Time-efficient Approach to Drill Condition Monitoring Based on Images of Holes Drilled in Melamine Faced Chipboard

Albina Jegorowa,[a,]* Izabella Antoniuk,[b] Jarosław Kurek,[b] Michał Bukowski,[c]
Wioleta Dołowa,[d] and Paweł Czarniak [a]

This paper presents a time-efficient approach to the drill wear classification problem that achieves a similar accuracy rate compared to more complex and time-consuming solutions. A total of three classes representing drill state are recognized: red for poor state, yellow for elements requiring additional evaluation, and green for good state. Images of holes drilled in melamine faced chipboard were used as input data, focusing on evaluating differences in image color values to determine the overall drill state. It is especially important that there are as few mistakes as possible between the red and green class, as these generate the highest loss for the manufacturer. In green samples presented in gray-scale, most pixels were either black (representing the hole) or white (representing the chipboard), with very few values in between. The current method was based on the assumption that the number of pixels with intermediate values, instead of extreme ones, would be significantly higher for the red class. The presented initial approach was easy to implement, generated results quickly, and achieved a similar accuracy compared to more complex solutions based on convolutional neural networks.

Keywords: Chipboard machining; Melamine faced chipboard; Tool condition monitoring; Drill wear classification

Contact information: a: Institute of Wood Sciences and Furniture, Warsaw University of Life Sciences WULS-SGGW, Poland; b: Institute of Information Technology, Warsaw University of Life Sciences WULS-SGGW, Poland; c: no affiliation, Poland; d: no affiliation, Poland;
** Corresponding author: albina_jegorowa@sggw.edu.pl*

INTRODUCTION

Tool condition monitoring is a research area dealing with evaluating and assessing how long different machine elements can be used before they wear out. Algorithms can incorporate different input signals, and various methods can be used for data collection depending on tool-specific properties. Many of the algorithms used for tool condition monitoring consider drill condition monitoring specifically. For the general evaluation of a drill state, it is very important for the manufacturer to determine the exact moment when a drill is starting to dull out, resulting in a product that is not satisfactory. If this point in time is not detected quickly, then such poor quality elements will generate losses for the company. While the drill state can be evaluated manually, it is a tiresome and time-consuming process, hence the need for automatization.

Existing solutions for drill condition monitoring vary greatly in data collection and preprocessing methodologies, and they usually incorporate dedicated sensors and devices. Signals that are measured can be related to feed force, noise and vibrations, cutting torque, acoustic emission, or other parameters (Kurek *et al*. 2016). The main disadvantage of this approach is that a complicated and quite expensive arrangement of diverse sensors needs to be installed to register the relevant signals. Furthermore, many preprocessing stages are required to obtain actual, usable data. During preprocessing, it needs to be ensured that chosen sensors are appropriate for the current environment; in addition, registered signals need to be checked to ensure that they are the correct signals for generation of usable diagnostic features. Finally, the best features need to be selected and used for building the final classification model. Error at any of these stages can result in a classifier that is not usable. Many different features are generated on the basis of the registered signals. However, the challenge is that the classification accuracy stays below the 90% threshold, while the entire process is lengthy and difficult to implement in an actual work environment (Kuo and Cohen 1999; Panda *et al.* 2006; Jemielniak *et al.* 2012). The initial setup is quite expensive and does not guarantee that satisfactory results will be achieved in a reasonable time (or at all if, for example, the wrong sensors are chosen), and often not compensating in any way for the amount of time spent during those initial stages.

In the presented approach, the main focus was on accelerating and simplifying this process in order to make it more applicable for the furnishing companies, without adding high initial costs. Three classes were defined for drill condition monitoring: green, red, and yellow. The green class described tools that were in good condition and could be used further, the red class denoted tools in poor condition, that should be immediately replaced, and the yellow class was for tools that were suspected of being too worn out for further use and, which required manual evaluation by a human expert. From the manufacturer's perspective, mistakes between green and red classes are far more undesirable than the others because they result in the highest possibility of financial loss; this distinction is more important than overall accuracy. The second important element concerns time required to prepare usable solution, which should be minimized. It is especially undesirable if a long time is required to set up and calibrate used sensors, since such setups cannot always remain unfolded in the actual work environment. The usability of a solution is further diminished if both a long time for initial setup is needed and lengthy computations are done before first results can be obtained (*i.e.* a prolonged training process is needed).

The current solution is based on some existing ones (Bengio 2009; Deng and Yu 2014; Schmidhuber 2015) as well as on the authors' previous research in this area (Kurek *et al.* 2017a,b, 2019a,b). The first major improvement was removing any specialized equipment for initial data collection. A camera, which was used to take pictures of drilled holes, was the only external element used for data collection. This approach simplified the entire solution and allowed much easier assembly in the working environment. Furthermore, this does not require a large financial investment from the furniture company. During initial approaches, different methods were tested, in which the algorithms were based on convolutional neural networks (CNN); this is a good solution that does not require specialized diagnostic features (Goodfellow *et al.* 2016). Limited training data was used, based on Kurek *et al.* (2017a), and an artificially extended initial set of samples, based on Kurek *et al.* (2017b) was used while recognizing two classes. The extended approach combined data augmentation methodology with transfer learning (Kurek *et al.* 2019a), achieving higher accuracy than the results obtained by using complicated setups with diverse sensors (over 93%). In the final solution (Kurek *et al.* 2019b), a classifier ensemble was prepared using different pre-trained networks (Krizhevsky *et al.* 2012; Russakovsky *et al.* 2015; AlexNet Model online); this time over 95% accuracy was achieved.

