THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 20, Number 4/2019, pp. 401–409

WOODEN DOWELS CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK

Agne PAULAUSKAITE-TARASEVICIENE[1], Kristina SUTIENE[2], Laurynas PIPIRAS[1]

[1] Kaunas University of Technology, Department of Applied Informatics, Kaunas LT-51368, Lithuania
[2] Kaunas University of Technology, Department of Mathematical Modelling, Kaunas LT-51368, Lithuania

Corresponding author: Kristina SUTIENE, E-mail kristina.sutiene@ktu.lt

Abstract. The detection of defects on the surface of a wooden dowel is required to be automatized in order to meet industry standards and to ensure consistent and cost-effective inspection. Some of these defects result from natural characteristics of wood, while the others are mechanical defects caused by manufacturing or warehousing process. Convolutional neural networks (CNN) have become one of the most widely used models of deep learning and have demonstrated the outstanding performance in image recognition tasks. This paper therefore examines the effect of defect detection performance using different CNN architectures such as LeNet, AlexNet, and GoogLeNet, which have been chosen because of the significant structural difference. The data set of images for deep learning implementation was prepared by taking pictures from real industrial environment. To ensure more accurate and consistent defect identification, the classification model was trained on a sample of images where each dowel was represented by three pictures. Then, the comparison analysis was performed using different rules of decision making for the final prediction. The experimental study showed that the proposed approach demonstrated stable detection results and good generality.

Key words: surface defect detection, convolutional neural network, deep learning, decision making.

1. INTRODUCTION

Round fluted wooden dowels are fasteners that are used for joining two or more items together. Cylindrical wooden dowels of short length are commonly used as structural reinforcements in furniture industry. Wooden dowels with the fluted edges ensure a firm grip. In addition, these flutes have the functional purpose of allowing air to escape and glue to fill the voids as one is inserting the dowel.

The purpose of this research is to create the automated classification system of dowels based on their visual inspection with the aim to identify the defects on their surface. Small wooden dowels (6 mm diameter and *30 mm length)* should be classified into two classes, namely good dowels and bad dowels. Manual inspection is a time consuming, cannot maintain long-term consistency, and needs regular breaks. Therefore, the results of manual inspection vary widely, depending on the number of factors that affect human visual inspection [1]. In the companies where defects inspection is critical, the pre-employment health check and eye testing are mandatory. However, the detection from 70% to 80% of defects is considered as a good performance of a human inspection provided that the inspector meets all the requirements and has good observation skills [2]. In the case of wooden dowels, a visual inspection usually means looking for cracks, branches included, scratches on the surface or dark colour dowels (see Fig. 1(c–d)). Dowels having at least one of the listed features are considered as dowels with defects and are assigned to the bad class. Ideally, the dowels of standard size (6×30 mm) having a solid light colour with no inclusions or defected ends are assigned to the good class (see Fig. 1(a–b)). However, one can observe visually different dowels among instances belonging to the same class. As an illustration, Fig. 1(e–f) demonstrates the dowels that look like instances of good class but are labelled as bad ones, and vice versa in Fig. 1(g–h).

In this research, the task of dowels' defects detection is mapped to the task of classification of images representing the dowels. Various machine learning technologies can be applied for image classification task, and the key challenge is to achieve high reliability of these technologies. In the case of very obvious defects such as big branches, cracks or abnormal colours, a high level of classification accuracy can be achieved as the features of different colour spaces would be enough to employ in the classification task. But the detection of small deviation from the standard colour, size or fluting pattern requires techniques which can automatically capture relevant features from the image and at the same time to be insensitive to irrelevant variations in the surface of the dowel.

