Deep learning methods for drill wear classification based on images of holes drilled in melamine faced chipboard

Albina Jegorowa¹ · Jarosław Kurek² · Izabella Antoniuk² · Wioleta Dołowa³ · Michał Bukowski³ · Paweł Czarniak¹

Received: 18 April 2020 / Accepted: 4 November 2020 / Published online: 23 December 2020
© The Author(s) 2020

Abstract

In this paper, a set of improvements made in drill wear recognition algorithm obtained during previous work is presented. Images of the drilled holes made on melamine faced particleboard were used as its input values. During the presented experiments, three classes were recognized: green, yellow and red, which directly correspond to a tool that is in good shape, shape that needs to be confirmed by an operator, and which should be immediately replaced, since its further use in production process can result in losses due to low product quality. During the experiments, and as a direct result of a dialog with a manufacturer it was noted that while overall accuracy is important, it is far more crucial that the used algorithm can properly distinguish red and green classes and make no (or as little as possible) misclassifications between them. The proposed algorithm is based on an ensemble of possibly diverse models, which performed best under the above conditions. The model has relatively high overall accuracy, with close to none misclassifications between indicated classes. Final classification accuracy reached 80.49% for biggest used window, while making only 7 critical errors (misclassifications between red and green classes).

Introduction

The furniture manufacturing process is a complicated one, with many different stages requiring high precision and well thought actions. Each of the decisions, that ends up in poor quality product or some errors in its general outline can result in huge losses for the company, since such elements will not meet overall requirements.

✉ Albina Jegorowa
albina_jegorowa@sggw.edu.pl

¹ Institute of Wood Sciences and Furniture, Warsaw University of Life Sciences, Warsaw, Poland

² Institute of Information Technology, Warsaw University of Life Sciences, Warsaw, Poland

³ Warsaw, Poland

Out of many factors, one that is very important and also hardest to measure without user input is monitoring the drill sharpness in order to point out the exact moment, when it should be replaced with a new one. Accurate prediction of this point can prevent some budget losses due to poor product quality. While the operator is able to manually check the drill state, this is not efficient enough, and more automatized solution is required to speed up the entire process.

From existing solutions, the entire section of algorithms and methodologies used for evaluation and assessment of various utensils is called tool condition monitoring (TCM). From those solutions, there also are quite few that focus on drill condition monitoring. In most of those approaches, usually large quantities of diverse sensors are required to measure different signals. Data obtained in that way are later used to diagnose each drill state. The most common signals used for such methods contain noise measurement, testing vibrations, inspecting acoustic emission or elements such as cutting torque or feed force (Kurek et al. 2015). Approaches involving above methods for obtaining measurements used to assess drill state can produce accurate results. Unfortunately, they also usually require lengthy and numerous preprocessing stages, before required precision is reached. Furthermore, any mistakes during those stages (like appropriate sensor and signal choice, generating and selecting best features used for diagnostics or finally building the classification model) can lead to the final result not being acceptable. Such solutions also are very complex in terms of setting them up and configuring each element, while required equipment tends to be expensive to purchase or rent. Additionally, the preparation time required to, for example, set up and calibrate sensors in case when such setup cannot stay in its unfolded state needs to be taken into consideration. If both the time required to prepare the test stand and computation time needed before the initial results can be obtained are significant, the usability and convenience of such solution are further diminished. All of the above disadvantages make most of existing solutions focusing on tool condition monitoring difficult to use, with no immediate gain for the company that decides to use them, or even certainty about such gains to exist in the near future. Experiments in previous works (Jemielniak et al. 2012; Kuo 2000; Panda et al. 2006) show, that even when taking into account those diverse features, generated based on a diverse set of registered signals, accuracy of such solutions did not exceed 90% while recognising three defined classes.

Application of machine learning algorithms in wood industry is rapidly increasing nowadays. For example in Ibrahim et al. (2017), the authors prepared a custom algorithm for wood species recognition, based on macroscopic texture images incorporating fuzzy-based database management. They divided the data set into 4 parts, based on pore size in each sample, and used SVM (support vector machine) for final classification. This allows easier addition of new wood species (that way, when the new class is added, only the SVM classifier for that part of database needs to be retrained).

When samples in the form of images are taken into consideration, convolutional neural networks “CNN” are often used (Hu et al. 2019; Kurek et al. 2017a, b, 2019a, b). Despite its increased popularity, there are some problems with adapting deep learning solutions to the presented problem. First of all, for most of the existing algorithms, large amount of training data is required, which is not always readily or

easily available. Usually, the situation is completely opposite, with only small initial data set. Secondly, depending on the given problem, the requirements might differ, so each task needs close cooperation with the manufacturer, or else the final solution might not be satisfactory, even with high accuracy of output results.

Basing initial approaches on the problem of existing works (Bengio 2009; Deng and Yu 2014; Schmidhuber 2015) and later on performed experiments (Kurek et al. 2017a, b, 2019a, b), which eventually led to the work presented in this paper, the authors focused on decreasing overall solution complexity, as well as its adjustment to specific manufacturer requirements. A first major improvement is exclusion of any more complex equipment, that requires calibration or adjustments. The only external element which is still utilized in the presented approach is a camera used for taking pictures of drilled holes (which are later used to assess the overall drill state). Such system is easy to implement in any production setup and does not require large financial investment from the company, before initial measurements can be taken. During previous approaches to this problem, different types of algorithms were tested, using convolutional neural networks (CNN, which is considered one of the most effective solutions, which also does not require specialized diagnostic features (Goodfellow et al. 2016; Kurek et al. 2017a, b) with transfer learning and data augmentation methodologies (Kurek et al. 2019a, b). The performed experiments not only confirmed, that the chosen solution can accurately predict the state of drill based only on shown pictures, but also were able to increase overall prediction accuracy in relation to methods using more sophisticated signals during evaluation of tool state. Since after such solution is prepared it doesn't require specialized knowledge to use (like it is the case when sensors need to be set up and calibrated), it can also be more easily adapted to the specific requirements of different manufacturers. While specialized equipment still might be required, i.e. if the manufacturer wanted to monitor the state of the drill on a continuous basis (specialized monitoring system, adjusted to taking clear pictures of succeeding holes would be required in that case), it also allows an approach that doesn't include such elements, and the state of the drill is checked every set number of drillings. Taking all of the above into consideration, solutions based on CNN networks are cheaper in terms of initial equipment costs and easier to implement, while achieving similar to or higher accuracy than more complex solutions based on sets of different sensors.

