

# SVM Algorithm for Industrial Defect Detection and Classification

*Krzysztof Lalik*<sup>1[0000-0003-4277-9801]</sup>, *Mateusz Kozek*<sup>1[0000-0001-7222-2970]</sup>, *Paweł Gut*<sup>1[0000-0003-2351-1834]</sup>, *Marek Iwaniec*<sup>1[0000 0002 1224 4753]</sup>, and *Grzegorz Pawłowski*<sup>2</sup>

<sup>1</sup>AGH University of Science and Technology, Faculty of Mechanical Engineering and Robotics, Kraków, Poland

<sup>2</sup>HandCraft Polska Sp. z o.o., ul. Węgierska 12 B, 33-340 Stary Sącz

**Abstract.** This article presents a new algorithm for recognizing defects and discontinuities. It is a neural classification algorithm of the SVM class used for the vision system in the technological sequence. At the basis of the used method of Support Vector Machines (SVM) lies the concept of decision-making space, which is divided by building boundaries separating objects with different class affiliation, that is, defects and discontinuities. The Support Vector Machines method is supposed to perform classification tasks by constructing in a multidimensional space hyperplane separating cases belonging to different classes.

## 1 Introduction

To match human capabilities, specialised vision equipment is used in quality control operations. These include cameras and dedicated image processing equipment. Such a system allows to eliminate adverse human factors such as inattention, fatigue or fatigue resulting from monotonous work. Despite elimination of these unfavorable factors, human advantage over such a system takes place in response to changing and unpredictable situations. This is due to the fact that a specific strictly defined decision-making algorithm is implemented. To improve the performance of a vision inspection system, a learning algorithm using artificial neural networks is used. Thanks to such a procedure we obtain quality control working with high speed, high efficiency and repeatability. Quality control is an important element of production and should be carried out after every complex operation. Earlier detection of damage to the element allows to save time and money resulting from the processing of the element which ultimately will not meet the quality requirements.

A complementary element of vision systems is software for image processing and analysis [1–3]. An integral part of the packages provided by vision system manufacturers is usually a number of algorithms that allow the creation of different control and measurement applications. Typical examples include algorithms for pattern matching, edge detection and OCR [4–6].

There are many vision programs available on the market, and choosing the right software significantly reduces integration time, costs and - in the future - the possibility of making

modifications. The main factor determining the selection of the program is the complexity of the vision application. [3,7,8]. For simple applications such as barcode reading, simple measurements or checking the completeness of parts, software with a graphical interface is sufficient. Sometimes, however, even for such applications, it is necessary to use vision libraries, because under changing conditions of lighting, background or uniqueness of code execution (often occurs when punching DataMatrix code on metal parts), only the use of specialized algorithms allows to execute the application correctly.

## 2 Methodology

During research, a prototype of a device for detecting surface defects of furniture laminate was created and a vision algorithm and neural image evaluation system were developed.

The vision system being the subject of the research provides real control, even on large surfaces of semi-finished products, as is the case with continuous products - laminated wooden boards. Two methods have been developed during the research for vision inspection of the surface:

- When dealing with a homogeneous surface structure, a programmed description of the appearance of defects is used, e.g. scratches, spots, holes, etc. With the help of special software produced during research works, such forms are found in the images of the controlled product,
- When the product has a constant, repeating geometric texture, the system compares the image taken with a pre-programmed pattern, i.e. a memorized reference image.

To recognize defects and discontinuities, a neural classification algorithm of the SVM class was developed during research. At the basis of the Support Vector Machines (SVM) method used is the concept of decision space, which is divided by building boundaries separating objects with different class affiliation, i.e. defects and discontinuities.

## 3 SVM Algorithm

The SVM (Support Vector Machine) algorithm is a very popular tool for machine learning, whose birth dates back to 1998, when Professor Vladimir Vapnik improved the shaping of the neural network structure and defining the problem of learning by eliminating the disadvantages of MLP and RBF neural networks. The operation of this network consists in constructing a hyperplane or a set of hyperplane in space of high or infinite dimension [9–11]. The most important thing here is the weighting process, which maximizes the separation margin between the nearest points in this data space, which defines the different classes. This means that during the construction of the model, only the most difficult separated points of the space are considered, which define the so-called supporting (supporting) vectors. Figure 1 is used to compare the operation of each network (the right one refers to the SVM network).

