Automation of Quality Control in the Automotive Industry using Deep Learning Algorithms

Charbel El Hachem
FEMTO-ST Institute, CNRS
Univ. Bourgogne Franche-Comté (UBFC)
Belfort, France
charbel.el_hachem@univ-fcomte.fr

Gilles Perrot
FEMTO-ST Institute, CNRS
Univ. Bourgogne Franche-Comté (UBFC)
Belfort, France
gilles.perrot@univ-fcomte.fr

Loïc Painvin
R&D center
Faurecia Clean Mobility
Bavans, France
loic.painvin@faurecia.com

Raphaël Couturier
FEMTO-ST Institute, CNRS
Univ. Bourgogne Franche-Comté (UBFC)
Belfort, France
raphael.couturier@univ-fcomte.fr

***Abstract*—Quality control is an essential operation for an automotive company like Faurecia. A vast number of references is produced, and many regions of interest need to be checked. For that, quality control is necessary and should be applied to every reference part. Visual inspection is achieved by the operator who checks each part manually. After several checks per day, the operator gets tired and thus may misqualify a welding seam or a component control. To avoid that, Faurecia is trying to integrate automatic quality control to obtain better overall equipment effectiveness (OEE), especially to avoid performance degradation over the operator's shift. Researches demonstrate the ability of a neural network to reach high precision in detecting object presence or absence. We have been able to achieve an accuracy of 99% with ResNet-50. Apart from accuracy, the other performance matrices used in this work are reliability and cycle time. Our contribution will help the current state of manufacturing by offering an automatic visual inspection, which will lead to other innovative projects in the automotive industry.**

***Index Terms*—Deep Learning , Image Classification , ResNet-50 , Convolutional Neural Network , Industry Automation.**

I. Introduction

Faurecia is a company that develops and produces automotive seating, interiors, and emissions control technologies. Ensuring control guarantees production quality that meets the requirements of the client. Thereby, quality control is an essential operation for an automotive company. A vast number of references is produced, and dimensions of the same part may have dissimilarities. For that, quality control is necessary and should be applied to every reference part. Several technologies are used such as: visual & geometric control, leak test, and many other functional tests. Today, these technologies are expensive from an investment point of view, tools, and human resources. Visual control is done by the operator who checks each part manually, making the reliability highly improvable. At first, the automation of the inspection can be carried out by different methods: camera, thermography, laser scanning, and in conjunction with artificial intelligence techniques [1], [2]. Then, it will be necessary to correlate these quality results with different parameters (processing and environmental) to reduce non-conformities.

The main goal is to provide automatic control of the manufactured products and overcome the need for human intervention. Will the automation of the visual inspection improve the reliability in detecting defective parts at Faurecia? Will it help reducing cycle time in the production line? Thereby, this project needs to detect the presence or the absence of one component in a vehicle's part. This quality control needs to be done primarily to assembling the exterior part of the car because once these two parts are joined, the component is not accessible anymore.

In order to reach high precision in detecting presence or absence, we need an adequate feature extractor that would transform raw data collected from the plant into a suitable feature vector.

This paper is organized as follows. Section II introduces the problem statement at Faurecia. Section III presents an overview of the existing work done in the area of image classification. In Section IV, the proposed supervised method for image classification is explained. The implementation details are contained in Section V, while in Section VI, the analysis of the experimental results is presented. The last section includes our concluding remarks.

II. Problem Statement

A. Industry automation

The process of automated resources making decisions without any human involvement is what we call automated

Supported by Faurecia

decision-making. One of the crucial reasons for automating an industrial operation is worker safety. Heavy machinery, running at hot temperatures, and sharp objects increase the hazards of the factory environment. Another benefit of automation is freedom from fatigue. Machines can produce the same way the same part all day long without losing quality over time. Industrial automation does impact other parameters such as security, data integrity, interoperability, scalability, reliability, availability, and many others. Companies took a giant stride forward on financial and technological aspects when investing in automation of manufacturing.

In agriculture [3], wood industry [4], robotic process [5], and many other fields, automation has proved its importance in accelerating decision making. Automotive companies are also working on this automation and add a particular focus on quality automation to secure their processes. Instead of using the operator for checking the reference part manually, an automatic system needs to be launched in order to test its feasibility and efficiency.

B. Reference Part

Figure 1 is an example of a sub-assembly that takes place in the vehicle. The presence or absence of multiple screws needs to be checked. It contains 10 regions of interest. In the figure, the regions of interest are presented with their right screws.