The first of the previous works considered in the current approach (Kurek et al. 2017b) focused on applying CNN to the problem of drill wear prediction with a very limited set of training data (242 images representing three classes: 102 green samples, 60 yellow samples and 80 red samples). Even with such limited collection, accuracy of 85% was reached with the presented CNN algorithm, using AlexNet model (Krizhevsky et al. 2012; Russakovsky et al. 2015; Shelhamer 2017), and the accuracy was additionally increased to 93.4% by using Support Vector Machine (SVN) as final CNN layer. In Kurek et al. (2017a), simple image operations were used to artificially expand the available data set. In this approach, only two classes were recognized, and the initial data set contained a total of 900 images (300 samples for first class and 600 samples for second one). The original system had 66.6% accuracy, but after expanding the available data set by using rotation, scaling, adding noise, etc., this value increased as well. Final results amounted to 89% accuracy

for 11700 samples and 95.5% for 33300 samples. The main disadvantage in that case was rather lengthy learning process, which lasted over 20 hours. The third solution (Kurek et al. 2019b) combined the ideas of data augmentation and using pretrained CNN for image classification. In that approach, pretrained CNN using AlexNet model was used, and initial training set, which contained 242 images was expanded using simple image operation (final set contained 4598 images). Accuracy of this solution equalled to 93.48%. The final approach presented in Kurek et al. (2019a) used a classifier ensemble containing three of the most popular pretrained CNN networks: AlexNet, VGG16 and VGG19, and achieved accuracy of 95.65% (while individual classifiers achieved accuracy of 94.30% for AlexNet, 92.39% for VGG16 and 96.73% for VGG19).

Even though all of the above approaches show increasing accuracy, without unnecessary prolonging computation time or complicating equipment setup, during those experiments one additional problem occurred. While overall algorithm accuracy is still an important factor, it turned out that for the furniture companies, the most important parameter concerns avoiding any misclassifications between red and green classes (calling those “critical errors”). While tool denoted with a yellow state is further evaluated by human expert, and therefore any errors with evaluation can be corrected, red and green classes are definitive. Therefore green tool, classified as red will be discarded, even though it could still be used for long period of time, and what’s even worse, red tool classified as green will be further utilized, possibly causing some financial loss due to poor product quality. In the current approach, the main focus is put mainly on this misclassification problem, trying to avoid, especially, assigning green class to red tools, while maintaining possibly high overall classification accuracy rate. Few different classifiers were chosen and tested, as well as classifier ensembles. The best of them was further modified, to improve the obtained results according to no green-red misclassification requirement.

The rest of this work is organized as follows. Methodologies used for data augmentation and adjustments, as well as data set preparation for different methods are described in Sect. [Materials](#). The section [Data preparation](#) presents setup and operations made during initial stages. The section [Methods](#) presents an overview of different algorithms tested during experiments and final methodology, based on best results from original methods set. An overview of obtained results and some possible areas of future work are presented in Sect. [Experiment results and discussion](#). The last section summarizes the work described in this paper and gives some concluding remarks.

Materials

The material used in the drilling process was a standard laminated chipboard (Kronopol U 511 SM; Swiss Krono Sp. z o. o., Żary, Poland), that is typically used in furniture industry (Fig. 1a). Dimensions of the test piece were $300 \times 35 \times 18$ mm. For the drilling process, regular 12 mm drill (FABA WP-01; Faba SA, Baboszewo, Poland) equipped with a tungsten carbide tip was used (Fig. 1b). The holes were made in a three-layer laminated chipboard at a spindle speed of 4500 rpm

Fig. 1 Example materials used during drilling process: side section of melamine faced chipboard (a) and example drill bit (b)

and a feed speed of 1.35 m/min (in accordance with the drill bit manufacturer's recommendations).

There are different approaches to tool condition monitoring, each related to various machine parts and their specifics, but usually in those cases three classes are considered for labelling the general state of tested element: red, green and yellow (Jegorowa et al. 2019, 2020). The first class describes tools that are in a poor state, and should be immediately replaced because of it. The second set is the exact opposite of it and consists of tools that are in a good shape and can be used further, without any possible losses for the company. The final class denotes tools that are suspected of being worn out, possibly requiring additional, manual evaluation from the human expert to definitively determine their state—depending on given opinion, they can be either discarded as not good enough, or used further during production process if their state is still satisfactory. The extent of the wear of the external corner [W (mm)] was adopted as a drill condition indicator and periodically monitored using a microscope (Mitutoyo TM-500) (Fig. 2). Based on this, the drill immediately got classified as “Green” (for $W < 0.2$ mm), “Yellow” ($0.2 \text{ mm} < W < 0.35 \text{ mm}$) or “Red” ($W > 0.35 \text{ mm}$).

Fig. 2 Example images obtained during monitoring process with microscope used for drill external corner wear (Mitutoyo TM-500)

Data acquisition

The data samples with images of drilled holes used during training process were collected in cooperation with Institute of Wood Sciences and Furniture at Warsaw University of Life Sciences, using standard CNC vertical machining centre (Buselato Jet 100, Thiene, Italy). For test purposes, the drilling process was performed on standard, melamine faced chipboard (U511SM—Swiss Krono Group), that is typically used in furniture industry. Dimensions of the test piece were $2500 \times 300 \times 18$ mm, used for drilling process (12 mm Faba WP-01 drill with a tungsten carbide tip was used). The test piece with drilled holes was later cut into smaller ones (with size $300 \times 35 \times 18$ mm) and photographed (using Nikon D810, Nikon Corporation, Shinagawa, Tokyo, Japan).