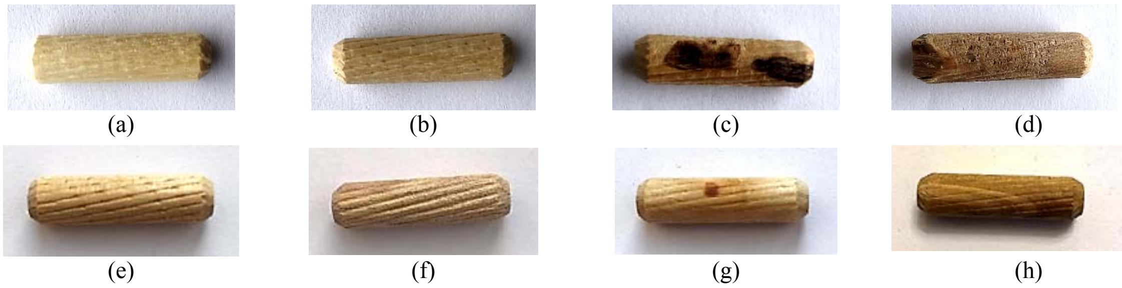

Fig. 1 – Images of wooden dowels: (a–b) perfect instances of good class, (c–d) perfect instances of bad class, (e–f) visually different instances in the bad class, (g–h) visually different instances in the good class.

Convolutional neural networks (CNNs) as a special type of multi-layer neural networks have become one of the most widely used models of deep learning and have demonstrated a very high accuracy results in various image recognition tasks [3,4,5]. Therefore, CNN has been chosen to create the automated dowels classification system in this research. Three different architectures have been implemented (LeNet, AlexNet and GoogLeNet (v1)) in order to identify defects with the highest accuracy as possible. These architectures have been chosen because of the significant structural difference.

LeNet is one of the convolutional neural network architectures employed in this paper as the technique for the deep learning. LeNet architecture was first published in 1998 for a particular purpose - to classify handwritten letters and numbers [6]. In order to improve the classification results, the network architecture has been revised several times, and the last version of this architecture has been called LeNet-5. LeNet-5 network consists of seven layers except the input layer: two convolution layers, two pooling layers, two fully connected layers, and one output layer with Gaussian connection. Compared to the other CNN architectures, LeNet has a simple and clear structure. Due to the low number of layers, the network of such architecture is easy to implement. However, this architecture can process only 32×32 pixel greyscale input images. Images of different sizes must be scaled and recoloured to meet this requirement.

In 2012 a quite similar to LeNet-5 but remarkably improved architecture called AlexNet was developed, which significantly outperformed previous architectures [7]. This architecture consists of five convolutional layers and three fully connected layers. In comparison with LeNet, AlexNet architecture is characterised by the following solutions: the max pooling is incorporated instead of average pooling, the ReLU nonlinearity is used to train the network much faster than using a common activation function like *tanh* or *sigmoid*, the data augmentation and dropout regularization allow to reduce overfitting. Moreover, it can process much larger multi-colour images, because it works with RGB images of size $227\times227\times3$ pixels.

Recently developed architectures can learn more complex features, achieve higher accuracy (on ImageNet dataset) and perform faster [8,9,10]. They differ from AlexNet by smaller size of filters, reduced number of parameters and significantly different architecture layout. One of such CNN architectures is GoogLeNet (v1) [11]. The main idea of GoogLeNet developers was to show that not everything in CNN should be performed sequentially as it was in the previous architectures. Sometimes it is useful to include components of the network performed in parallel. Therefore, the novel element „Inception module“ has been implemented in this architecture. In general, while CNN architecture requires the set of filter sizes (or max pooling) to be defined by the user, the inception module allows using of multiple convolution filters that concatenate the results. This enables the model to take an advantage of multi-level feature extraction. It was suggested to use 1×1 convolutions for dimensionality reduction of features at first and only then to perform the larger convolution using a set of 3×3 and 5×5 filters. GoogLeNet (v1) architecture consists of 22 layers in deep (including only layers with parameters) and starts with $224\times224\times3$ images. Later versions of GoogLeNet have been improved by adding a batch normalization for training regularization and inserting different inception modules with factorized convolutions, as well as modified initial operations called as stems [12,13].