While the above approaches showed high accuracy, two problems were encountered. Firstly, the methods did not take into consideration the additional manufacturer requirement regarding green-red misclassification rate, which should be minimized (high overall accuracy rate is inadequate if the number of mistakes between those two classes is significant). Secondly, although the training process was simplified and faster than solutions based on more complicated sensor setups, it still was not fast enough to meet manufacturer requirements. Therefore, in the current approach the main focus was placed on improving these aspects, while maintaining high overall accuracy. To achieve this, image color distribution was evaluated, with the assumption that samples classified as red will have a much higher number of pixels having values between black and white (in an image represented in gray-scale) than those not classified as red. In order to select the best possible setup, recursive feature elimination and cross-validated selection was used, along with applied Bayesian optimization of hyperparameters.

The rest of this work is organized as follows: The next section contains a description of the overall data preparation process and analysis (in regards to additional manufacturer requirements), as well as the prepared algorithm (which is capable of precise and efficient classification). The section after this then describes the results obtained during performed experiments. Conclusions are presented in the final section.

EXPERIMENTAL

Materials and methods

Holes were drilled using a standard CNC (Busellato Jet 100, Thiene, Italy) vertical machining center. The drillings were made using tungsten carbide through-hole drill bits Faba WP-01 (Faba SA, Baboszewo, Poland). An example drill (diameter of chosen tool equaled 12 mm) is presented in Fig. 1. Drills used in presented experiments can be used on glued wood, chipboard, and derivatives. A total of five drills were used to obtain input images, while holes drilled by each of them were separately stored in a time series fashion (representing the exact order in which the successive drillings were made).

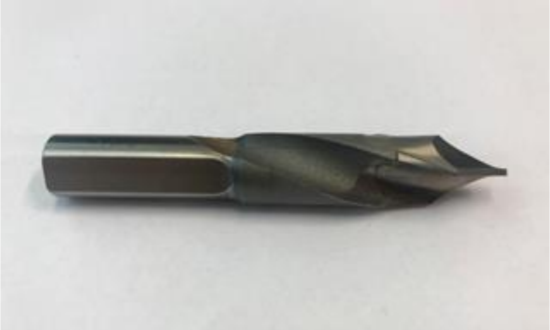

Fig. 1. General view of drill bit used in experiments (FABA WP - 01)

The tool was periodically monitored using a standard workshop microscope (TM – 505; Mitutoyo, Kawasaki, Japan) for manual evaluation of the drill state. Wear of the outer corner was observed and defined as *W* (mm) (Jegorowa *et al.* 2019, 2020). This was determined separately for each of the blades, and the arithmetic mean of the wear of the cutting edges was calculated. Three classes for drill wear were selected based on obtained values. The upper limit of the “green” class (0.2) was defined by the manufacturer of the tools used in current experiment. Remaining classes: “yellow” (0.2-0.35) and “red” (above

0.35) were arbitrarily determined based on expert observation of the machining quality to distinguish the "worn" state of the tool from the one that is "completely worn out". Those classes were also used for drill wear definition in the current, automated approach. Figure 2 shows the method used for manual evaluation of tool state.

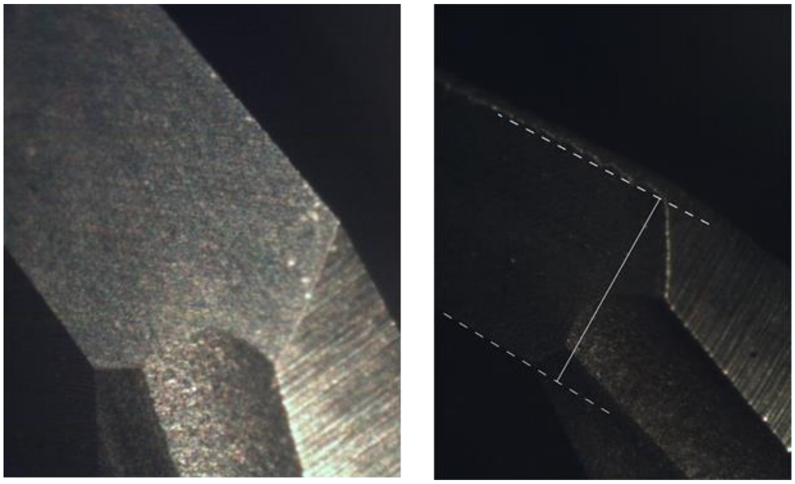

Fig. 2. Method used during manual tool state evaluation

During the drilling process the spindle speed was set at 4500 rpm, while the feed speed equaled 1,35 m/min. The above parameters were selected according to the recommendations of the tool manufacturer. The drillings were made using three-layer melamine faced chipboard (Kronopol U 511 SM; Swiss Krono Sp. z o. o., Żary, Poland), an example of which is presented in Fig. 3. The test piece size equaled 2500 x 300 x 18 mm. The processed material is characterized by a different density resulting from a multilayer structure. The density profiles of the melamine faced chipboard (Fig. 4) were measured on a GreCon DAX 5000 device (Fagus-GreCon Greten GmbH & Co. KG, Alfeld/Hannover, Germany) (Sala *et al*. 2020). Laminate thickness was measured using an Olympus BX40 microscope (Olympus Corporation**,** Shinjuku, Tokyo, Japan) and equaled 0.092 ± 0.004 mm.

Previous work incorporated images of drilled holes in different solutions (Kurek *et al.* 2016, 2017a,b, 2019a,b) using various approaches to classification with convolutional neural networks (both training them from scratch as well as using transfer learning methodologies). Data samples with images of drilled holes were collected in cooperation with the Institute of Wood Sciences and Furniture at the Warsaw University of Life Sciences.

After the drilling process was finished, the obtained test piece was cut into smaller samples and photographed using a Nikon D810 (Nikon Corporation, Shinagawa, Tokyo, Japan) single-lens reflex digital camera with 35.9 x 24.0 mm CMOS image sensor. The images were stored in png format. Since slices of laminated chipboard contained sets of holes, individual samples still needed to be extracted. An additional method was used to prepare images for the current algorithm, while avoiding their manual preparation. In this case, image processing procedures were adapted for the task of extracting consecutive holes from the original picture and storing them in separate images.