A total of five new drills were used during that process. During the experiment, each tool reached a wear level above 0.35 mm, which qualifies them for replacement with a new one. Measurement of wear on the microscope took place every 140 holes. Since the drills are double-edged, the microscope was measured separately for each of the blades, and then the arithmetic mean wear of both blades was determined. For each of the drills, the process was repeated in cycles, while monitoring external corner wear parameter between each cycle in order to assign the appropriate class to the images obtained. Images of holes made by each drill were stored separately, while maintaining the order in which they were made (since it shows gradual drill wear state degeneration and could be used as additional information during learning process). Table 1 outlines the data acquisition process, with measured final corner wear for each drill at the end of the last drilling cycle. Figure 3 presents examples of images representing each of the drill wear classes used in the current approach.

Data preparation

The data set used during the training process was similar to that in previous works (Kurek et al. 2017b, 2019a, b), but in this case it was larger, containing 5 sets of images, one for each of the drills used, with a total of 8526 samples: 3780 representing green class, 2800 for yellow class and 1946 for red class. Since green class was significantly better represented (almost half of the samples), there

Table 1 Summary of data acquisition process, with initial measurements

	No. of cycles	Value of final wear [mm]	No. of drillings
Drill No. 1	12	0.321	1666
Drill No. 2	13	0.420	1820
Drill No. 3	12	0.321	1680
Drill No. 4	12	0.368	1680
Drill No. 5	12	0.397	1680

Fig. 3 Examples of holes produced by each drill wear class: green, yellow and red

was a possibility that such lack of balance in the data set could result in poor accuracy after training, therefore it was decided to optimize the data set. During initial preparations, it was ensured that each set contained the same number of samples for each class. Rotation by 180 degree was used to obtain assumed number of samples. Table 2 contains initial quantities of samples for each class in set representing each of the used drills, before and after using data augmentation. This operation makes each collection balanced, ensuring the training process will run correctly. In the performed experiments, one of the initial sets was used for testing, while the remaining ones were used during the training process. Each

Table 2 Sample counts for each class before and after data augmentation

No. of drill	Original	Augmented	Total
1	840/420/406	840/840/840	2520
2	840/700/280	840/840/840	2520
3	700/560/420	700/700/700	2100
4	840/560/280	840/840/840	2520
5	560/560/560	560/560/560	1680

Values in each cell are presented in the following order: Green, Yellow, Red

training set was additionally divided, so that 90% samples were used for actual training, while remaining 10% samples were used for validation.

Methods

In previous experiments (Kurek et al. 2017a, b, 2019a, b), even small adjustments to different algorithms resulted in significant differences in final accuracy of presented solution. Therefore, to ensure that the method used for the current approach was best suited (especially since the main factor apart from overall accuracy concerned misclassification rate between green and red classes), initial experiments concerned testing different algorithms in chosen aspects, starting with solutions, that gave satisfactory results in previous approaches. Convolution neural networks (CNN) were used for all experiments, with different pretrained models as their base, or by training them from scratch. For each experiment, threshold of 500 training epochs was used, but none of the tested algorithms actually reached that number (early stop method was implemented, to make sure, that the entire process will stop, when no significant increase in model accuracy happens). Validation accuracy was monitored during each training epoch with parameter patience = 50. It means that if there was no increase in validation accuracy during next 50 epochs, the algorithm was stopped and the best model saved.

All experiments were performed using two NVIDIA TITAN RTX graphics card (with total 48 GB of RAM), using Tensorflow platform with Python and keras library and applying Adam optimizer.

Data sets for numerical experiments

Drill wear classification is in the focus of the authors research for quite some time. During the performed experiments, extensive dialog with manufacturer and experts was conducted, regarding necessary features of the system used for tool wear recognition. During this communication, few additional requirements were established, that were omitted in previous attempts, and to the authors knowledge are ignored in general when it comes to the presented problem.

First of all, although overall accuracy of any prepared solution is important, algorithm's ability to clearly distinguish red and green classes is far more crucial. Mistakes between those two classes are referred to as critical errors, since whenever red class is classified as green or green class is classified as red, such situations have a high potential of generating financial loss for the manufacturer. In the first case, the product obtained during production process has a high chance to be of poor quality. In the second case, a good tool will be prematurely discarded. Both cases are highly undesirable.

The second discussed element was overall performance and time required to produce results. While deep learning algorithms are more often applied to problems such as tool condition monitoring, there are a series of issues connected to the overall training process and data acquisition. First of all, the algorithm should be able to work correctly with limited amount of data, since collecting large amounts of input files is not always possible. Secondly, even while having large amounts of training data, total time required to train the usable solution should also be considered. This element is reflected in algorithm choice in the current approach, as well as amount of data augmentation used (this methods are only used to balance the entire set, so each class is equally represented, but total number of samples are not further increased, to avoid unnecessary prolongment of the training process). Final division of the training data used for current experiments is outlined in Table 3.

Since artificial data set extension was not considered in that case, an additional method was used to obtain information about changes in drill state. For that purpose, window parameter was introduced, based on the way the images were stored. In the initial set, consecutive samples were stored in exact order in which they were made, thus showing gradual changes in overall drill state. It was assumed that the used algorithm will benefit from such information, and instead of using single image as input, sets of different sizes were incorporated, to show gradual wear state degradation. In the current approach, after consultations with experts, windows of size 5, 10, 15 and 20 were tested in comparison with the original approach without such extension.

During the performed experiments, different algorithms were evaluated for a given set of requirements. Each of them was separately prepared and tested. Final solution developed in this approach is a result of those tests, confronted with manufacturer requirements and overall results obtained during different stages for chosen algorithms. In the following subsections, the tested algorithms are outlined.

