification. If ANN software is available (MATLAB Neural Network Toolbox or other similar software product), the development process is relatively straightforward with few parameters to adjust (e.g., number of neurons in the hidden layer); thus the training process does not require highly qualified software experts. A discussion has to be made also on how many networks should be employed: Is it possible to use one network for a larger set of tools or does each tool have to be provided with its own network? The implementation of the system is made gradually as the manufacturing system is running, without interrupting the production process. The only interruption of the production is for the time needed to install the camera and the positioning device. After installing the camera and the positioning device, images are acquired without interrupting the production process. In time, as the tools are used to process the parts, an image database is developed. As soon as the image database for a specific tool is complete, the training stage is accomplished during the runtime of the manufacturing system. When the training has produced a reliable neural network, this is implemented in the system. In this paper, we described the concept of a new system based on image acquisition and processing of the tool flank. The system is capable of automatically detecting tool wear in an early stage. Regarding the image processing methods, we present two new methods to obtain image features, which make the discrimination between worn and unworn tool-flank images possible (EN and NSP). We applied, to the best of our knowledge for the first time, a classification of the worn and unworn tool-flank images using a two-hidden-layer autoencoder ANN, which proved to be 100% successful for a large range of the number of nodes. We also present a detailed comparison of three ANNs in order to establish their capabilities to classify worn and unworn tool-flank images.

**Acknowledgments:** 1. This research was financed by: POSDRU 107/1.5/S/77265 (2010) of the Ministry of Labor, Family and Social Protection, Romania, co-financed by the European Social Fund—"Investing in people". 2. The publishing sponsor was R & D center "Cercetare Dezvoltare Agora Oradea".

**Author Contributions:** The contribution of the first three authors was in engineering issues and of the last two authors was ANN modeling.

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

## References

1. Denkena, B.; Krüger, M.; Schmidt, J. Condition-based tool management for small batch production. *Int. J. Adv. Manuf. Technol.* **2014**, *74*, 471–480.
2. Çelik, Y.H.; Kilickap, E.; Güney, M.J. Investigation of cutting parameters affecting on tool wear and surface roughness in dry turning of Ti-6Al-4V using CVD and PVD coated tools. *J. Braz. Soc. Mech. Sci. Eng.* **2016**, *1–9*, doi:10.1007/s40430-016-0607-6.
3. Jain, V.; Raj, T. Tool life management of unmanned production system based on surface roughness by ANFIS. *Int. J. Syst. Assur. Eng. Manag.* **2016**, *1–10*, doi:10.1007/s13198-016-0450-2.
4. Bhat, N.N.; Dutta, S.; Vashisth, T. Tool condition monitoring by SVM classification of machined surface images in turning. *Int. J. Adv. Manuf. Technol.* **2016**, *83*, 1487–1502.
5. Huang, P.B. An intelligent neural-fuzzy model for an in-process surface roughness monitoring system in end milling operations. *J. Intell. Manuf.* **2016**, *27*, 689–700.