SVM networks form a two-layer neuropathic structure that uses different types of activation functions (i.e. linear, radial, polynomial, or sigmoid) as well as a specific way of learning by means of square programming, which is distinguished by the fact that there is only one global minimum. SVM networks are mainly designed for classification issues, where one class is separated by as much margin as possible. In order to do this, a maximum margin must be set, which is achieved with the help of the carrier vector machine, which can be considered as an extension of the perceptron model. [9,12].

## 4 Measurement stand

This subsection shows the relationships between the elements (Fig 2) of the control and measurement system. The PLC receives information in the form of electrical signals from the machine feeding whether the element has been fed. The PLC then sends a command to trigger the camera after the PROFINET connector. After receiving the command, the camera takes a photo and submits it to preliminary analysis and sends the photo and information back to the PLC after the PROFINET connector. The PLC sends the received image to the cloud via the OPC UA protocol. In the cloud, a decision algorithm based on an artificial neural network is implemented and the inspection result is transmitted to the PCL where the final decision on the correctness of the element is made.

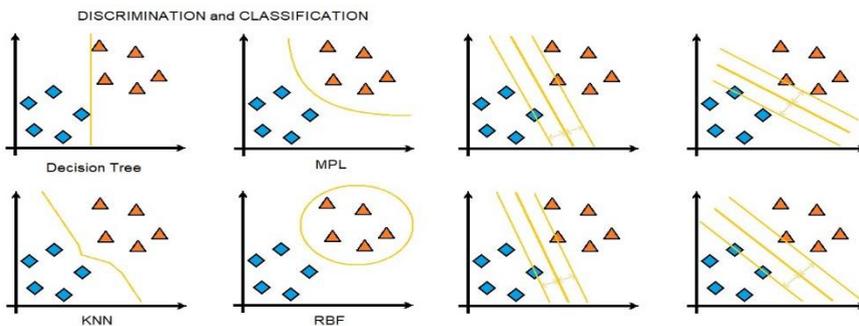


Fig. 1. Comparing the SVM algorithm to other networks.

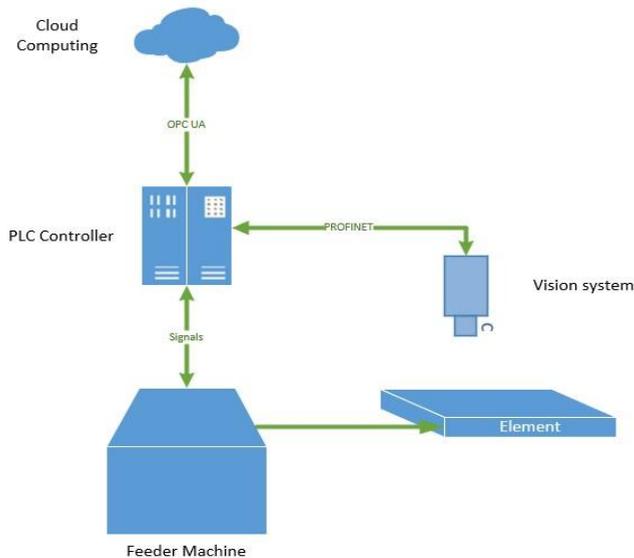
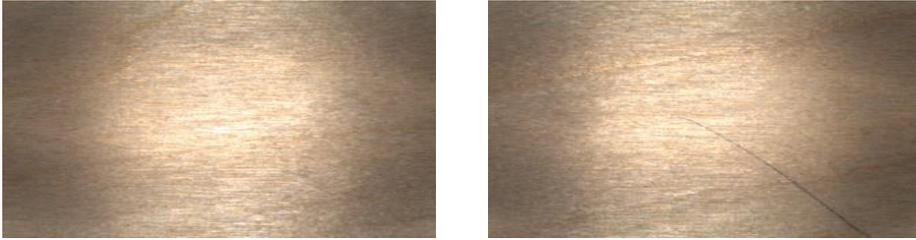


Fig. 2. Connection diagram between system components.