Table 3 Division of data sets taking part in numerical experiments

#	No. drills-training	No. drills-testing	No. trials-training	No. trials-testing
1	2 + 3 + 4 + 5	1	9000	2520
2	1 + 3 + 4 + 5	2	9000	2520
3	1 + 2 + 4 + 5	3	9420	2100
4	1 + 2 + 3 + 5	4	9000	2520
5	1 + 2 + 3 + 4	5	9660	1680

Baseline: image grayscale analysis

During initial preparations, a baseline method for measuring accuracy and critical error rate was prepared, based on image colour distribution. Each of the images was converted to grayscale, and resized to 500 by 500 pixels, in order to obtain comparable samples. While simple image operations can be used in case of CNN algorithms to even out the data set, it is not the case with this approach (since even after rotating the image, the pixel distribution will remain the same), hence in case of this solution the initial set was used, without any alterations.

After the grayscale conversion, values were normalized to feed between 0–1 range. Later, pixels were grouped according to their value, and assigned to one of the three sets: black (representing the hole), white (representing the laminated chipboard) and grey (representing the edge). While the tool is degrading, the edge gets more jagged, and an increase in total count of pixels classified as grey can be seen. Initial classification was done at this point, confirming the initial assumptions about increasing number of grey pixels, but for better overall accuracy, an additional classification model was used.

The model prepared for final classification used light gradient boosting machine (Ke et al. 2017, also called LGBM or light GBM), and took as input the initial array containing counts of image grayscale values (list of 256 elements, containing number of pixels per each grayscale value). The prepared model used Bayesian optimization of hyperparameters and multi log loss metric. As for the Light GBM algorithm, in general it is a tree-based solution, with different approach to tree expansion. While most methods will grow trees horizontally, in this case they are grown vertically (or leaf wise), with maximal delta loss, reducing more loss than in case of level-wise algorithms. The main advantage of these methods is that it can handle large data sets with less memory, it focuses on result accuracy, and has an efficiency parameter used as one of quality indicators. While Light GBM can be prone to overfitting problem, the extensive dataset used in the experiments was enough to prevent that from occurring.

CNN designed from scratch

Since it is always advisable to start from the simplest solution, without unnecessary complications, the first prepared algorithm implemented CNN network, trained from scratch on prepared data-set, without using transfer learning methodology. Since the amount of collected data was sufficient for the training process, using convolutions neural networks for presented task of drill wear classification was the obvious choice. CNN is one of the best classification methods when it comes to images, and with appropriate amount of data (as it was the case in the current approach), training of such network can yield good results without any additional operations.

Overall outline of prepared CNN network is presented in Table 4. It is composed firstly of layers that belong to basic network structure which automatically extract image features. Second part of the presented model (starting with flatten layer) is a DNN (Dense Neural Network), used for actual learning process and classification in the form of last, 3-neuron dense layer with softmax activation function. The presented model uses dropout layers to address and prevent the over-fitting problem

Table 4 Architecture of CNN model designed from scratch

Layer (type)	Output shape	Param #
conv2d (Conv2D)	(None, 64, 64, 32)	896
conv2d ₁ (Conv2D)	(None, 64, 64, 32)	9248
max _{pooling2d} (MaxPooling2D)	(None, 32, 32, 32)	0
batch _{normalization} (BatchNo	(None, 32, 32, 32)	128
dropout (Dropout)	(None, 32, 32, 32)	0
conv2d ₂ (Conv2D)	(None, 32, 32, 128)	36992
conv2d ₃ (Conv2D)	(None, 32, 32, 128)	147584
max _{pooling2d} ₁ (MaxPooling2	(None, 16, 16, 128)	0
batch _{normalization} ₁ (Batch	(None, 16, 16, 128)	512
dropout ₁ (Dropout)	(None, 16, 16, 128)	0
conv2d ₄ (Conv2D)	(None, 16, 16, 86)	99158
conv2d ₅ (Conv2D)	(None, 16, 16, 86)	66650
max _{pooling2d} ₂ (MaxPooling2	(None, 8, 8, 86)	0
batch _{normalization} ₂ (Batch	(None, 8, 8, 86)	344
dropout ₂ (Dropout)	(None, 8, 8, 86)	0
flatten (Flatten)	(None, 5504)	0
dense (Dense)	(None, 512)	2818560
dropout ₃ (Dropout)	(None, 512)	0
dense ₁ (Dense)	(None, 256)	131328
dropout ₄ (Dropout)	(None, 256)	0
dense ₂ (Dense)	(None, 64)	16448
dropout ₅ (Dropout)	(None, 64)	0
dense ₃ (Dense)	(None, 3)	195

(by randomly removing neurons) and max pooling layers to decrease overall model complexity. Moreover, during the learning process, the early-stop method was used, halting the learning process when no significant improvement was made (this parameter is monitored during each validation accuracy epoch). During each epoch, the best model is saved (for each improvement in validation accuracy), and after the early-stop method, the best model is loaded and further used.

Bagging of 5xCNN designed from scratch

This model was used to aggregate 5, designed from scratch, CNN sub-models. In this approach, the number of neurons in dense layers is drawn in the accepted range. This methodology is called bagging and is a team learning technique, where final classification is obtained by majority voting of all used models (5 in this case). Each of the individual results obtained by sub-models is treated equivalently. What it means is that final classification is based on simple m -valued dominant from all individual sub-model results.

The ranges used for each dense layer are drawn in the following ranges: Dense1: (500, 600), Dense2: (200, 300), Dense3: (100, 200), Dense4: (50, 100).

VGG19 pretrained network

In this approach, transfer learning technique has been applied. The first of the pre-trained networks used in the current approach was VGG19. VGG19 is a trained Convolutional Neural Network, from Visual Geometry Group, Department of Engineering Science, University of Oxford. The number 19 derives from the number of layers which are used for the training process (containing weights): 16 convolutional layers and 3 dense layers. Pretrained VGG19 network has been trained based on more than a million images from the ImageNet (2016) database. VGG19 is able to classify 1000 various types of objects such as keyboard, mouse, pencil, many animals, etc.