2. IMPLEMENTATION

2.1. Data Set of Wooden Dowels

The data set was built of pictures depicting a wooden dowel. All pictures were taken from the top of dowel laying on the white conveyor belt under the same lightning conditions. Colour images were obtained with GigE (Gigabit Ethernet) camera, which was triggered using a position sensor. The digital images of dowels were captured in almost the same position. The size of original image was $2\,472\times3\,296$ pixels, then it was cropped down to a $2\,048\times2\,048$ essentially keeping the regions including a dowel. Finally, these images have been resized to 512×512 as the CNN used in this paper works with small dimension images.

The prepared data set of dowels was divided into two classes, specifically bad class and good class. Such classification was made by the domain-specific expert before taking the pictures. After comparison of instances from distinct classes, we made the conclusion that the dowels were always considered as bad ones if any bigger branch or crack was visible on the dowel. Dowels of dark colour were considered as defected ones. In the data set, approximately 68% of good dowels had a very light colour, no branches and no shape deformations. The remaining part of good dowels had microscopic inclusions or small deviations from standard colour. However, it should be mentioned that we cannot avoid errors because the expert is a human, and supervised learning process is performed with manually labelled dowels. This means that correctness of manually labelled dowels mostly depends on expert's competencies, skills, and precision.

2.2. Technologies and System Parameters

For the system realization, a machine learning TensorFlow framework [14] has been chosen for the following reasons. Primarily, TensorFlow incorporates certain API (Application Programming Interface) to build different architectures of convolutional neural networks. It provides methods for the creation of convolutional layers, fully connected layers, activation functions, dropout regularization, etc. Furthermore, it allows developers to transfer computational resources for CNN training to GPU and to speed up the computations or perform parallel operations on multiple platforms (CPUs, GPUs, or TPUs).

2.3. Structure of Classification System

The structure of classification system is presented in Fig. 2. The package "nets" combines three modules of selected convolutional neural networks. The package "utilities" is responsible for data preprocessing. It is composed of two modules:

- The *data preparation* module provides functions for data inspection, creation of particular sets (training, validation, and testing), generation of additional intermediate files necessary to ensure the functionality of the considered system;
- The image processing module is responsible for the transformation of the data that are fed to the selected CNN architecture. This module is separated from CNN and performed on CPU, while CNN is executed on GPU. The controlling module is the main part of the system, which combines all CNN architectures considered in the paper and data preprocessing modules into one system. Using the functionality of other modules, the controlling module performs all the main functions in the system:

- Read *Configuration file* and set CNN parameters;
- Create *Initial data set* of dowels labelled as instances of both bad and good classes;
- Provide *Testing data set*;
- Perform operations with *pre-trained CNN model* and reuse it in the future using system generated files;
- Generate the output files called as *Results* for the defect identification.

For CNN configuration, learning and testing processes, a high level API contrib is used as it supports different estimators and it was found convenient to use by many developers. The files of TFRecord format, which is a simple format for storing a sequence of binary records, were used to load the input data into the system.

The structure of classification system depicting the main components to be implemented is presented in Fig. 3. In the image data pipeline, all the data (images) are prepared in a suitable form for CNN. Various transformations including scaling, rotations, illumination, salt and pepper noise are applied to ensure that data has a good diversity. This allows to prevent overfitting and achieve the generalization during the learning process as the object of interest is represented in varying sizes, angles, poses and lighting conditions. Even minor changes such as flips or rotations are enough for neural network to interpret these pictures as distinct ones, which makes it less likely that CNN recognizes unwanted features in the sample.