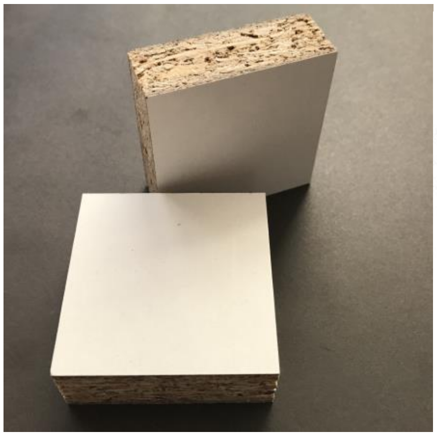

Fig. 3. General view of the material used during the experiments: the melamine faced chipboard, (Kronopol U 511 SM)

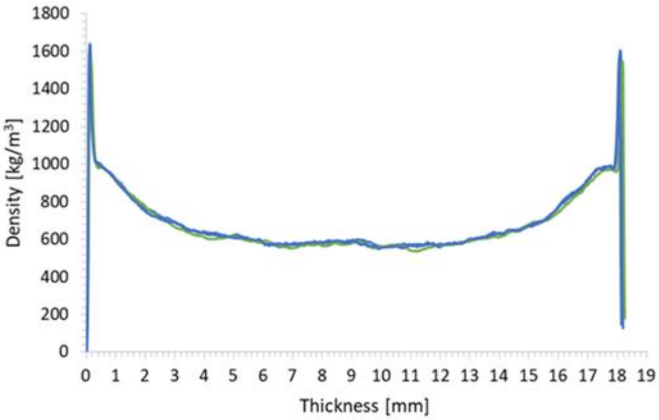

Fig. 4. Density profiles of melamine face chipboard

For CNN training, it is important that the set is balanced and has a similar number of samples representing each class, but this was not the case in the current approach. Since classification was based on image color distribution (while using gray-scale image representation, so pixel values ranged from 0 to 255), simple operations that were used to equalize number of samples for each class would not help with that aspect (*i.e.*, after rotating the image by 180°, the number of pixels for each color stays the same), hence the original set was left without additional modifications. The data set used in the current experiments contained a total of 8526 images of drilled holes, divided into 5 folders (equal to the number of drills that were used in the experiments). Out of those samples, 3780 represented the green class, 2800 the yellow class, and 1946 the red class. The above number of samples is more than sufficient for the initial approach presented in this paper for algorithm requirements and to decrease total computation time.

During initial data preparation, images were stored in the order in which the holes were drilled. This approach provides additional information about changes occurring while the drill is slowly degrading. Different windows were used to incorporate this information in the classification process taking into account the increasing numbers of consecutive images. Thus, instead of single images, sequences were used (sets containing 1, 5, 10, 15, and 20 images were used), containing pictures of holes drilled by the same tool. Figure 5 shows examples of drilled holes for used wear classes. Table 1 contains counts of samples representing each of the three classes in the 5 data folders.

Initial image classification

Sets of images were analyzed by converting them to gray-scale values during initial experiments. Pixels were then grouped into three sets: black (representing the hole), white (containing pixels that belong to the laminated part of chipboard), and gray (pixels that are in between those sets, belonging mainly to the hole edge). Pixel color value was then used for evaluation. An increase in overall gray pixel count was visible after comparing values of images representing consecutive samples for holes made by a slowly degrading tool. Since there was no clear border between each of the classes, the images could not be classified purely on that parameter alone, but the general dependency was still apparent. The current approach was based on the observation that there is a significant difference in color representation between red and green for images of drilled holes.