The initial model was adjusted to better fit the presented problem and recognize 3 required classes as in previous approaches. The weights have been downloaded from <http://www.image-net.org> and frozen. Only DNN part of the whole model was trainable while CNN network has been frozen. Complete list of layers for VGG19 is presented at Table 5. In this case, for the pretrained VGG19 model deep neural network

Table 5 List of CNN layers in the model based on VGG19

Model: “vgg19”		
Layer (type)	Output shape	Param #
input ₁ (InputLayer)	[(None, 64, 64, 3)]	0
block1 _{conv1} (Conv2D)	(None, 64, 64, 64)	1792
block1 _{conv2} (Conv2D)	(None, 64, 64, 64)	36928
block1 _{pool} (MaxPooling2D)	(None, 32, 32, 64)	0
block2 _{conv1} (Conv2D)	(None, 32, 32, 128)	73856
block2 _{conv2} (Conv2D)	(None, 32, 32, 128)	147584
block2 _{pool} (MaxPooling2D)	(None, 16, 16, 128)	0
block3 _{conv1} (Conv2D)	(None, 16, 16, 256)	295168
block3 _{conv2} (Conv2D)	(None, 16, 16, 256)	590080
block3 _{conv3} (Conv2D)	(None, 16, 16, 256)	590080
block3 _{conv4} (Conv2D)	(None, 16, 16, 256)	590080
block3 _{pool} (MaxPooling2D)	(None, 8, 8, 256)	0
block4 _{conv1} (Conv2D)	(None, 8, 8, 512)	1180160
block4 _{conv2} (Conv2D)	(None, 8, 8, 512)	2359808
block4 _{conv3} (Conv2D)	(None, 8, 8, 512)	2359808
block4 _{conv4} (Conv2D)	(None, 8, 8, 512)	2359808
block4 _{pool} (MaxPooling2D)	(None, 4, 4, 512)	0
block5 _{conv1} (Conv2D)	(None, 4, 4, 512)	2359808
block5 _{conv2} (Conv2D)	(None, 4, 4, 512)	2359808
block5 _{conv3} (Conv2D)	(None, 4, 4, 512)	2359808
block5 _{conv4} (Conv2D)	(None, 4, 4, 512)	2359808
block5 _{pool} (MaxPooling2D)	(None, 2, 2, 512)	0
Total params: 20,024,384		
Trainable params: 0		
Non trainable params: 20,024,384		

Table 6 List of DNN layers in the model based on VGG19

Layer (type)	Output shape	Param #
vgg19 (Model)	(None, 2, 2, 512)	20024384
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1024)	2098176
dropout (Dropout)	(None, 1024)	0
dense ₁ (Dense)	(None, 1024)	1049600
dropout ₁ (Dropout)	(None, 1024)	0
dense ₂ (Dense)	(None, 512)	524800
dropout ₂ (Dropout)	(None, 512)	0
dense ₃ (Dense)	(None, 3)	1539
Total params: 23,698,499		
Trainable params: 3,674,115		
Non trainable params: 20,024,384		

(DNN) was added at the end of it. Table 6 shows the whole model which has been taken into consideration in the first algorithm.

Bagging of 5xVGG16 and 10xVGG16 pretrained network

The next solution checked during the experiments used pretrained VGG16 model (which is also a pretrained CNN network), additionally improving it with bagging approach. The initial network was trained to recognize 1000 classes, which was adjusted to three classes used in the current problem. VGG16 model was chosen for the bagging approach, since it is less complex than VGG19, and with this algorithm the over-fitting problem is less likely to occur. Two versions of this approach were prepared, the first one using 5 and the second one using 10 randomly initialized classifiers. Outline of single VGG16 classifier structure is presented in Table 7, while the DNN classifier is shown in Table 8. Similarly, as with 5xCNN designed from scratch, four random neuron numbers, from the intervals: (300, 400), (200, 300), (100, 200) and (50, 100), were chosen for all estimators. Moreover, during the training process, only 90% of the training data set was randomly chosen for each estimator. After the training process, each of the estimators would show the probability of the considered example. These probabilities were later summed up, and maximal value was chosen as final classification.

Table 7 List of layers in CNN model based on VGG16

Model: “vgg16”		
Layer (type)	Output shape	Param #
input ₁ (InputLayer)	[(None, 64, 64, 3)]	0
block1 _{conv1} (Conv2D)	(None, 64, 64, 64)	1792
block1 _{conv2} (Conv2D)	(None, 64, 64, 64)	36928
block1 _{pool} (MaxPooling2D)	(None, 32, 32, 64)	0
block2 _{conv1} (Conv2D)	(None, 32, 32, 128)	73856
block2 _{conv2} (Conv2D)	(None, 32, 32, 128)	147584
block2 _{pool} (MaxPooling2D)	(None, 16, 16, 128)	0
block3 _{conv1} (Conv2D)	(None, 16, 16, 256)	295168
block3 _{conv2} (Conv2D)	(None, 16, 16, 256)	590080
block3 _{conv3} (Conv2D)	(None, 16, 16, 256)	590080
block3 _{pool} (MaxPooling2D)	(None, 8, 8, 256)	0
block4 _{conv1} (Conv2D)	(None, 8, 8, 512)	1180160
block4 _{conv2} (Conv2D)	(None, 8, 8, 512)	2359808
block4 _{conv3} (Conv2D)	(None, 8, 8, 512)	2359808
block4 _{pool} (MaxPooling2D)	(None, 4, 4, 512)	0
block5 _{conv1} (Conv2D)	(None, 4, 4, 512)	2359808
block5 _{conv2} (Conv2D)	(None, 4, 4, 512)	2359808
block5 _{conv3} (Conv2D)	(None, 4, 4, 512)	2359808
block5 _{pool} (MaxPooling2D)	(None, 2, 2, 512)	0
Total params: 14,714,688		
Trainable params: 0		
Non trainable params: 14,714,688		

Table 8 List of layers in DNN model based on VGG16

Model: “sequential”		
Layer (type)	Output shape	Param #
vgg16 (Model)	(None, 2, 2, 512)	14714688
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1024)	2098176
dropout (Dropout)	(None, 1024)	0
dense ₁ (Dense)	(None, 1024)	1049600
dropout ₁ (Dropout)	(None, 1024)	0
dense ₂ (Dense)	(None, 512)	524800
dropout ₂ (Dropout)	(None, 512)	0
dense ₃ (Dense)	(None, 3)	1539
Total params: 18,388,803		
Trainable params: 3,674,115		
Non trainable params: 14,714,688		

Ensemble of 3 best models

After initial tests and obtaining accuracy results from previous algorithms, three best performing solutions were chosen for classifier ensemble. In that case, CNN model trained from scratch, VGG19 and 5xVGG16 were used, since those performed best in terms of overall accuracy, assuming that every chosen model should operate in a different manner, and be efficient enough for the manufacturer requirements. It was assumed that combining different network results for the same classification task will increase overall accuracy of prepared solution.