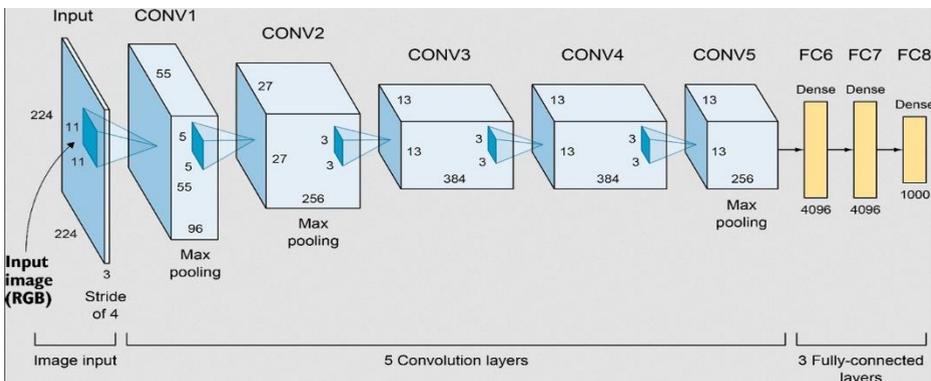
The images were pre-processed by the camera’s internal algorithm, but this method was extremely sensitive to light conditions, as well as to the shape of the scratch and its color. Therefore, an additional algorithm located in the cloud was required.



**Fig. 3.** Sample images of the teaching set: (a) Correct element, (b) Element with scratch defect.

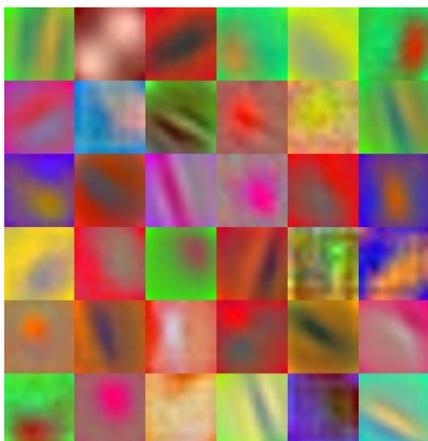
## 5 Results

To improve the performance of the inspection system, it was decided to use an additional decisionmaking algorithm with the role of an artificial neural network. Due to the work with images, it was decided to use a neural network with layers that perform a convolution operation. As a result of the image with a filter, we get a new image, in which each pixel has been created based on its neighborhood. Depending on the choice of the filter, you can get a blurred, sharpened image or an image with highlighted edges. It has been decided to apply the transfer learning method, which consists in modifying layers of previously trained neural network. This process is one of the elements of deep learning. This solution has been chosen because the teaching of the neural network from the beginning requires a lot of time and high computing power. To transfer the learning, we used alexnet [13] due to its structure (Fig 4), which is ideal for image analysis.



**Fig. 4.** The structure of alexnet base DNN [14].

To teach the neural network we used a database of 540 images of the material without defects and 540 images containing 4 different defects. These photos were taken with the KEYENCE IV-500CA video camera. The images that were subjected to the convolution operation of the first time can be visualized as shown in Fig 5.



**Fig. 5.** Image visualization after the first convolution.



**Fig. 6.** Image visualization after the last convolution.

The images are subjected to the convolution operation of five times, the result of the last convolution operation is shown in Fig 6.