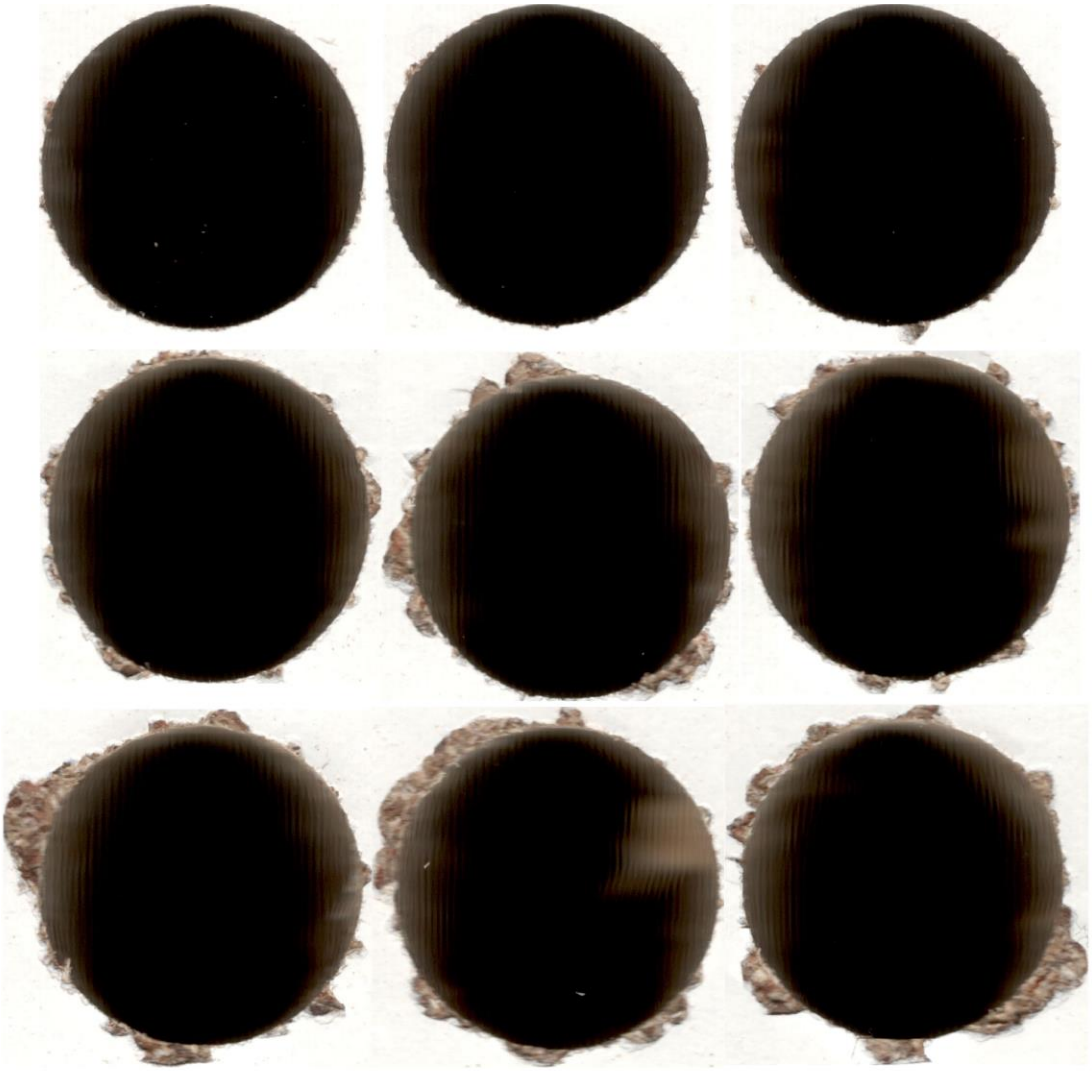

Fig. 5. Examples of holes produced by different drill wear classes: green (top), yellow (middle) and red (bottom)

Table 1. Sample Counts for Each Class in Used Data Folders

Folder Number	Green	Yellow	Red
1	840	420	406
2	840	700	280
3	700	560	420
4	840	560	280
5	560	560	560

The first class contains mostly pixels that are closer to black (the hole), or white (the chipboard) after the image is converted to gray-scale. In the red class, the edges of holes are chipped and more pixel values fall between the extremes.

The first step of the initial algorithm concerned image preparation for further use. Each initial image was converted to gray-scale and resized. Two image sizes, 256 × 256 and 500 × 500 pixels, were used to examine whether a change in dimensions influenced the algorithm accuracy or the rate of errors between red and green classes. In the second step, the number of pixels was counted for all gray-scale values in each of the input images, and then they were sorted in descending order. In the next step, pixels were also normalized to ensure that values obtained at this point fit in the 0-1 range; this was accomplished by dividing the number of pixels for each of the occurring colors by the total number of pixels in the current image.

The first classification attempts were performed using a percentage of pixels representing each color value. The next step for the algorithm involved preparing an array representing each of the images (256 elements, for each pixel value in the gray-scale image). This array was later used as input for the classification model using a light gradient boosting machine (LGBM or light GBM; Ke *et al.* 2017) with a multi log loss metric and a Bayesian optimization of hyperparameters. Light GBM uses tree-based learning, but instead of growing trees horizontally (or level-wise), it grows them vertically (or leaf-wise). This algorithm grows a leaf with maximal delta loss, and while growing the same leaf it reduces more loss than a level-wise algorithm. This method can handle large data sets while using less memory and it focuses on result accuracy, which is important for the overall accuracy of the presented solution. It also has an efficiency parameter, which is used as one of its main quality indicators. Because an extensive data set was used, there was no risk that the overfitting problem would occur (which LGBM is sensitive to); the use of this method can increase the accuracy and efficiency of the presented solution.

Image classification was done after obtaining results from this classification model based on a 256 input array. Although the initial classification attempt was good enough for the chosen solution, an additional solution was also considered, using feature selection after initial image processing.

Feature selection

An array was used as input for the procedure after obtaining the initial pixel color distribution of an image. The LGBM algorithm was still used, but feature selection was added before the actual classification was done. The algorithm chosen after careful consideration was recursive feature elimination with cross-validation (RFECV; Bahl *et al.* 2019). Finding the optimal features for any machine-learning algorithm can be difficult. There are many methods for optimal feature selection, each focusing on a different aspect. While methods such as principal component analysis (PCA) are not bad in general, they are difficult to apply to the problem presented in this work. Due to their characteristics, such methods as PCA return combinations of features, but they do not directly say which elements from the original set are important. The RFE algorithm, with additional cross-validation module (to prevent model overfitting), is a perfect solution. Instead of mapping different features to smaller subsets, it recursively removes the least important predictors until a subset of features with a given size remains. At each step, the remaining set of features is evaluated according to calculated importance, and after removing elements with low scores, everything is recalculated and repeated. As a result, the optimal subset of features can be obtained for the chosen problem.

An additional approach was used to expedite the process; this involved grouping an initial pixel count set (256 values) into a smaller number of bins (combining closest pixel values). Although using a smaller number of features would speed up the entire process, the accuracy and critical error rate of the obtained model required evaluation.

In this separate approach, the initial array obtained from the input image was first separated into bins, and then feature selection was performed before the resulting set was used by the classifier. While this method included two additional steps when compared to the original approach, it was assumed that better accuracy and lower critical error rate should not only compensate for additional computation, but should actually speed up the process; this is because the input set would contain a smaller number of features with larger bins. This applies both to the feature selection algorithm and to the classification model.

All algorithms were prepared using Python programming language (Python programming language).

RESULTS AND DISCUSSION

The first and most important parameter to consider was the number of severe (or critical) errors the classifier made (mistakes between red and green classes). From the manufacturer's point of view these errors generate the highest possibility of financial loss and, therefore, should be minimized. The second parameter concerned overall solution accuracy. While less important than the first factor, it provided additional information about model fitness for the given problem. The final element to consider was the effectiveness of the prepared solution (in terms of overall setup complexity, initial preparation time, time before first results are obtained and overall computation time). In the presented approach, initial costs are rather low and the data collection process is not complicated, however, it is still very important to the manufacturer to obtain results as quickly as possible, which can be achieved by not needing a lengthy setup stage (when the tools required for data collection are initially prepared).