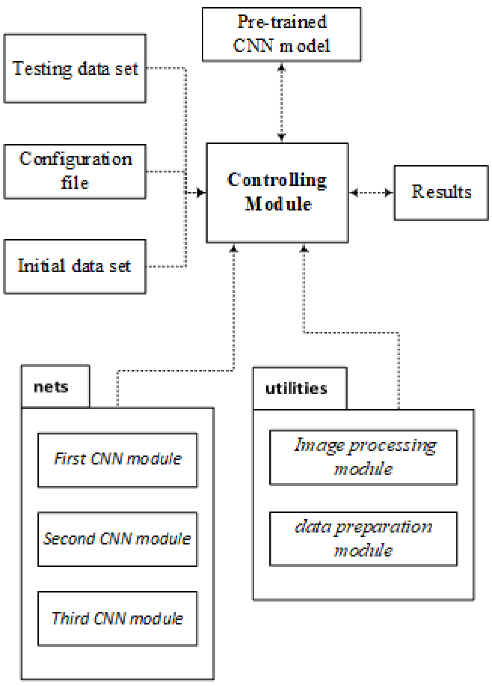

Fig. 2 – Design of system model.

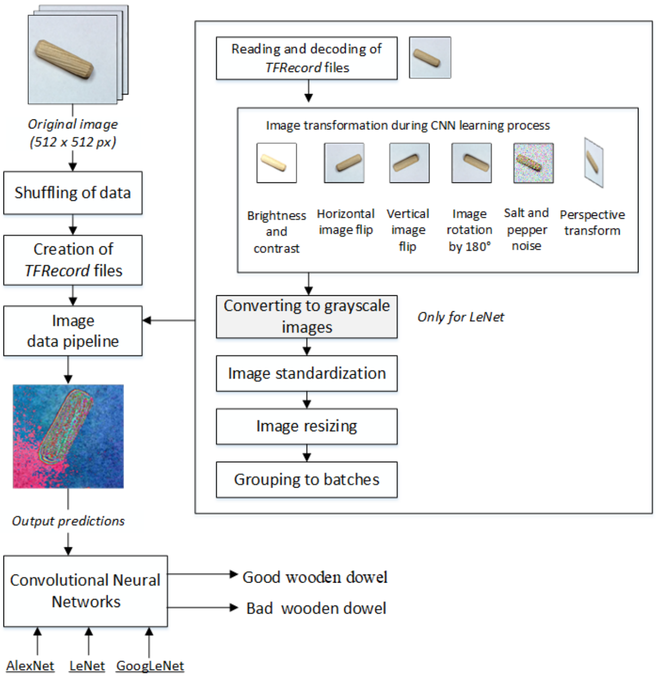

Fig. 3 – General structure of defect detection system.

Although most of the CNN architectures are able to handle RGB pictures, there are efficient architectures working with grayscale pictures as well. Therefore, the conversion to grayscale image was incorporated as well. After the transformations are applied, all pictures are standardized. Another, but not less important task, is how to pass all the data to the network. Sometimes we encountered difficulties when we needed to train a network with a very big data set. Usually one does not have enough memory, and it's time consuming process. To avoid this, all data were grouped into the small batches and fed to the network for a training. Only the set of images having the same size can be processed in batches.

The defect detection system proposed in the paper incorporates the following steps:

- Reading and decoding *TFRecords* files;
- Random selection of brightness and contrast for the image;
- Image transformation such as horizontal flip of the image, vertical flip of the image, flip equivalent to rotating the image by 180 degrees, perspective transform, etc.;
- Image conversion from RGB to grayscale image (performed for LeNet architecture only);
- Image standardization using the formula:

$$\boldsymbol{Y} = \frac{\boldsymbol{X} - \mu}{\sigma_{adj}}, \tag{1}$$

where $\boldsymbol{X}$ – the matrix of image, μ – the mean of matrix, σ_{adj} – the adjusted standard deviation, $\boldsymbol{Y}$ – the standardized matrix of image. The adjusted standard deviation σ_{adj} calculated by

$$\sigma_{adj} = \max\left(\sigma, \frac{1}{\sqrt{N}}\right), \tag{2}$$

where σ is the standard deviation of all values in the image matrix and N is the number of elements in the image matrix;

- Image scaling. All input images with initial dimension of 512×512 pixels are resized according to the CNN architecture requirements such as 32×32 pixels for LeNet, 227×227 pixels for AlexNet, and 224×224 pixels for GoogLeNet;
- Grouping the entire data set into batches. All images are stored in one buffer in which data set is shuffled and then grouped into batches. These batches are fed to the CNN for further processing.

