

# Reducing Pseudo-error Rate of Industrial Machine Vision Systems with Machine Learning Methods

B. Szűcs<sup>1</sup>, Á. Ballagi<sup>2</sup>

<sup>1</sup>Audi Hungaria Zrt., Product Unit Diesel I4/V6,  
Audi Hungária út 1., H-9027 Győr, Hungary  
e-mail: [balazs.szucs@audi.hu](mailto:balazs.szucs@audi.hu)

<sup>2</sup>Széchenyi István University, Automation Department,  
Egyetem tér 1., H-9026 Győr, Hungary  
e-mail: [ballagi@sze.hu](mailto:ballagi@sze.hu)

**Abstract:** Nowadays machine learning and artificial neural networks are hot topic. These methods gains more and more ground in everyday life. In addition to everyday usage, an increasing emphasis placed on industrial use. In the field of research and development, materials science, robotics and thanks to the spread of Industry 4.0 and digitalization, more and more machine learning based systems introduced in production. This paper gives examples of possible ways of using machine learning algorithms in manufacturing, as well as reducing pseudo-error (false positive) rate of machine vision quality control systems. Even the simplest algorithms and models can be very effective on real-world problems. With the usage of convolutional neural networks, the pseudo-error rate of the examined system reducible.

**Keywords:** *machine learning; classification; convolutional neural network; machine vision; industry 4.0*

## 1. Introduction

This paper presents the usability of ML (*machine learning*) and ANNs (*artificial neural networks*) in the domain of industrial image processing. The goal of the research is to find, develop and test practical use-cases of ML and AI (*artificial*

*intelligence*) methods in production, focusing on automotive industry and manufacturing.

Nowadays the high quality standards necessitate the use of in-line, automatic measurement and supervision systems. One of these systems is the vision system, which performs part inspections and automatic measurements. For the simplest task like object detection, or advanced functions like distance and/or angle measurements, position or colour detection, surface scan and measurement, these systems are ready-to-go solutions. These methods are easy to use, when the inspected images are very similar to the images used to configure and tune the system. However, when the images contain unusual parts or external disturbances, the pseudo-error rate of the supervisions system can be increasing. To fight these problems ML and AI methods offers possible solutions.

In the serial production, every day very big amount of data generated by automated systems. The images and the measurement data is stored for years to support the quality assurance. These historical datasets are good starting points of ML and AI methods. In the following sections, we show how to use these data to reduce pseudo-error rate of a machine vision system and we present a potential, real-world use case.

## **2. Framing the problem**

The problem, which the methods presented on, is the following: An automated station, which integrated into the assembly line, checks the electrical connectors of an internal combustion engine. The station checks, whether the connector attached and properly locked, or not. The inspected electrical connector of an internal combustion engine seen on Figure 1. (The part numbers and identifiers are deliberately blurred.)



*Figure 1. Inspected electrical connector*

The algorithm of the station searches for two parallel edges in a manually predetermined region of interest, these represents the upper edge of the plastic label and the lock of the connector, and then calculates the distance of edges. The edge detection based on the changes of the average pixel values (intensity change) of the picture. (Figure 2, the edges used to the calculation marked green.) If the distance is in a predefined interval, the algorithm classifies the connection as OK. As seen, the connector attached fully and locked, but in some cases, the vision system cannot detect these edges, thus cannot compute the distance, the result will be NOK (*not okay*).



Figure 2. The measurement of the distance between upper edge of the plastic label and the connector lock

Typical pseudo-errors (false positives) are on Figure 3. For a human inspector it is unambiguous that the connectors are fully attached and locked, but the algorithm of the machine vision system cannot detect one or both of the edges, and cannot place the measuring window on the pictures, thus the result of the inspection will be NOK.



Figure 3. Typical pseudo-errors

The problem is that some cables not properly arranged and they cover the edges, thus the edges are not detectable. The shadows on the inspected area, the changes in lighting or the colour of the material can cause the same effect.

The goal is to reduce these errors by learning high-level representations of the well-attached connector, with which the system still able to recognize the good connections, even when disturbances present in the images, and the edges not detectable.

### **3. Methods**

#### **3.1. Toolbox**

We used Python to design, build and evaluate machine learning algorithms and models. Python is an interpreted, high-level, general-purpose programming language, which is very popular in data science, ML and AI. To build models, we used the libraries scikit-learn [1], TensorFlow [2] and Keras. These libraries are open-source and highly supported by the data science and AI community.

#### **3.2. The dataset**

The dataset used in this research contains 22755 images. In Table 1, the original ratio of the OK – NOK images shown. The images collected and categorized by the automated control station and the result of thirty days productions.