6. Li, N.; Chen, Y.; Kong, D. Force-based tool condition monitoring for turning process using v-support vector regression. *Int. J. Adv. Manuf. Technol.* **2016**, 1–11, doi:10.1007/s00170-016-9735-5.
7. Rimpault, X.; Chatelain, J.-F.; Klemberg-Sapieha, J.E.; Balazinski, M. Tool wear and surface quality assessment of CFRP trimming using fractal analyses of the cutting force signals. *CIRP J. Manuf. Sci. Technol.* **2017**, 16, 72–80.
8. Wang, G.; Guo, Z.; Qian, L. Tool wear prediction considering uncovered data based on partial least square regression. *J. Mech. Sci. Technol.* **2014**, 28, 317–322.
9. Khajavi, M.N.; Nasernia, E.; Rostaghi, M. Milling tool wear diagnosis by feed motor current signal using an artificial neural network. *J. Mech. Sci. Technol.* **2016**, 30, 4869–4875.
10. Arruda, E.M.; Ribeiro Filho, S.L.M.; Assunção, J.T.; Brandão, L.C. Online prediction of tool wear in the milling of the AISI P20 steel through electric power of the main motor. *Arab. J. Sci. Eng.* **2015**, 40, 3321–3328.
11. Postnov, V.V.; Idrisova, Y.V.; Fetsak, S.I. Influence of machine-tool dynamics on the tool wear. *Russ. Eng. Res.* **2015**, 35, 936–940.
12. Stavropoulos, P.; Papacharalampopoulos, A.; Vasiliadis, E.; Chrysosolouris, G. Tool wear predictability estimation in milling based on multi-sensorial data. *Int. J. Adv. Manuf. Technol.* **2016**, 82, 509–521.
13. Zhang, G.; To, S.; Zhang, S.H. Evaluation for tool flank wear and its influences on surface roughness in ultra-precision raster fly cutting. *Int. J. Mech. Sci.* **2016**, 118, 125–134.
14. Cerce, L.; Pusavec, F.; Kopac, J. 3D cutting tool-wear monitoring in the process. *J. Mech. Sci. Technol.* **2015**, 29, 3885–3895.
15. Garcia-Ordás, M.T.; Alegre, E.; González-Castro, V. A computer vision approach to analyze and classify tool wear level in milling processes using shape descriptors and machine learning techniques. *Int. J. Adv. Manuf. Technol.* **2016**, 1–15, doi:10.1007/s00170-016-9541-0.
16. Javed, K.; Gouriveau, R.; Li, X. Tool wear monitoring and prognostics challenges: A comparison of connectionist methods toward an adaptive ensemble model. *J. Intell. Manuf.* **2016**, 1–18, doi:10.1007/s10845-016-1221-2.
17. Kong, D.; Chen, Y.; Li, N. Tool wear monitoring based on kernel principal component analysis and v-support vector regression. *Int. J. Adv. Manuf. Technol.* **2016**, 1–16, doi:10.1007/s00170-016-9070-x.
18. Chetan, E.; Narasimhulu, A.; Ghosh, S.; Rao, P.V. Study of tool wear mechanisms and mathematical modeling of flank wear during machining of Ti alloy (Ti6Al4V). *J. Inst. Eng. (India)* **2015**, 96, 279–285.
19. Mia, M.; Al Bashir, M.; Dhar, N.R. Modeling of principal flank wear: An empirical approach combining the effect of tool. *Environ. Workpiece Hardness J. Inst. Eng. (India)* **2016**, 97, 517–526.
20. Yang, W.A.; Zhou, W.; Liao, W. Prediction of drill flank wear using ensemble of co-evolutionary particle swarm optimization based-selective neural network ensembles. *J. Intell. Manuf.* **2016**, 27, 343–361.
21. Muratov, K.R. Influence of rigid and frictional kinematic linkages in tool–workpiece contact on the uniformity of tool wear. *Russ. Eng. Res.* **2016**, 36, 321–323.
22. Park, K.H.; Yang, G.D.; Lee, D.Y. Tool wear analysis on coated and uncoated carbide tools in inconel machining. *Int. J. Precis. Eng. Manuf.* **2015**, 16, 1639–1645.
23. Yingfei, G.; Muñoz de Escalona, P.; Galloway, A. Influence of cutting parameters and tool wear on the surface integrity of cobalt-based stellite 6 alloy when machined under a dry cutting environment. *J. Mater. Eng. Perform.* **2017**, 26, 312–326.
24. Mathworks®. MATLAB, Neural Network Toolbox, Image Processing Toolbox, R2016b, User's Guide. Available online: <https://www.mathworks.com/help/> (accessed on 4 November 2017).
25. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, 18, 1527–1554.
26. Bengio, Y.; Lamblin, P.; Popovici, D.; Larochelle, H. Greedy layer-wise training of deep networks. In Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 4–7 December 2006.