The component control should be done while taking into consideration the different lighting conditions of the plant.

Fig. 1. Sub-assembly part to be checked

As described in figure 2, three types of screws can be used for this component control: *flexitol*, *rivstud* and *rivnut*. The same type of screw may be the right one for a specific region of interest and the wrong screw for another region. Three possibilities can show up:

1) the right screw is present in the right region of interest.
2) the screw is missing.
3) there is a screw, but not the one we expected. This case is considered as NOT OK (NOK).

Fig. 2. Types of possible screws

For the first possibility, the part is OK. It can continue its cycle on the production line. For the second and third possibilities, the part is NOK. It should be sent to rework. The distribution of these screws is presented in Table I.

TABLE I
SCREWS DISTRIBUTION THROUGHOUT THE 10 REGIONS

Region	Right screw
1	Rivstud
2	Rivstud
3	Rivnut
4	Rivnut
5	Rivnut
6	Flexitol
7	Rivnut
8	Flexitol
9	Rivnut
10	Flexitol

III. RELATED WORKS

A. CNN Applications

Traditional image processing techniques such as Structural Similarity Index (SSIM) or Mean Squared Error (MSE) may detect the difference between two images. The problem with these methods is their robustness when changing lighting conditions.

Whereas, machine learning models, and specifically deep learning models have achieved promising results. Used in the classification of images to prevent any bad decision making, deep learning's key aspect is that these feature layers are not designed by human engineers. One of the most popular deep learning architectures is the convolutional neural network (CNN). It has proved to be very efficient in extracting deep features [1], [6], [7]. The architecture of CNN based on a succession of layers, reduces images without losing the important features, which leads to getting good prediction. Each layer transforms the input volume to an output volume. CNN performs well on a series of visual applications and tasks, such as image classification [8]–[10], image detection [11]–[13] and image denoising [14]–[16]. Detection of the presence or absence of multiple screws is an image classification task that can be solved using CNNs. Many deep learning models can be used such as DenseNet, EfficientNet, and many others. We chose ResNet-50 for it has shown promising results when applied in an industrial context, that's why we implemented such a feature extractor using a deep network.

B. ResNet Applications

Mangalam et al. [17] have invested in ResNet-50 for bird call recognition. Spectrograms (visual features) extracted from the bird calls were used as input and were able to achieve 60 to 72% of birds call recognition.
Rezende et al. [18] applied ResNet-50 to classify malicious software. Malware samples were represented as byteplot grayscale images as the input layer. The experimental results reached an accuracy of 98.62% for classifying malware families.
While Theckedath et al. [19] investigated in 3 architectures: VGG16, ResNet-50 and SE-ResNet50. This method reuses weights of already developed models to train a CNN and detects 7 basic affect states. The evaluation shows that ResNet-50 outperforms other networks reaching a validation accuracy of 99.47%.
Wen et al. [2] is an example where ResNet-50 is applied in the manufacturing domain. The dataset includes bearing damage, motor bearing, and self-priming centrifugal pump dataset. By adding one layer in order to transfer time-domain fault signals to RGB images format as input for ResNet-50, the prediction accuracies reach around 99%.
Our approach for sub-assembling family classification is assumed from a CNN model based on the Residual Network architecture with 50 layers (ResNet-50) [20], [21].

IV. Methods

ResNet-50 has 48 convolution layers along with one Average Pool layer and one Fully Connected layer. ResNet-50 uses shortcut connections which ensures that the higher layer will perform at least as good as the lower layer, and not worse. As input, we have the image tensor. Each three convolution layers forms a block called bottleneck block. The input of each block is added to the next block. This is what we call a shortcut connection (or skip connection). Figure 3 illustrates how layers communicate with each other.
The dotted lines mean that we need to change the dimension of the input; for example, if the number of channels of the previous layer is equal to 256 and the next one is equal to 512, we need to change the number of channels of the previous basic block in order to be equal to the next block. In this case, it should be multiplied by 2. Finally, we apply the average pooling and perform fully connected layers. Average pooling means downsampling the feature map (output of one filter applied to the previous layer) by calculating the average of each patch of the feature map (2x2 square). Then, the fully connected layer flattens the output of the average pooling, going through its backpropagation process in order to define the most accurate weights.
Let's consider each and every block in detail: each block takes an input to perform convolution, produces an output, and then activates the output. It will repeat this same procedure for the second and the third block. On the other hand, for the second and third block, to activate the output, we add the same input, and then we activate the output. So, instead of adding convolution layers to improve accuracy, ResNet adds, skip connections.