*Table 1. The original ratio of the OK – NOK images*

<i><b>OK</b></i>	<i><b>NOK</b></i>	<i><b>Sum</b></i>
21844	911	22755

#### **3.3. Image pre-processing**

The computation cost and the training time are two notable factors when we design and train ANNs. In special cases, like in the manufacturing, a lot of environmental variable fixable. In this case, the set of inspected items is limited to two to three types; the position of the engines, thus the connector's positions varies in a very limited range. That means the region of interest, thus the size of the image is reducible. To reduce the size of the input dimension and to speed up the human inspection too, the images cropped and rotated. The original and the pre-processed images shown on Figure 4, the attributes of the pictures in Table 2.



Figure 4. Original and pre-processed image

Table 2. Parameters of the original and the pre-processed images (dimension: input of the ANNs, width x height)

<i>Parameter</i>	<i>Original</i>	<i>Pre-processed</i>
Width	1200 px	510 px
Height	1000 px	540 px
Size	70 kB	30 kB
Dimension	1 200 000	275 400

### 3.4. Dataset revision

To determine the true numbers of OK and NOK images, including false positive (false NOK) and false negative results (false OK), revision of the categories had to be made. To speed up the process, we selected the unambiguous pictures with a very useful ML algorithm, the k-means++ [3].

The k-means algorithm mostly used for data mining. It clusters the given data by separating samples in  $n$  groups of equal variance, minimizing a criterion known as the inertia, also known as within-cluster sum-of-squares. (Eq. 1.) The algorithm requires specifying the number of clusters; in this case, the number of classes chosen to be two, thus the model separates the unambiguous pictures from those who contains disturbing factors like cables and shadows. The algorithm divides a set of  $n$  samples into  $C$  disjoint clusters, each described by the mean, or the  $\mu_j$  inertia of the samples in the cluster [3].

$$\sum_{i=0}^n \min_{\mu_j \in C} (\|x_i - \mu_j\|^2) \quad (1)$$

In this case, the similarity measure was the Euclidean distance of the sample and the cluster centroid. From the analysis of the disturbances, it is stated, when an unambiguous disturbing factor, for example a cable over the connector is present, the average pixel value of the sample is highly different from the value of the disturbance free samples. The ambiguous samples, which contains smaller disturbances like partial attachment, we manually reviewed.

The result of clustering the NOK images seen on Figure 5. The plot axes are dimensionless. To represent the samples in a 2D plot, we generated the coordinates from the pixel values of images. To this, we used principal component analysis (PCA) [4]. The PCA is widely used in machine learning. The essence of PCA is to reduce the dimensions of a large data set while keeping the present variance as best as possible.

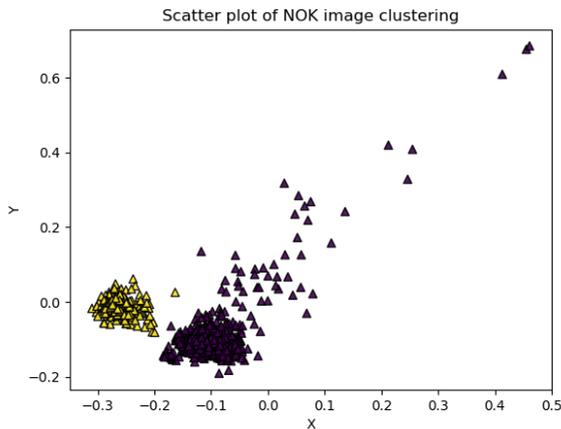


Figure 5. Scatter plot of clustered NOK Images. With yellow: images without disturbing factors, with purple: images with disturbing factors. On the samples in the lower density space, there are no connector at all.

After the clustering, we checked again the category manually, which contains disturbing factors. The results of pre-processing seen in Table 2.

Table 3. Ratio of images after pre-processing

<b>OK</b>	<b>NOK</b>	<b>Wrong pos.</b>	<b>Sum</b>
22582	81	92	22755

As seen, formerly there was 738 pictures categorized as NOK, despite they were truly OK. Wrong positions means that the combustion engine position were wrong, thus on the picture there was no connector at all. If we sum categories NOK and Wrong pos., we will have 173 true NOK pictures, 0.76 percent of total pictures. Theoretically, that means if we train an ANN with these dataset and reach about 98 percent accuracy, the network could categorize all NOK pictures as OK despite still have an accuracy of 98%. To prevent this problem we need more NOK samples for the training, and for that purpose, image augmentation [14] can be the solution.