3. EXPERIMENTAL RESULTS

3.1. Experimental Set-up

The *k*-fold cross-validation has been performed in order to estimate the prediction accuracy of a classifier induced by CNN algorithm and to choose the best model from a given set. The stratified version of this method with $k=10$ folds was used so that the correct proportion of each of the class values would be assigned to each fold as was noted in the paper [15]. The confusion matrix-based performance measures were used to examine the prediction performance of the selected CNN models. The average prediction accuracy was estimated based on how many instances were correctly classified by the model during validation and testing. To evaluate model's ability to distinguish the dowels of bad class from those which belong to the good class, the measures such as sensitivity (also known as true positive rate, TPR) and specificity (also known as true negative rate, TNR) were estimated. For the comparison reasons, we kept the discrimination threshold at 0.5.

3.2. Classification Based on One Image per Dowel

In this section, the experiment has been performed using a sample where each dowel was represented by one image. Accordingly, each of the ten folds consisted of 250 images with 50% of the labels for each class. In each iteration, nine of ten folds have been used for the training, and the classification performance was validated on the remaining fold of the sample (testing set). The classification performance was estimated for all CNN architectures presented in the paper. The mean and the standard deviation of the accuracy were estimated and depicted in Table 1.

Table 1

Classification accuracy using 10-fold cross validation

	LeNet	AlexNet	GoogLeNet
Average accuracy, %	80.20	83.14	70.58
Standard deviation, %	1.95	2.81	3.09
Max accuracy, %	83.24	88.19	77.82
Min accuracy, %	75.72	77.57	65.38

In terms of the average, the highest classification accuracy 83.14% was reached by AlexNet architecture, while the worst accuracy 70.58% was demonstrated by GoogLeNet architecture. Taking into the consideration that LeNet architecture works with grayscale images of 32×32-pixel size, the classification accuracy 80.20% achieved by this algorithm was pretty high. To estimate the stability of the classifier, the standard deviation, as well as the difference between min and max values of classification accuracy can be used for this purpose. As can be seen from Table 1, the smallest deviation was obtained using LeNet architecture, while the largest one was observed using GoogLeNet. The same tendency is approved if the difference between min and max values of accuracy is compared. In general, we can conclude that the discriminatory power of LeNet and AlexNet does not differ significantly and is remarkably better in comparison with GoogLeNet architecture.

3.3. Classification based on three images per dowel

The second experiment as an alternative to previous experiment was performed in order to examine how the classification accuracy changes if three images of the same dowel are used instead of one image. Hence, the data set included three images from different sides of the same dowel. The reason of this decision was made after having analysed the data set of images where each dowel was captured from one side by the top camera. It was noticed that that small size defects (e.g. small branches) were not always visible from above (see Fig. 4). Accordingly, we expected that using dowel pictures taken from three different sides would increase the identification of defects typical for the dowels of bad class. For this purpose, each dowel has been photographed at the same conveyor but from the top, right side, and left side (see Fig. 5).

According to the provided decision making algorithm (Fig. 6), the wooden dowel was classified as good one if all three CNN outputs resulted in the class of good dowels. But if at least one of the CNN outputs predicted the dowel as the instance of bad class, it was classified as a bad dowel. The reason of such rule having been introduced was based on the analysis of the data set where almost 34% of bad dowels had a defect visible only from one side. However, for the comparison analysis, the decision principle on the basis of voting majority rule has been performed as well.