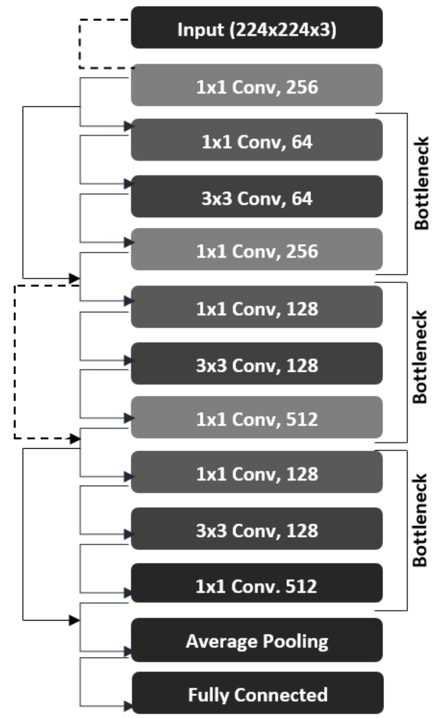

Fig. 3. Skip connection and bottleneck in ResNet-50 Architecture

The architecture of the bottleneck block is as follows:

1) a 1x1 convolution which will change the channels.
2) the channel is constant but height and width change because the filter size is different than 1.
3) the number of channels is expanded by multiplying it by 4.

V. Implementation details

A. Data Collection

All images have been collected in the plant during multiple shifts to cover different lighting conditions. The camera used is an 18MP camera, positioned at the top of the reference part with a distance of 500mm. Each global image is cropped in 10 small images in order to cover all 10 regions of interest. The coordinates of each region of interest were defined prior to cropping, guaranteeing that the crop covers the 3 different type of screws. Achieving 1920 global images, the data set contains 19200 cropped images. 70% of the images collected were randomly selected for the model training, 20% were used for model validation, and the remaining 10% were used to test the efficiency of the model.

B. Analysis

Based on the collected data, some regions should have the same type of component and look the same for the camera.

As in Table I, region 1 and region 2 should have a rivstud as their right component. These 2 regions look the same for the camera as in Figure 1.
Other regions have the same right component (as region 3 and region 7 in Table I), but look totally different for the camera as in Figure 1. This helps in regrouping some regions which will allow in reassembling the most possible number of images in a specific class. These common areas are grouped in 4 zones:

1) **Zone 1:** contains regions 1 and 2.
2) **Zone 2:** contains regions 3, 4 and 5.
3) **Zone 3:** contains regions 6, 8 and 10.
4) **Zone 4:** contains regions 7 and 9.

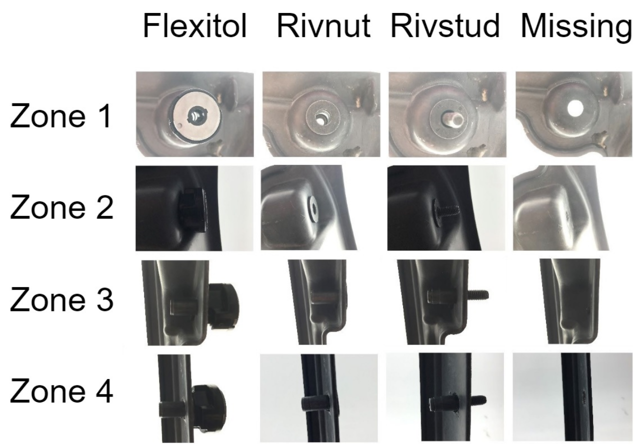

Fig. 4. Sample of each class in the trained model

As in Figure 4, each zone is divided into 4 different classes according to the different possible cases. For example, images from Zone 1 are split as follow: *zone1-flexitol*, *zone1-rivstud*, *zone1-rivnut*, *zone1-missing*. The class *'zone1-missing'* represents the images where none of the components is present in Zone 1.

C. Experimental Environment

The experimental environment is powered by Intel i5 CPU with 64-bit Windows 10 system, a memory of 8 GB, and 2.30 GHz basic frequency. The software programming environment is python. It is based on Keras framework, having Tensorflow and Theano as backend.
In the image classification model, Adam is selected as an optimizer of the CNN. The learning rate is set to 10^{-5}. Concerning the ResNet-50 weights, a pre-trained version of the network trained with the ImageNet database has been loaded. The input of the CNN is a 4D array. It has a shape of (batch size, height, width, depth). The batch size (equal to 32 in our case) defines the number of samples that will be propagated through the network. The other parameters represent the height, width, and depth of the image. In our case, the input has the following values (32, 224, 224, 3). The model has been trained with 100 epochs. Each epoch passes the entire data set through the neural network, only once.