### **3.5. Image augmentation**

To deal with the problem of having few training images, the data augmentation and image generation is one possible solution. With TensorFlow and Keras it is possible to randomly shift vertically and horizontally, rotate, flip, rescale, zoom-in or zoom-out, brighten or darken images. To do this we used the 173 NOK and 173 randomly selected OK picture as the input of the image generator. The image generator generated the input images of the model in the batch of 32 in every step of every epoch. In this case, we used 25 epochs and 25 steps per epoch, results in a total of 20000 images. With this method, the bias of the OK-NOK rate reducible and the training speed increasable. We used the 80% of the images to train the network, the rest 20% was the validation dataset.

### **3.6. Convolutional neural networks**

To detect and reduce the pseudo-errors we used a convolutional neural network [6]. Convolutional neural networks or CNNs are a special kind of ANNs inspired by biological processes [7]. CNNs are very effective in areas such as image recognition and classification. CNNs used to identifying faces, objects and traffic signs and used in robotic vision and autonomous cars. CNNs use relatively little pre-processing, the network learns the filters that in traditional machine vision algorithms were hard-coded, thus CNNs are independent from prior, human knowledge. These networks learns higher representation of images by learning the filters corresponding to the objects or parts of the image. The output neurons share weights as the same filter is applied but to different image regions, this approach reduces the number of weights used by the network.

The network consists of an input layer, an output layer, and multiple hidden layers. The hidden layers consist of convolutional layers, activation function, pooling layers, fully connected layers and normalization layers. For classification task, the output layer usually have softmax or sigmoid activation. A typical CNN architecture seen on Figure 6.

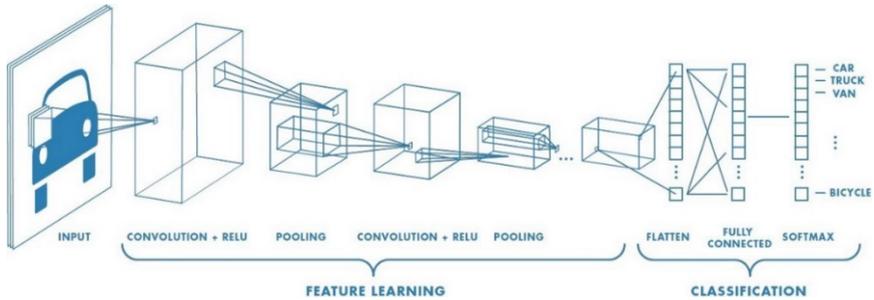


Figure 6. A CNN architecture. [8]

The convolutional layers apply filters to the original image, or to other feature maps in deeper networks. The most important parameters are the number of kernels (filters) and the size of the kernels.

Pooling layers performs the subsampling, a specific function such as max pooling or average pooling, which takes the maximum or the average value in a certain filter. These layers are reducing the dimensionality of the previous layers.

Fully connected layers used to flatten the results of the pooling layers before the activations propagated to the classification layer.

#### 4. The model

The architecture similar to the LeNet-4 models architecture, which is a relative simple, yet accurate model. [9] We fitted the model parameters to the input image parameters. The architecture of the model is the following: after the input layer comes two convolutional layer with ReLU (*rectified linear unit*) activation, the third layer is a MaxPooling layer followed by a Flatten layer, which connects the convolutional part of the network to the fully connected part of the network. The fully connected, also known as the dense part of the network contains three layers, and the output layer contains one node with softmax activation. For the better generalization, we used 20% dropout [10] between the fully connected layers. We tested many dropout rates, but in the literature widely used value, performed the best.

To train the model we used the Keras ImageDataGenerator class. This class responsible for the image augmentation also. We trained the model for 25 epochs, with 25 steps per epoch. In each step, we fed the generated images to the network in the batches of 32 picture. The optimizer was an Adam [11] optimizer, the loss

function is mean squared error and the metric for the optimization is the model accuracy.

For validation, we used another batch of pictures from a second generator. After 25 epochs of training, the validation accuracy was about 90% (Figure 7.).

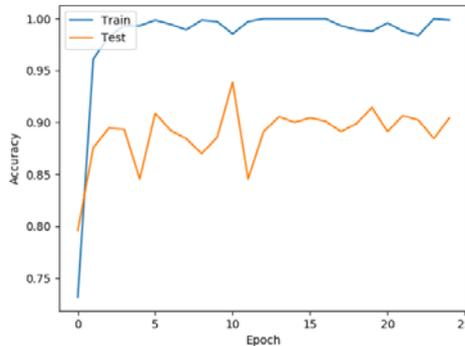


Figure 7. Model accuracy

## 5. Results

After the training phase, the 911 images, which formerly classified as NOK by the original machine vision algorithm, we fed to the model. The improvement of misclassification rate seen in Table 4.