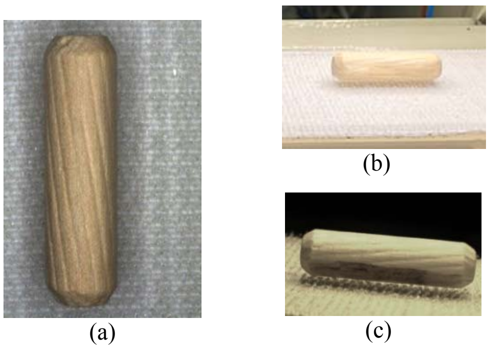

Fig. 4 – Pictures of the same dowel from different sides: a) from top camera, b) from left side camera, c) from right side camera.

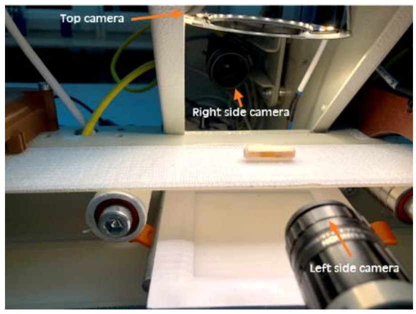

Fig. 5 – Conveyor of wooden dowels with three cameras from different sides

For this experiment, the data set of 7 500 images containing pictures of 2 500 dowels from three different sides was composed. The decision making algorithm for the final prediction of the class to which the dowel belongs is presented in Fig. 6.

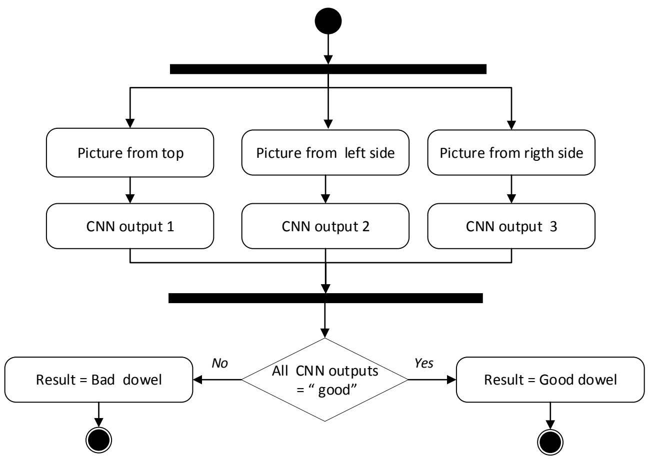

Fig. 6 – The proposed decision making algorithm for the dowels classification problem.

In the following, the testing results of prediction performance based on three images per dowel are presented. In comparison, the ability of a classifier to identify the defect is mainly considered. Figs. 7–9 outline the confusion matrix-based measures on the test set of 750 dowels. As can be seen in Figs. 7–9, the usage of three images per dowel during the learning process ensured both higher accuracy and higher specificity in comparison with the case when one image per dowel was used. It means that one can be nearly certain that the model is able to identify the defect for actually defected dowels. The increase of these measures was observed for all CNN architectures used in the experimental study. In terms of sensitivity measure, it was observed that the sensitivity was increased when the majority voting rule was applied but decreased if the final prediction was made based on the proposed decision making algorithm (see Fig. 6) in comparison with one image classification experiment. The same tendency was demonstrated by all CNN architectures as well.

		Predicted class	
		Good	Bad
Target class	Good	318 42.40%	57 7.60%
	Bad	89 11.87%	286 38.13%
Accuracy		80.53%	
Sensitivity		84.80%	
Specificity		76.27%	

a) one image per dowel used

		Predicted class	
		Good	Bad
Target class	Good	308 41.07%	67 8.93%
	Bad	61 8.13%	314 41.87%
Accuracy		82.93%	
Sensitivity		82.13%	
Specificity		83.73%	

b) three images per dowel used and the decision making algorithm (Fig. 6) applied

		Predicted class	
		Good	Bad
Target class	Good	319 42.53%	56 7.47%
	Bad	75 10.00%	300 40.00%
Accuracy		82.53%	
Sensitivity		85.07%	
Specificity		80.00%	

c) three images per dowel used and majority voting rule applied

Fig. 7 – Testing results of different experiments using LeNet architecture.