TABLE II
ACCURACY'S COMPARISON BETWEEN HUMAN-BEING CHECK AND RESNET-50

Component type	Human-being check(%)	ResNet50-offline(%)
flexitol	98	99.999
rivstud	97	99.998
rivnut	98	99.999
missing	93	99.999

VI. Experimental Results

A. Accuracy Improvements

As in Table II, the accuracy reaches 99% with deep learning. The software is able to detect not only the presence or absence of the component but also the presence of a screw of the wrong type.

As in Table II, comparing the results between the human-being check and the autonomous control of ResNet, we realize an upgrade of the precision from 96.5% to 99% respectively. These results show that replacing the human-being check by an automatic visual inspection system provide better accuracy.

B. Cycle Time and Reliability Improvements

As for the cycle time, we chose randomly 10 operators (having 5 to 25 years of experience) and we gave them an OK part (containing all 10 regions with their right screws). It is easier for the operator to make its visual inspection than an NOK part. Moreover, we launched this same part with our neural network. The test was repeated 10 times. As in Table III, we realize that the average of the time spent by an operator to check this part varies between 6, 7 and 8 seconds, while the average of the neural network is around 5 seconds. The neural network provides a stable performance, it won't be affected during the day. Its architecture needs 0.4 seconds per region and a total of 1 second to crop all regions.

TABLE III
CYCLE TIME'S COMPARISON BETWEEN HUMAN-BEING CHECK AND RESNET50

Time in seconds	Test1	Test2	Test3	Test4	Test5	Average
Operator1	9.24	8.40	8.67	9.45	8.22	8.796
Operator2	9.85	8.63	8.19	9.03	8.15	8.77
Operator3	8.40	7.29	7.01	7.46	7.58	7.548
Operator4	7.56	7.77	7.65	6.54	7.18	7.34
Operator5	8.74	9.52	7.20	7.33	8.42	8.242
Operator6	6.48	5.88	6.39	5.91	6.02	6.136
Operator7	8.85	9.32	8.66	9.27	8.33	8.886
Operator8	8.58	9.08	9.25	8.85	8.19	8.79
Operator9	8.72	7.48	7.56	7.37	8.11	7.848
Operator10	7.36	7.82	8.66	7.44	8.23	7.902
ResNet-50	5.21	5.42	5.08	5.28	5.39	5.276

To evaluate a NOK part, the operator will have for each region of interest 4 different possibilities that he should take into consideration. While we estimate that an operator will need more time to evaluate a NOK part, we are sure that the ResNet-50 will keep its 5 seconds. Cycle time is then improved with the automation of the visual inspection.

In addition to these two parameters, we must take into account the overall equipment effectiveness (OEE). Its calculation is based on three factors:

1) **Availability:** it concerns the events that stopped planned production long enough, and needs to track a reason for being down.
2) **Performance:** it concerns anything that causes the industrial process to run at less than the maximum possible speed when it is launched.
3) **Quality:** it considers Good Parts as parts that successfully pass through the manufacturing process the first time without needing any rework. By that, it takes into account all manufactured parts that need rework. It is calculated as the ratio of Good Count to Total Count.

With our proposed model, OEE is clearly enhanced. Firstly in terms of performance: reducing the number of operators for the visual inspection increases the reliability. This is due to the fact that the algorithm will avoid performance degradation over the operator's shift. On the other hand, better quality is reached: by detecting defective parts at an early stage before being manufactured. This increases the percentage of good parts passing through the manufacturing process for the first time without needing any rework.
Controlling presence or absence in automotive industry is not needed for screws only, many topics need quality control such as welding, spatters, holes, and many others. The proposed solution can be tested on these different topics and may be applied safely in plants.

VII. Conclusion

This paper proposes a method based on deep learning to control component presence. The features of the dataset are extracted by ResNet-50, which fully achieves the ability to extract images' features. The results show that the maximum accuracy of the proposed method is 99%, which can meet the requirements for visual inspection in the production line. The images collected as data set for classification have been all carried out offline. After the required accuracy is reached, new images were taken in the actual industrial production. The model tested on these images reached 99%. Thereby, the effectiveness and practicability of the proposed model are verified by the plant. Results show that the automation of quality control in the automotive industry does improve reliability, accuracy, and cycle time.
In our future work, we plan to detect object movement and being able to readjust the camera's angle in order to have the most flexible and robust system. In other words, a self-adjusting system regardless of environmental disturbances.