Two different sizes, 256 × 256 pixels and 500 × 500 pixels, were used while scaling original images. Results for the first set of dimensions are presented in Fig. 6, for the no feature selection approach, and in Fig. 7, for the approach with feature selection. Results obtained for larger samples are presented in the same manner in Figs. 8 and 9. The lowest results for each error type as well as the highest obtained accuracy are marked in green.

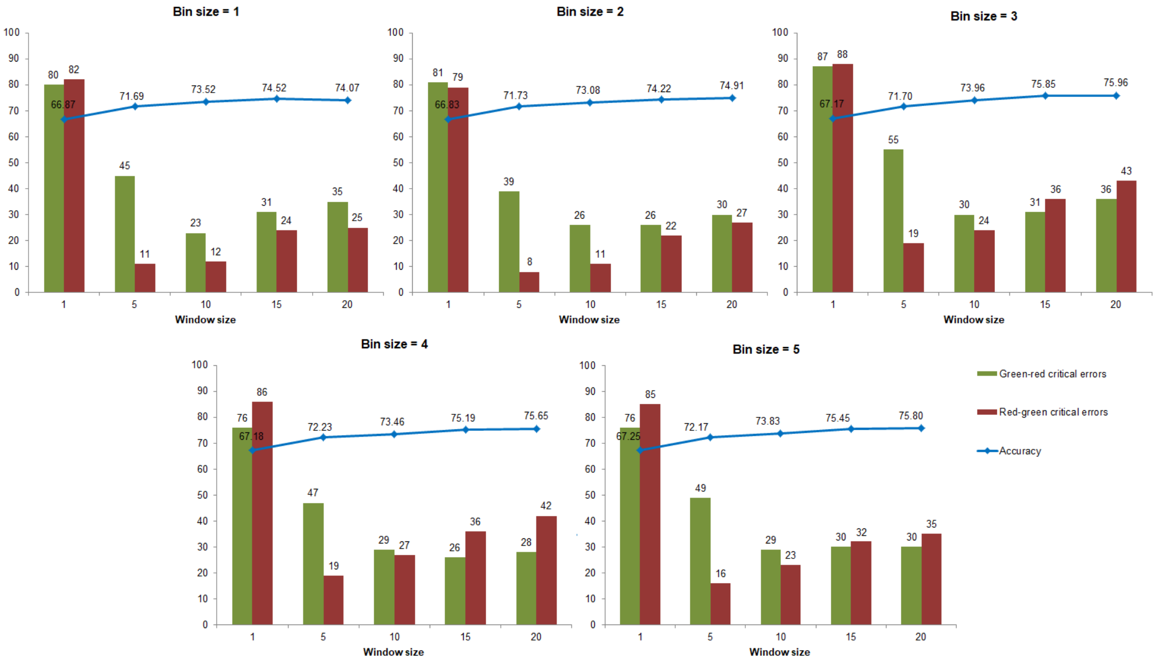

Fig. 6. Critical error and accuracy rate without feature selection, for different bin sizes using processed images with size 256 x 256 pixels

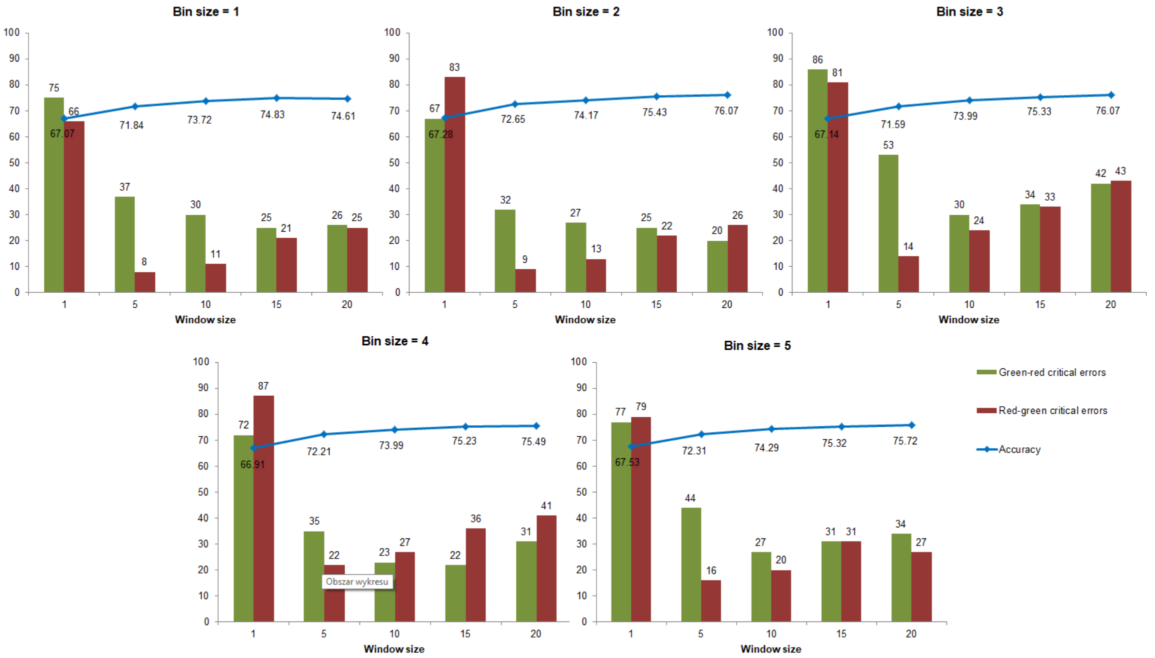

Fig. 7. Critical error and accuracy rate with feature selection, for different bin sizes using processed images with size 256 x 256 pixels