Experiment results and discussion

While preparing the current approach for drill wear classification, experiments were divided into two main parts: the first one evaluating different algorithm adjustments to chosen problem with additional manufacturer requirements, and the second one, where a custom solution was prepared using initial results obtained in the first step.

Initial algorithm set evaluation

During the first stage of the performed experiments, overall algorithm adjustment to drill wear classification problem was tested. Additionally, it was checked how well

Table 9 Classification report for CNN model designed from scratch for window = 20

	Precision	Recall	F1-score	Support
Green	0.8667	0.9032	0.8846	3780
Yellow	0.7562	0.5778	0.6551	3780
Red	0.7611	0.9087	0.8284	3780
Accuracy			0.7966	11,340

Table 10 Classification report for CNN model designed from scratch with bagging for window = 20

	Precision	Recall	F1-score	Support
Green	0.9152	0.8796	0.8971	3,780
Yellow	0.6950	0.7029	0.6989	3,780
Red	0.7863	0.8079	0.7970	3,780
Accuracy			0.7968	11,340

Table 11 Classification report for pretrained model VGG19 for window = 20

	Precision	Recall	F1-score	Support
Green	0.9065	0.8955	0.9010	3780
Yellow	0.6710	0.6960	0.6833	3780
Red	0.7761	0.7566	0.7662	3780
Accuracy			0.7827	11,340

Table 12 Classification report for bagging of 5xVGG16 pretrained network for window = 20

	Precision	Recall	F1-score	Support
Green	0.9426	0.8468	0.8921	3780
Yellow	0.6695	0.7347	0.7006	3780
Red	0.7864	0.7897	0.7880	3780
Accuracy			0.7904	11,340

Table 13 Classification report for bagging of 10xVGG16 pretrained network for window = 20

	Precision	Recall	F1-score	Support
Green	0.9249	0.8243	0.8717	3780
Yellow	0.6526	0.7511	0.6984	3780
Red	0.8056	0.7717	0.7883	3780
Accuracy			0.7824	11,340

Table 14 Classification report for ensemble of 3 best models for window = 20

	Precision	Recall	F1-score	Support
Green	0.9210	0.8942	0.9074	3780
Yellow	0.7153	0.6921	0.7035	3780
Red	0.7805	0.8286	0.8038	3780
Accuracy			0.8049	11,340

Fig. 4 Confusion matrixes for CNN network designed from scratch for no window and window size = 20

each of the chosen solutions met the additional manufacturer requirements for no green-red misclassifications. Tables 9, 10, 11, 12, 13, 14 present overall accuracy results for each of the evaluated algorithms with largest window used. Figures 4, 5, 6, 7, 8, 9 show confusion matrixes for two border cases, when it comes to the used image range: case with no window, and with largest window used (equal to 20 consecutive images).

Fig. 5 Confusion matrixes for 5 random classifiers using CNN network designed from scratch for no window and window size = 20

Fig. 6 Confusion matrix for VGG19 algorithm for no window and window size = 20

Fig. 7 Confusion matrixes for 5 random VGG16 estimators voting algorithm for no window and window size = 20

Fig. 8 Confusion matrixes for 10 random VGG16 estimators voting algorithm for no window and window size = 20

Fig. 9 Confusion matrixes for classifier ensemble using VGG19, ensemble of 5 VGG16 networks and CNN network created from scratch for no window and window size = 20

Out of the initial algorithms, the two single classifier approaches performed relatively well, with high overall accuracy (with 79.66% overall accuracy for custom CNN and 78.27% for VGG19, with window size equal to 20), and small number of critical errors (3 for custom CNN and 7 for VGG19, for the same window size). At the same stage, two bagging approaches were discarded, since they did not produce satisfactory results. The first one, using 5 random initializations of CNN network trained from scratch, although it obtained slightly higher overall accuracy (79.04%), also made significantly more critical errors (16 in total). The second one, using 10 random initializations of VGG16 classifier, not only had lower overall accuracy, compared to bagging approach using only 5 classifiers of that kind (78.24% compared to 79.04%), but also made more critical errors (25, compared to 15 in total). Since both those solutions were also more complex and time consuming, they were discarded.

The final solution tested during that stage used an ensemble of three models that performed best up to this point: CNN trained from scratch, VGG19 and 5xVGG16.

Table 15 Classification results for chosen algorithms, for classification with no window and with different window sizes (5, 10, 15 and 20) for consecutive samples

#	Algorithm	No window (%)	W = 5 (%)	W = 10 (%)	W = 15 (%)	W = 20 (%)
1	Baseline	66.95	71.98	73.89	75.87	76.57
2	CNN-designed	69.78	75.47	78.30	79.42	79.66
3	5xCNN-designed	67.35	73.41	76.74	78.62	79.68
4	VGG19	66.77	72.81	75.90	77.26	78.27
5	5xVGG16	67.13	73.90	77.04	78.15	79.04
6	10xVGG16	66.98	73.18	75.92	77.23	78.24
7	Ensemble (2,4,5)	69.26	75.61	78.63	79.88	80.49

Table 16 Number of critical errors between green-red and red-green for no window and window = 20