Acknowledgements

This work was done as a part of a CIFRE (N 2018/1029) project with Faurecia, funded by the Ministry of Higher Education and Research of France, managed by the Association Nationale de la Recherche et de la Technologie (ANRT) and was partially supported by the EIPHI Graduate School (contract "ANR-17-EURE-0002"). The first author would like to thank Faurecia plant's staff for their availability.

References

[1] Z. Gao, Y. Zhang, and Y. Li, "Extracting features from infrared images using convolutional neural networks and transfer learning," *Infrared Physics & Technology*, vol. 105, p. 103237, 2020.

[2] L. Wen, X. Li, and L. Gao, "A transfer convolutional neural network for fault diagnosis based on resnet-50," *Neural Computing and Applications*, pp. 1–14, 2019.

[3] E. Guzueva, T. Vezirov, D. Beybalaeva, A. Batukaev, and K. G. Chaplaev, "The impact of automation of agriculture on the digital economy," in *IOP Conference Series: Earth and Environmental Science*, vol. 421. IOP Publishing, 2020, p. 022047.

[4] R. Salim, A. Manduchi, and A. Johansson, "Investment decisions on automation of manufacturing in the wood products industry: A case study," *BioProducts Business*, pp. 1–12, 2020.

[5] A. M. Radke, M. T. Dang, and A. Tan, "Using robotic process automation (rpa) to enhance item master data maintenance process," *LogForum*, vol. 16, no. 1, 2020.

[6] Z. Gong, P. Zhong, Y. Yu, W. Hu, and S. Li, "A cnn with multiscale convolution and diversified metric for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 6, pp. 3599–3618, 2019.

[7] T. J. Brinker, A. Hekler, A. H. Enk, J. Klode, A. Hauschild, C. Berking, B. Schilling, S. Haferkamp, D. Schadendorf, T. Holland-Letz *et al.*, "Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task," *European Journal of Cancer*, vol. 113, pp. 47–54, 2019.

[8] C. Yu, R. Han, M. Song, C. Liu, and C.-I. Chang, "A simplified 2d-3d cnn architecture for hyperspectral image classification based on spatial-spectral fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2020.

[9] M. Gour, S. Jain, and T. Sunil Kumar, "Residual learning based cnn for breast cancer histopathological image classification," *International Journal of Imaging Systems and Technology*, 2020.

[10] T.-H. Hsieh and J.-F. Kiang, "Comparison of cnn algorithms on hyperspectral image classification in agricultural lands," *Sensors*, vol. 20, no. 6, p. 1734, 2020.

[11] M. I. G. Ocaña, K. L.-L. Román, N. L. Urzelai, M. Á. G. Ballester, and I. M. Oliver, "Medical image detection using deep learning," in *Deep Learning in Healthcare*. Springer, 2020, pp. 3–16.

[12] F. Sultana, A. Sufian, and P. Dutta, "A review of object detection models based on convolutional neural network," in *Intelligent Computing: Image Processing Based Applications*. Springer, 2020, pp. 1–16.

[13] M. Akil, Y. Elloumi, and R. Kachouri, "Detection of retinal abnormalities in fundus image using cnn deep learning networks," 2020.

[14] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu, "Attention-guided cnn for image denoising," *Neural Networks*, vol. 124, pp. 117–129, 2020.

[15] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep cnn with batch renormalization," *Neural Networks*, vol. 121, pp. 461–473, 2020.

[16] S. Ghose, N. Singh, and P. Singh, "Image denoising using deep learning: Convolutional neural network," in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2020, pp. 511–517.

[17] M. Sankupellay and D. Konovalov, "Bird call recognition using deep convolutional neural network resnet-50," in *Proc. ACOUSTICS*, vol. 7, 2018, pp. 1–8.

[18] E. Rezende, G. Ruppert, T. Carvalho, F. Ramos, and P. De Geus, "Malicious software classification using transfer learning of resnet-50 deep neural network," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 1011–1014.

[19] D. Theckedath and R. Sedamkar, "Detecting affect states using vgg16, resnet50 and se-resnet50 networks," *SN Computer Science*, vol. 1, no. 2, pp. 1–7, 2020.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.