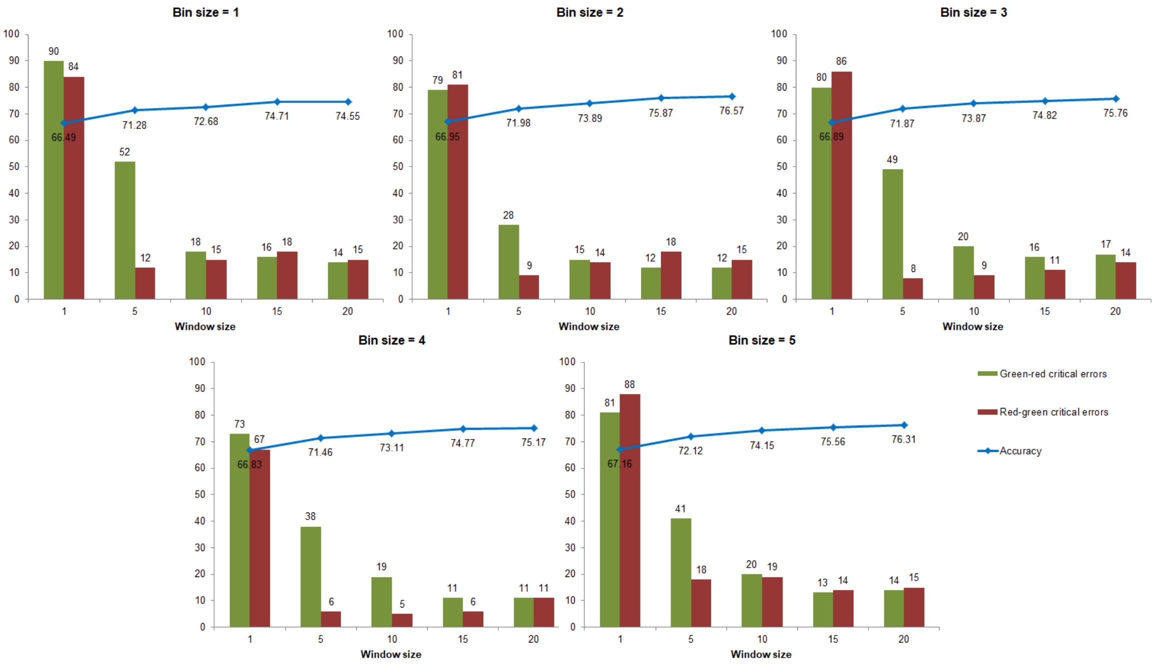

Fig. 8. Critical error and accuracy rate without feature selection, for different bin sizes using processed images with size 500 x 500 pixels

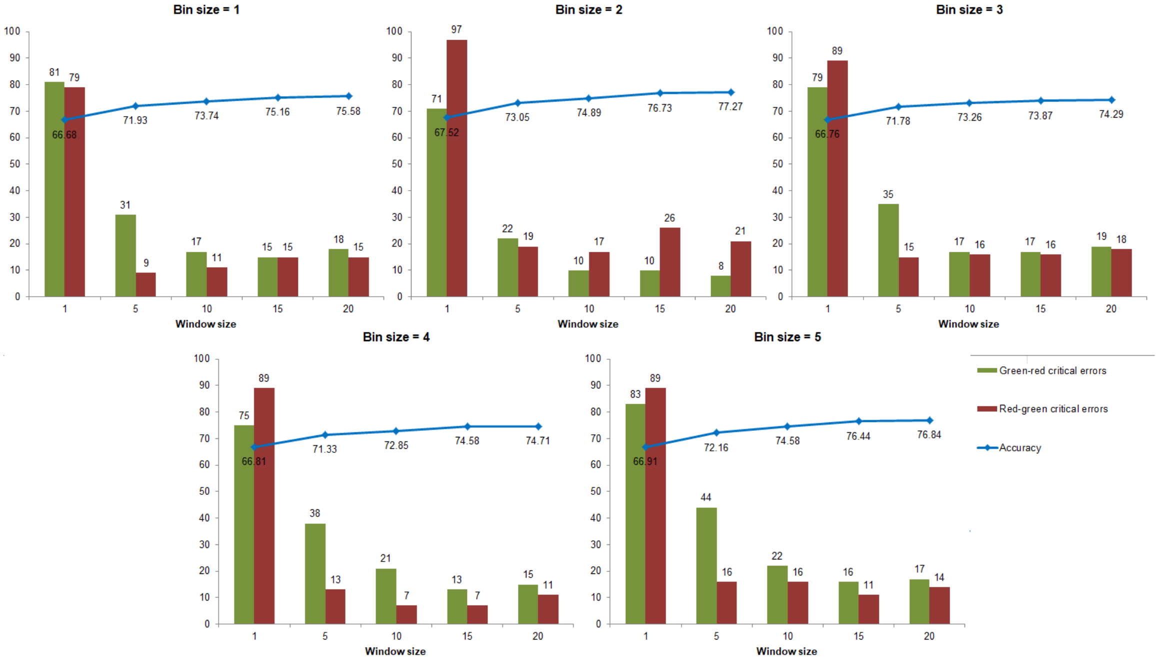

Fig. 9. Critical error and accuracy rate with feature selection, for different bin sizes using processed images with size 500 x 500 pixels

Both approaches achieved an accuracy that was above 70%. The first approach achieved the best results for a bin size equal to 3, with a window size of 20. Not using feature selection, the overall accuracy rate was 75.96%, with 36 red-green and 43 green-red misclassifications (total of 79 critical errors). For the approach using feature selection, the accuracy rate reached 76.07%, with 43 red-green and 42 green-red errors (total of 85 critical errors). In this case, even though overall accuracy increased, it resulted in worse results when it came to the critical error rate, which was the most important factor. Experiments with no feature selection achieved the best results for the number of mistakes between red and green classes; this was with a bin size of 1 and a window size of 10, resulting in 23 red-green and 12 green-red mistakes (total of 35 critical errors), with overall accuracy of 73.52%. For the approach with feature selection, the critical error rate was slightly worse (13 red-green and 27 green-red errors, with a total of 40 critical errors), in the best case, for a bin size of 2 and a window size of 10, while overall accuracy reached 74.17%.