#	Algorithm	No window	W = 20
1	Baseline	174	29
2	CNN-designed	156	3
3	5xCNN-designed	155	16
4	VGG19	149	7
5	5xVGG16	176	15
6	10xVGG16	195	25
7	Ensemble (2,4,5)	133	7

Since in previous experiments such combined approach tended to increase overall accuracy of the presented solution, it was assumed that it would behave similarly in this case. Table 15 summarizes overall accuracy rates for all tested algorithms, with different sizes of image sequences used (starting with no window, and with maximal window size equal to 20). Table 16 outlines critical error rates for tested solutions, for two border cases: no window, and window size set to 20. When it comes to overall accuracy, the ensemble solution performed best for all cases, except the no window case, where single, custom CNN achieved best results. It was also able to achieve highest overall accuracy rate up to this point, for window size 20, of 80.49%. When it comes to critical error rate, it achieved best result when it comes to no window case (133 critical errors in total, when second best solution, VGG19 had 149 errors), and second best result for window = 20 (7 critical errors, with best solution being CNN designed from scratch with only 3 such mistakes).

While the above results could be accepted as satisfactory, it was ultimately decided, that the ensemble solution achieved too similar results to single CNN model trained from scratch when it comes to overall accuracy (0.83% difference). Additionally, since it made more critical errors, the solution was discarded, and additional experiments were performed, to improve the entire solution, resulting in the final, improved algorithm for drill wear classification.

Improved approach for drill wear classification

Continuing the assumption based on previous research made in chosen subject (Kurek et al. 2019a, b), when connecting different classifiers into an ensemble and using the voting method to obtain final classification improved overall solution, a similar approach was used for the final solution presented in this paper. While initial experiments obtained some acceptable results, they were still not satisfactory, and additional work to improve overall results was required.

Although the final solution presented in this paper also uses classifier ensemble, a different approach was used when deciding which algorithms will be a part of it. This time, instead of overall classification accuracy, solutions were chosen based on their ability to distinguish between key red and green classes. Based on that aspect first, and overall algorithm accuracy second, including additional parameters connected to overall ensemble properties, the three following algorithms were chosen:

1. **5 random VGG16 estimators**—since this approach performed rather good in case of overall accuracy (79.04% for window size 20) and relatively well in case of red and green class distinction (176 critical errors for the case with no window, and 7 for window size 20).
2. **CNN model prepared from scratch, without using transfer learning** - since this model had satisfactory results for both considered accuracy types (156 critical errors in case of no window, and only 3 for window size 20, with overall accuracy of 79.66% for largest window).
3. **ensemble of 5 random CNN models, based on the one prepared from scratch**—even though it did not improve single CNN model significantly and performed worse in case of critical error rate for largest window (155 critical errors for no window, and 16 for window size 20, with overall accuracy of 79.68% for largest window) it was decided, that it added important diversity to the classifier ensemble.

Above algorithms were chosen for a few reasons. Firstly, accuracy of each algorithm was confronted with how well each of them can distinguish between red and green class. Only those algorithms were chosen that had those values in acceptable threshold. Secondly, for those algorithms it was verified that increasing window size resulted in an increased accuracy (since window contains sequences of hole images drilled by the same drill, it was safe to assume, that any algorithm appropriately adjusted to chosen problem should be able to considerably increase its accuracy with larger window used for evaluation). Third and final element included while choosing algorithms was picked to increase the diversity of methods in the final set. In case of algorithms ensemble it is best that the chosen elements are as diverse as possible, to ensure higher accuracy of final solution. All the algorithms used in the final set are significantly different, either by specifics of CNN network used for classification (VGG16 and CNN network that was designed and trained from scratch), or by type of final method used for classification (CNN classification or voting of classifiers).

After designing the final solution, the same set of experiments as with previous algorithm was used to evaluate its performance. The algorithm was tested using the same set of window sizes (starting with no window, and finishing with window size = 20). Final classification was obtained by summing up probabilities of considered image belonging to each of three classes and choosing the one with highest score. For testing purposes, the samples were shuffled, to ensure that classification will be based on random examples instead of sequenced images. Due to initial data augmentation, each class was equally represented in sets representing all drills used during the experiments (5 total), ensuring a balanced set, both for learning and testing purposes.

During tests, the prepared algorithm behaved according to initial predictions. The first run, with no set window for checking sequence of images (while training) was the worst one, with overall accuracy of 70% and considerably high number of misclassifications between red and green class. In the following steps, the obtained overall accuracy was steadily increasing, while number of misclassification started to decrease. Accuracy of 75% was obtained for window size = 5 and 78% for window size = 10. After that, overall accuracy of classification stabilizes around a value of 80%. At this point, acceptable accuracy was reached (similar to the ones obtained in existing approaches with much more complicated setups containing diverse sensors and elaborate methodologies used for the initial data collection and preparation) with close to none misclassifications between two opposite classes (total of 7 critical errors).

As was mentioned before, in case of such applications decreasing number of misclassifications between distant classes is in general more important than overall algorithm classification accuracy. In this case, the prepared method is currently evaluated in the actual working environment, and online application prepared during experiments is being tested by actual manufacturer. Initial feedback was very positive, and it was forwarded that the prepared solution is fully applicable to the problem of drill wear recognition. Furthermore, its simplicity in terms of used input data, and equipment required to obtain it, is an additional, huge advantage over most of the existing solutions.

As for possible areas of future work, the next step would concern increasing overall algorithm accuracy. While the obtained results are in an acceptable range, it would still be preferable, to improve it. Furthermore, possible alternatives to window parameter (denoted as sequence of images used during training) need to be evaluated. While using sets of ordered images during the learning process is very effective, obtaining such data is not always easy, and sometimes might prove to be impossible. Therefore it would be wise to find some alternative for that process with at least similar accuracy rate.