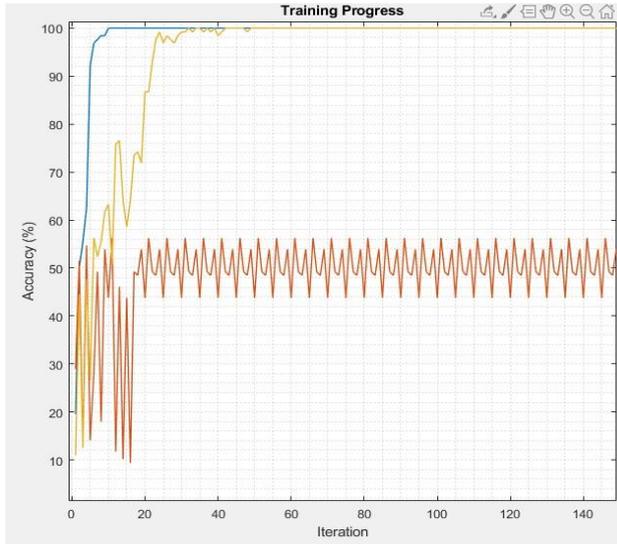
Images must be properly prepared before they are subjected to the convolution operation. After importing the images into the matlab environment to a dedicated ImageDatastore type, several image pre-processing operations must be performed. It was decided to use the augmentedImageDatastore function, which specifies the dimensions to which the image is to be scaled, i.e. 227 x 227 px. Because the network has been pre-learned on photos with these dimensions. The alexnet has been trained on a huge collection of photos. To adapt the network to the search for essential features, the classification and decision layer must be modified. Use the Layer command to find the layers that will be modified. The result of this command is shown in Fig 7.

```
25x1 Layer array with layers:
 1 'data' Image Input 227x227x3 images with 'zerocenter' normalization
 2 'conv1' Convolution 96 11x11x3 convolutions with stride [4 4] and padding [0 0 0 0]
 3 'relu1' ReLU
 4 'norm1' Cross Channel Normalization cross channel normalization with 5 channels per element
 5 'pool1' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0 0 0]
 6 'conv2' Grouped Convolution 2 groups of 128 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]
 7 'relu2' ReLU
 8 'norm2' Cross Channel Normalization cross channel normalization with 5 channels per element
 9 'pool2' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0 0 0]
10 'conv3' Convolution 384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
11 'relu3' ReLU
12 'conv4' Grouped Convolution 2 groups of 192 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
13 'relu4' ReLU
14 'conv5' Grouped Convolution 2 groups of 128 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
15 'relu5' ReLU
16 'pool5' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0 0 0]
17 'fc6' Fully Connected 4096 fully connected layer
18 'relu6' ReLU
19 'drop6' Dropout 50% dropout
20 'fc7' Fully Connected 4096 fully connected layer
21 'relu7' ReLU
22 'drop7' Dropout 50% dropout
23 'fc8' Fully Connected 1000 fully connected layer
24 'prob' Softmax softmax
25 'output' Classification Output crossentropyex with 'tench' and 999 other classes
```

**Fig. 7.** Description of alexnet layers.

From the figure marked Fig 7 you can read that the layers to be modified are marked 23 'fc8'. It is a layer containing 1000 neurons indicating 1000 solutions and 25 'output' layers. This layer classifies the image based on the neuron indications from layer 23. Replacement

operations can be performed using matlab commands or using the available Deep Network Designer tool.



**Fig. 8.** Comparison of learning depending on the parameters to be set.

After the modifications are made, it is necessary to start the process of learning the network on the base set. However, you should set the learning options beforehand. This can be done with `trainingOptions()` command. Depending on the settings, different learning rates are achieved, and in critical cases, it may cause the network to be impossible to learn (Fig 8). The progress of network learning can be observed in the Training Progress window shown in Fig 9.

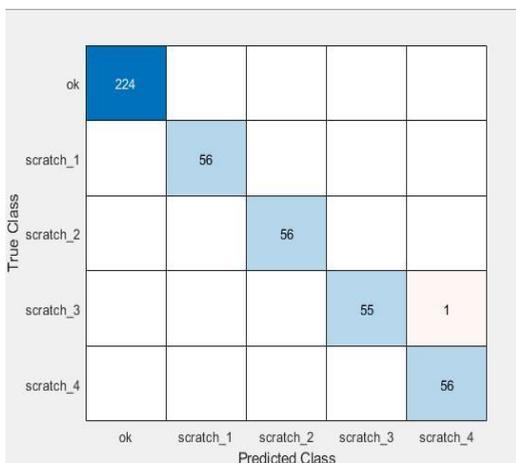
Once the learning process is complete, you can test the operation of the learned network. These tests were made on a collection of 448 photos for this collection and the surprisingly effective classification is 99.78%. Figure 10 shows a confusion matrix.