Table 4: Improvement of misclassification rate

<b>Total Samples</b>	<b>True NOK</b>	<b>Misclassification</b>	
		<b>Original system</b>	<b>CNN</b>
911	173	738 (81.01%)	23 (2.52%)

## 6. Conclusion

The model is much simpler than the state-of-the-art (VGG [12], CapsNet [13]) image classification models, but for special tasks, it can provide promising results. In contrast with the artificial general intelligence, where the goal is to distinguish and classify as much objects as possible, the goal and the scope of the industrial image processing is well defined and restricted to few cases.

With the use of machine learning, the performance of the conventional industrial image processing systems effectively improvable. If we use the model as the extension of the original system, it reduces the pseudo-error (false positive) rate, thus, reducing the amount of human intervention, and the productivity increases.

The usage of ML methods in production can reduce the manufacturing cost and boost the effectiveness of production, thus besides financial-economic advantages higher levels of production quality can be reached. Researching further potential use-cases and the targeted development of industrial application are highly recommended.

## **7. Further goals**

The further goals of the research besides improving the existing model is to develop and test new and robust models with sophisticated categorisation capabilities. The goal is to reach up to 98% accuracy on categorising true (positive) and false (false positive) errors, and to categorize the true errors.

## **Acknowledgement**

AUDI HUNGARIA Zrt supported the publishing of this paper and the research.

## **References**

- [1] F. Pedregosa, G. Varoquaux, A. Gramfort et al, Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12 (2011) pp. 2825-2830.  
URL <https://dl.acm.org/citation.cfm?id=1953048.2078195>
- [2] M. Abadi, A. Agarwal, P. Barham et al., TensorFlow: Large-scale machine learning on heterogeneous systems, p. 19, 2016.  
arXiv: <https://arxiv.org/abs/1603.04467>
- [3] D. Arthur, S. Vassilvitskii, k-means++: The Advantages of Careful Seeding, SODA '07 Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, New Orleans, Louisiana, 2007, pp. 1027-1035.  
URL <https://theory.stanford.edu/~sergei/papers/kMeansPP-soda.pdf>
- [4] J. Shlens, A Tutorial on Principal Component Analysis, educational guide, Machine Learning, Cornell University, p. 12, 2014.  
arXiv: <https://arxiv.org/abs/1404.1100>

- [5] Y. Lecun, Y. Bengio, Convolutional Networks for Images, Speech, and Time-Series. in: *The Handbook of Brain Theory and Neural Networks*, MIT Press Cambridge, MA, USA, 1998, pp. 255-258.
- [6] K. O'Shea, R. Nash, An Introduction to Convolutional Neural Networks, ArXiv preprint, p. 10, 2015.  
arXiv: <https://arxiv.org/abs/1511.08458>
- [7] K. Fukushima, Neocognitron, *Scholarpedia* 2 (1): 1717, 2007.  
doi: <https://doi.org/10.4249/scholarpedia.1717>
- [8] S. Saha, A Comprehensive Guide to Convolutional Neural Networks—the ELI5 way [cited 2019-05-03]  
URL <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [9] Y. Le Cun, L. Bottou, C. Cortes et al., Comparison of learning algorithms for handwritten digit recognition, *International Conference on Artificial Neural Networks*, in: F. Fogelman, P. Gallinari (Eds.), EC2 Cie Publishers, Paris, 1995, pp. 53-60.
- [10] N. Srivastava G. Hinton, A. Krizhevsky et al., Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15 (2014) pp. 1929-1958.
- [11] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, *International Conference on Learning Representations*, arXiv preprint, p. 15, 2014.  
arXiv: <https://arxiv.org/abs/1412.6980>
- [12] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv preprint, p. 14, 2014.  
arXiv: <https://arxiv.org/abs/1409.1556>
- [13] S. Sabour, N. Frosst, G. E. Hinton, Dynamic Routing Between Capsules, arXiv preprint, p. 11, 2017.  
arXiv: <https://arxiv.org/abs/1710.09829>

- [14] L. Perez, J. Wang, The Effectiveness of Data Augmentation in Image Classification using Deep Learning, arXiv preprint, p. 8, 2017.  
arXiv: <https://arxiv.org/abs/1712.04621>
  
- [15] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, B. Yu, Recent Advances in Convolutional Neural Network Acceleration, *Neurocomputing* 323 (2019) pp. 37-51.  
doi <https://doi.org/10.1016/j.neucom.2018.09.038>