For larger input images, the accuracy was best for a bin size of 2 and window size of 20, and it was highest both in the approach without (76.57%) and with feature selection (77.27%). For those instances, the number of critical errors was lower when smaller images were used. For the no feature selection approach, only 12 red-green and 15 green-red errors were made (27 total), while for the solution with feature selection, 8 red-green and 21 green-red mistakes were made (29 total). At the same time, the lowest critical error rate was obtained for the first approach with a bin size of 4 and window size of 15, with 11 red-green and 6 green-red mistakes (17 total), and with overall accuracy of 74.77%. For the second solution, the best results were obtained for the same bin size and window size, with 13 red-green and 7 green-red errors (20 total), and overall accuracy of 74.58%.

The presented solution quite significantly improves on standard methods when it comes to overall time complexity. It is quite common to require a few hours of training for a CNN algorithm when applied to a similar data set; this is before it can be used for an actual classification task. In the authors' previous research, the fastest solution required over 4 hours of training when a similar data set was used. The solutions with higher accuracy needed even more time before the resulting network could be used for the chosen classification method. Figure 10 indicates that even using the least optimal set of parameters for the current approach (largest image size, feature selection included and only single bin) half the time (127 minutes) is required when compared with the fastest CNN algorithm needed for training. This time only decreased when larger bin sizes were used; it should be noted, that most of the time the approach without feature selection was significantly faster. Another thing to note is that the feature selection approach, combined with larger bin size, achieved the greatest time improvement of all the approaches (starting with 127 minutes for bin size = 1, and ending with only 14 minutes for bin size 5, for 500 x 500 pixel images).

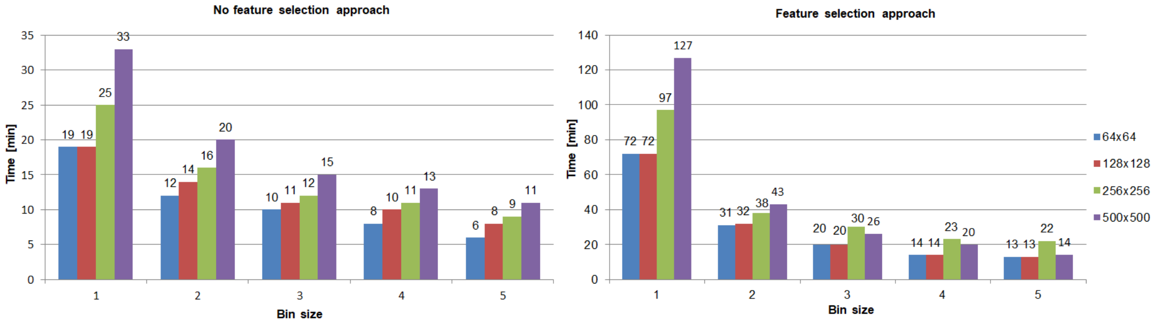

Fig. 10. Calculation time for different image sizes, for the approach without (left) and with (right) feature selection

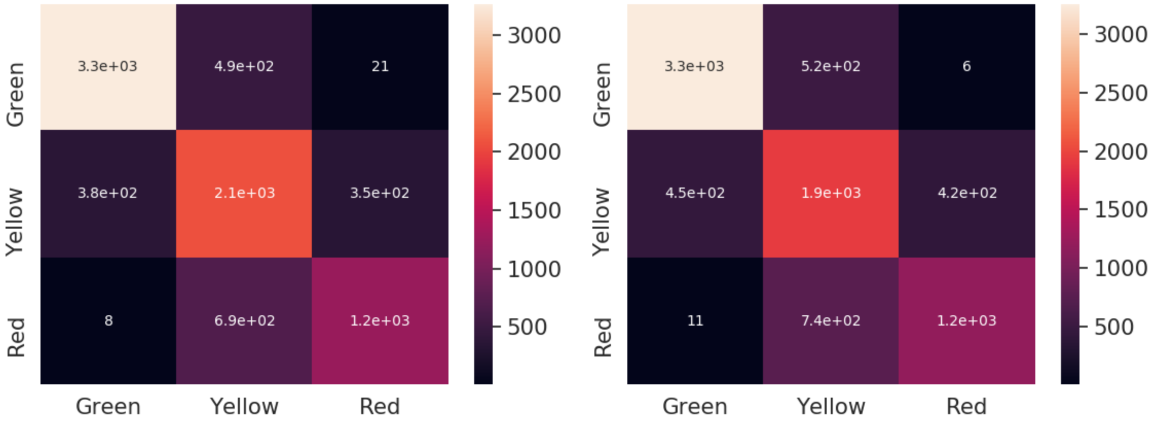

Fig. 11. Full confusion matrixes for two best solutions (for 500 x 500 pixel images): feature selection, bin size = 2, window size = 20, accuracy = 77.27% (left), no feature selection, bin size = 4, window size = 15, accuracy = 74.77% (right)

While using bins to reduce the number of checked features decreased the computation time, it did not result in increased overall accuracy of the solution and it did not diminish the critical error rate. It behaved as predicted for window size, with better accuracy rates for the larger window. It was not always the case with critical error rate, as the window size of 20 often resulted in a worse score than the smaller windows, especially in the case of the approach without feature selection. The larger images, when used as input, produced better results in terms of overall accuracy and error rate for red-green misclassifications. Although not all of the initial assumptions met the expectations when it came to algorithm performance, the presented solution was able to achieve a very good overall accuracy rate and a low critical error rate. Full confusion matrixes for best solutions in terms of overall accuracy and critical error rate are presented in Fig. 11.