		Predicted class	
		Good	Bad
Target class	Good	334 44.53%	41 5.47%
	Bad	78 10.40%	297 39.60%
Accuracy		84.13%	
Sensitivity		89.07%	
Specificity		79.20%	

a) one image per dowel used

		Predicted class	
		Good	Bad
Target class	Good	325 43.33%	50 6.67%
	Bad	42 5.60%	333 44.40%
Accuracy		87.73%	
Sensitivity		86.67%	
Specificity		88.80%	

b) three images per dowel used and the decision making algorithm (Fig. 6) applied

		Predicted class	
		Good	Bad
Target class	Good	338 45.07%	37 4.93%
	Bad	62 8.27%	313 41.73%
Accuracy		86.80%	
Sensitivity		90.13%	
Specificity		83.47%	

c) three images per dowel used and majority voting rule applied

Fig. 8 – Testing results of different experiments using AlexNet architecture.

		Predicted class	
		Good	Bad
Target class	Good	278 37.07%	97 12.93%
	Bad	125 16.67%	250 33.33%
Accuracy		70.40%	
Sensitivity		74.13%	
Specificity		66.67%	

a) one image per dowel used

		Predicted class	
		Good	Bad
Target class	Good	265 35.33%	110 14.67%
	Bad	78 10.40%	297 39.60%
Accuracy		74.93%	
Sensitivity		70.67%	
Specificity		79.20%	

b) three images per dowel used and the decision making algorithm (Fig. 6) applied

		Predicted class	
		Good	Bad
Target class	Good	287 38.27%	88 11.73%
	Bad	101 13.47%	274 36.53%
Accuracy		74.80%	
Sensitivity		76.53%	
Specificity		73.07%	

c) three images per dowel used and majority voting rule applied

Fig. 9 – Testing results of different experiments using GoogLeNet architecture.

To summarize, in terms of classification accuracy the final decision for the dowel defect detection based on the majority vote rule allowed us to achieve classification results better than using one image classification, but slightly worse than the presented decision making algorithm. The experimental results have shown that the majority vote rule has the advantage of reducing classification error of good instances. On the other hand, the gain of specificity was larger than the loss in sensitivity when the presented decision making algorithm was applied, thus the general classification accuracy resulted in 74.93% for GoogLeNet architecture, 82.93% for LeNet architecture, and 87.83% for AlexNet architecture, which are the best results in our experimental study. The classification performance using GoogLeNet architecture was not as we expected. There are several possible explanations for this result. First, GoogLeNet(v1) is a very deep classifier, therefore it may result in vanishing/exploding gradients. Consequently, it is very hard to learn the parameters of the former layers in the network. This problem becomes worse as the number of layers in the architecture increases and this can make the model to almost stop learning. Second, 1x1 convolutions lead to the reduction in data by decreasing resolution, which in result may cause the loss of relevant information. GoogLeNet(v1) is a very specific architecture suitable for particular problems to be solved. Our experimental results have shown that GoogLeNet demonstrated the moderate performance in identifying defects of dowels we had in our study.

4. CONCLUSIONS

The learning capabilities of convolutional neural network for dowels classification task were explored in the paper. The research provides the comparison study of three CNN architectures (LeNet, AlexNet, and GoogLeNet(v1)) for dowels defect detection only on the basis of their external appearance. Issues related to the physical properties of the dowel indicating its unsuitability for use have not been considered.