## 6 Conclusions

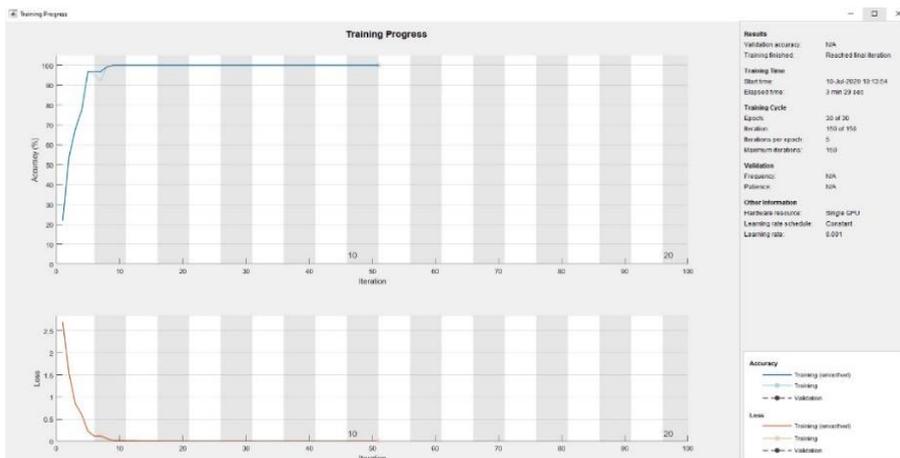
The proposed mechanism was first integrated for automation in the furniture industry. The presented results prove the high effectiveness of the use of Artificial Neural Networks in detecting defects of wooden laminates. It outperforms classical morphological algorithms, which are usually integrated with vision systems. Thanks to the application of the proposed algorithm a lower sensitivity to ambient light was achieved. In the classical approach, this influence is reduced by increasing the power of the illuminating unit. However, in the case of wooden surfaces that are covered with reflective laminate, such a classic solution does not lead to an improvement in detection efficiency, but even reduces it. In this case, the light reflected from the reflective surface is an additional interference for classical morphological operations. The proposed convolution network allowed to effectively eliminate this effect.

The structure of the proposed solution is in line with Industry 4.0 standard, which is more and more widely used in automation and robotics. In this case, thanks to the OPC UA server, it was possible to communicate the automatic tester station with any programming

platform on which the kernel of the neural decision system was installed. This is a big innovation in relation to Edge computing, where dedicated, specialized and high-performance computing systems are required, which must be parameterized separately for each new product.



**Fig. 9.** Learning progress of a modified web on a base set.



**Fig. 10.** Error matrix for the test samples.

## Acknowledgments

This paper was written under grant entitled „Innowacyjnatechnologiamasowejprodukcjimeblikastomizowanych HandCraft Polska Sp. z o.o.”

## References

1. S. Weyer, M. Schmitt, M. Ohmer, D. Gorecky IFAC-Papers OnLine (2015).
2. H. Golnabi, A. Asadpour Robotics and Computer-Integrated Manufacturing (2007).
3. Z. Liu, H. Ukida, P. Ramuhalli, K. Niel Integrated imaging and vision techniques for industrial inspection: Advances and applications (2015).
4. Y. Bengio Foundations and Trends in Machine Learning (2009).
5. M. Gupta International Series in Operations Research and Management Science Preprint 1807.07987 (2019).
6. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S.Reed, D. Anguelov, D. Erhan, V. VanhouckE, A. Rabinovich Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Preprint 1409.4842) (2015).
7. G. Wang, G.B. Giannaki, J. Chen IEEE Transactions on Signal Processing ISSN 1053587X Preprint 1808.04685 (2019).
8. K. Lalik, M. Kozek, I. Dominik, P. Lukasiewicz (2020).
9. N. Guenther, M. Schonlau. Stata Journal (2016).
10. C.C. Chang, C.J. Lin ACM Transactions on Intelligent Systems and Technology (2011).
11. C. Hsu, C. Chang, C.J. Lin BJU international, Preprint 0-387-31073-8 (2008).
12. M.A. Farquad, I. Bose Decision Support Systems (2012).
13. A. Krizhevsky, I. Sutskever, G.E. Hinton Commun. ACM, 60, 84–90 (2017).
14. M. Elgendy Deep Learning for Vision Systems, Manning Publications (2020).