Conclusion

In this paper an improved algorithm for drill wear classification using images of drilled holes as an input data was presented. Balanced set of images was prepared for the experiments, to make sure that it will not favour any of the classes. Since

they were done in a sequence (meaning that images of holes drilled by the same tool over time were put in order in which they were made), that data were used to increase algorithm accuracy. Series of tests were then performed, evaluating different algorithms based on CNN networks (both using transfer learning methodologies, and preparing such network from scratch and training it on obtained examples). During experiments overall accuracy of each algorithm for different sizes of image sequence window (used during training process) was checked. At this point, the rate of misclassifications between red and green classes, was also evaluated. This last parameter was added to the set since it was very important for the manufacturer—mistakes between those two groups can result in loss due to poor product quality, or by discarding tool that is still in a good shape. The final algorithm that was prepared consisted of an ensemble of three methods: ensemble of 5 random VGG16 classifiers, CNN model trained from scratch, and ensemble of 5 of those CNN models, randomly initialized. 80% accuracy was reached for window size = 20, with close to none misclassifications (7 critical errors total).

The prepared solution has an overall accuracy that fits into the initial acceptable range, while maintaining good misclassification rate for the important elements. Although similar to single CNN model, it is superior due to the diversity of used classifiers, and as a result, less susceptible to overfitting problem. It is also advanced enough to be used with real data, and online version of it is available to the manufacturer. Initial feedback was very positive, both because of accuracy of the presented solution, as well as its simplicity—using images instead of complicated data set (which requires equally elaborate sensors, and long period of time during which those sensors are configured, and later evaluated, to properly collect required information). All of the above allowed for faster data collection and easier adaptation of the presented solution to actual work environment, without high initial costs.

Acknowledgements On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Bengio Y (2009) Learning deep architectures for AI. *Found Trends TM Mach Learn* 2(1):1–127. <https://doi.org/10.1561/22000000006>
- Deng L, Yu D (2014) Deep learning: methods and applications. *Found Trends TM Signal Process* 7(3–4):197–387. <https://doi.org/10.1561/20000000039>
- Goodfellow I, Bengio Y, Courville A (2016) Deep learning. The MIT Press, Cambridge, MA

- Hu J, Song W, Zhang W, Zhao Y, Yilmaz A (2019) Deep learning for use in lumber classification tasks. *Wood Sci Technol* 53(2):505–517. <https://doi.org/10.1007/s00226-019-01086-z>
- Ibrahim I, Khairuddin ASM, Talip MSA, Arof H, Yusof R (2017) Tree species recognition system based on macroscopic image analysis. *Wood Sci Technol* 51(2):431–444
- ImageNet (2016) Stanford vision lab. <http://www.image-net.org>
- Jęgorowa A, Górski J, Kurek J, Kruk M (2019) Initial study on the use of support vector machine (SVM) in tool condition monitoring in chipboard drilling. *Eur J Wood Prod* 77(5):957–959
- Jęgorowa A, Górski J, Kurek J, Kruk M (2020) Use of nearest neighbors (K-NN) algorithm in tool condition identification in the case of drilling in melamine faced particleboard. *Maderas Ciencia y tecnología* 22(2):189–196
- Jemielniak K, Urbański T, Kossakowska J, Bombiński S (2012) Tool condition monitoring based on numerous signal features. *Int J Adv Manuf Technol* 59(1–4):73–81. <https://doi.org/10.1007/s00170-011-3504-2>
- Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY (2017) Lightgbm: a highly efficient gradient boosting decision tree. In: *Advances in neural information processing systems*, pp 3146–3154
- Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, pp 1097–1105
- Kuo R (2000) Multi-sensor integration for on-line tool wear estimation through artificial neural networks and fuzzy neural network. *Eng Appl Artif Intell* 13(3):249–261. [https://doi.org/10.1016/S0952-1976\(00\)00008-7](https://doi.org/10.1016/S0952-1976(00)00008-7)
- Kurek J, Kruk M, Osowski S, Hoser P, Wiecezorek G, Jęgorowa A, Górski J, Wilkowski J, Śmiateńska K, Kossakowska J (2016) Developing automatic recognition system of drill wear in standard laminated chipboard drilling process. *Bull Pol Acad Sci Tech Sci*. <https://doi.org/10.1515/bpasts-2016-0071>
- Kurek J, Świdorski B, Jęgorowa A, Kruk M, Osowski S (2017a) Deep learning in assessment of drill condition on the basis of images of drilled holes. In: *Eighth international conference on graphic and image processing (ICGIP 2016)*, international society for optics and photonics, vol 10225, p 102251V. <https://doi.org/10.1117/12.2266254>
- Kurek J, Wiecezorek G, Kruk M, Świdorski B, Jęgorowa A, Osowski S (2017b) Transfer learning in recognition of drill wear using convolutional neural network. In: *2017 18th international conference on computational problems of electrical engineering (CPEE)*, IEEE, pp 1–4. <https://doi.org/10.1109/CPEE.2017.8093087>
- Kurek J, Antoniuk I, Górski J, Jęgorowa A, Świdorski B, Kruk M, Wiecezorek G, Pach J, Orłowski A, Aleksiejuk-Gawron J (2019a) Classifiers ensemble of transfer learning for improved drill wear classification using convolutional neural network. *Mach Graph Vis* 28:13–23
- Kurek J, Antoniuk I, Górski J, Jęgorowa A, Świdorski B, Kruk M, Wiecezorek G, Pach J, Orłowski A, Aleksiejuk-Gawron J (2019b) Data augmentation techniques for transfer learning improvement in drill wear classification using convolutional neural network. *Mach Graph Vis* 28:3–12
- Panda S, Singh A, Chakraborty D, Pal S (2006) Drill wear monitoring using back propagation neural network. *J Mater Process Technol* 172(2):283–290. <https://doi.org/10.1016/j.jmatprotec.2005.10.021>
- Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) ImageNet large scale visual recognition challenge. *Int J Comput Vis* 115(3):211–252. <https://doi.org/10.1007/s11263-015-0816-y>
- Schmidhuber J (2015) Deep learning in neural networks: an overview. *Neural Netw* 61:85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>
- Shelhamer E (2017) BVLC AlexNet Model. https://github.com/BVLC/caffe/tree/master/models/bvlc_alexnet