In the paper, the experiments including different number of images per dowel have been performed. The obtained accuracy results satisfy our expectations. In general, the classification using three images per dowel allowed us to improve the discriminatory power of a classifier. The overall average accuracy using three images was increased by between 2.40% and 4.53% (depending on the architecture used) using the decision making algorithm proposed in the paper, and between 2% to 4.40% using the majority voting rule. These two algorithms demonstrated rather similar prediction accuracy but provided different results in terms of sensitivity and specificity. The proposed decision making algorithm has demonstrated the best performance for dowel defect detection in terms of classification accuracy, while the majority voting algorithm has provided the better identification of dowels of a good class. The best classification accuracy achieved in this study was 87.83% when CNN with the AlexNet architecture was used as a base classifier, which is in line with the findings published in the comprehensive reviews [3,16]. Given these points, it can be concluded that the prediction of dowel quality classifying them into two classes is rather reasonable taking into account that the performance of human visual inspection, which usually varies from 70% to 80% of defects. Presumably, we claim that the domain-specific application we have tackled with in the paper resulted in the classification performance, which could still be improved, due to visually challenging instances within the class we received from the manufacturer as was explained in the paper. Investigations of 3D image recognition technologies or classification based on unsupervised learning algorithms could be considered as the topic for further research and development.

REFERENCES

1. J.A. MELCHORE, *Sound practices for consistent human visual inspection*, AAPS PharmSciTech, **12**, *1*, pp. 215–221, 2011.
2. R.L. LEVERSEE, J.G. SHABUSHNIG, *A survey of industry practice for the visual inspection of injectable products*, Presented report at PDA Annual Meeting, 2008.
3. W. RAWAT, Z. WANG, *Deep convolutional neural networks for image classification: A comprehensive review*, Neural Computation, **9**, *29*, pp. 2352–2449, 2017.
4. S. SLADOJEVIC, M. ARSENOVIC, A. ANDERLA, D. CULIBRK, D. STEFANOVIC, *Deep neural networks based recognition of plant diseases by leaf image classification*, Computational Intelligence and Neuroscience, p. 3289801, 2016.
5. L. MOU, P. GHAMISI, X.X. ZHU, *Deep recurrent neural networks for hyperspectral image classification*, IEEE Transactions on geoscience and remote sensing, **55**, *7*, pp. 3639–3655, 2017.
6. Y. LECUN, L. BOTTOU, Y. BENGIO, P. HAFFNER, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, **86**, *11*, pp.2278–2324, 1998.
7. A. KRIZHEVSKY, I. SUTSKEVER, G.E. HINTON, *ImageNet Classification with Deep Convolutional Neural Networks*, NIPS'12 **(1)**, pp. 1097–1105, 2012.
8. S. WU, S. ZHONG, Y. LIU, *Deep residual learning for image steganalysis*, Multimedia Tools and Applications, **77**, *9*, pp. 10437–10453, 2018.
9. K. SIMONYAN, A. ZISSERMAN, *Very deep convolutional networks for large-scale image recognition*, ICLR conference, 2014.
10. K. HE, X. ZHANG, S. REN, J. SUN, *Deep residual learning for image recognition*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.
11. C. SZEGEDY, W. LIU, Y. JIA, P. SERMANET, S. REED, D. ANGUELOV, D. ERHAN, V. VANHOUCKE, A. RABINOVICH, *Going deeper with convolutions*, IEEE conference on Computer Vision and Pattern Recognition, 2015.

12. S. IOFFE, C. SZEGEDY, *Batch normalization: accelerating deep network training by reducing internal covariate shift*, Proceedings of 32nd International Conference on Machine Learning, pp. 448–456, 2015.
13. C. SZEGEDY, S. IOFFE, V. VANHOUCKE, A. ALEMI, *Inception-v4, Inception-ResNet and the impact of residual connections on learning*, AAAI Conference on Artificial Intelligence*,* pp. 1–12, 2016.
14. M. ABADI, *TensorFlow: An open source machine learning framework for everyone*, available at www.tensorflow.org
15. R. KOHAVI, *A study of cross-validation and bootstrap for accuracy estimation and model selection,* Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI'95, **2**, pp. 1137–1143, 1995.
16. M.Z. ALOM, T.M. TAREK, C. YAKOPCIC, S. WESTBERG, P. SIDIKE, M.S. NASRIN, M. HASAN, B.C. ESSEN, A. AWWAL, V.K. ASARI, *A state-of-the-art survey on deep learning theory and architectures*, Electronics, **8**, *3*, p. 1–66, 2019.

Received March 12, 2019