CONCLUSIONS

1. The presented solution achieved a very high accuracy rate and low critical error rate (number of misclassifications between red and green classes), when using a much simpler data collection setup and more time efficient machine learning algorithms.
2. The best results in terms of critical error rate (which is also the main manufacturer requirement) were achieved for input images with size 500 × 500 pixels, bin size of 4, window size of 15, and without using feature selection (6 red-green and 11 green-red mistakes, total of 17 critical errors) with sufficient accuracy of 74.77%.
3. The best overall accuracy was achieved for images with dimensions of 500 × 500 pixels, bin size of 2, window size of 20, with feature selection and accuracy of 77.27%, with only 8 red-green and 21 green-red critical errors (29 total).
4. The presented solution can be easily applied in the actual working environment, and in that respect it has several advantages over existing methods, including the simple initial setup, low costs of required equipment, and an uncomplicated data acquisition process.

ACKNOWLEDGMENTS

The authors thank the Warsaw University of Life Sciences – SGGW for the financial support.

REFERENCES CITED

Bahl, A., Hellack, B., Balas, M., Dinischiotu, A., Wiemann, M., Brinkmann, J., Luch, A., Renard, B. Y., and Haase, A. (2019). “Recursive feature elimination in random forest classification supports nanomaterial grouping,” *NanoImpact* 15, 100179. DOI: 10.1016/j.impact.2019.100179.

Bengio, Y. (2009). “Learning deep architectures for AI,” *Foundations and Trends in Machine Learning* 2(1), 1-127. DOI: 10.1561/2200000006

BVLC AlexNet Model (2020). (https://github.com/BVLC/caffe/tree/master/mode1s/ bvlc_alexnet.), Accessed 01 April 2020.

Deng, L., and Yu, D. (2014). “Deep learning: Methods and applications,” *Foundations and Trends in Signal Processing* 7(3–4), 197-387. DOI: 10.1561/2000000039

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*, MIT Press, Cambridge, MA, USA.

Hu, J., Song, W., Zhang, W., Zhao, Y., and Yilmaz, A. (2019). “Deep learning for use in lumber classification tasks,” *Wood Science and Technology* 53(2), 505-517. DOI: 10.1007/s00226-019-01086-z

Jegorowa, A., Górski, J., Kurek, J., and Kruk, M. (2019). "Initial study on the use of support vector machine (SVM) in tool condition monitoring in chipboard drilling," *European Journal of Wood and Wood Products* 77, 957-959. DOI: 10.1007/s00107-019-01428-5

Jegorowa, A., Górski, J., Kurek, J., and 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 Tecnologia* 22(2), 189-196. DOI: 10.4067/S0718-221X2020005000205

Jemielniak, K., Urbański, T., Kossakowska, J., and Bombiński, S. (2012). "Tool condition monitoring based on numerous signal features," *Int. J. Adv. Manuf. Technol.* 59, 73-81. DOI: 10.1007/s00170-011-3504-2

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T. Y. (2017). "LightGBM: A highly efficient gradient boosting decision tree," in: *31st International Conference on Neural Information Processing Systems (NIPS'17)*, Long Beach, CA, USA, pp. 3149-3157.

Kuo, R. J., and Cohen, P. H. (1999). "Multi-sensor integration for on-line tool wear estimation through radial basis function networks and fuzzy neural network," *Neural Networks* 12(2), 355-370. DOI: 10.1016/S0893-6080(98)00137-3

Kurek, J., Kruk, M., Osowski, S., Hoser, P., Wieczorek, G., Jegorowa, A., Górski, J., Wilkowski, J., Śmietańska, K., and Kossakowska, J. (2016). "Developing automatic recognition system of drill wear in standard laminated chipboard drilling process," *Bulleting of the Polish Academy of Science. Technical Sciences* 64, 633-640. DOI: 10.1515/bpasts-2016-0071

Kurek, J., Wieczorek, G., Kruk, B. S. M., Jegorowa, A., and Osowski, S. (2017a). "Transfer learning in recognition of drill wear using convolutional neural network," in: *18th International Conference on Computational Problems of Electrical Engineering (CPEE)*, Kutna Hora, Czech Republic. pp. 1-4 DOI: 10.1109/CPEE.2017.8093087

Kurek, J., Swiderski, B., Jegorowa, A., Kruk, M., and Osowski, S. (2017b). "Deep learning in assessment of drill condition on the basis of images of drilled holes," in: *8th International Conference on Graphic and Image Processing (ICGIP 2016),* Tokyo, Japan, 10225. DOI: 10.1117/12.2266254

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

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

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in: *25th International Conference on Neural Information Processing Systems (NIPS'12)*, Lake Tahoe, NV, USA, pp. 1097-1105.

Panda, S. S., Singh, A. K., Chakraborty D., and Pal, S. K. (2006). "Drill wear monitoring using back propagation neural network," *Journal of Materials Processing Technology* 172, 283-290. DOI: 10.1016/j.jmatprotec.2005.10.021

Python programming language. Webpage available online: https://www.python.org/. Last accessed: 2020.06.17.

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," *International Journal of Computer Vision* 115(3), 211-252. DOI: 10.1007/s11263-015-0816-y

Sala, C.M., Robles, E., and Kowaluk, G. (2020). "Influence of adding offcuts and trims with a recycling approach on the properties of high-density fibrous composites," *Polymers*, 12, 1327. DOI: 10.3390/polym12061327

Schmidhuber, J. (2015). "Deep learning in neural networks: An overview," *Neural Networks* 61, 85-117. DOI: 10.1016/j.neunet.2014.09.003

Article submitted: April 22, 2020; Peer review completed: June 13, 2020; Revised version received and accepted: September 1, 2020; Published: October 29, 2020.
DOI: 10.15376/biores.15.4.9611-9624