

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/338941941>

Machine Learning Applied to an Intelligent and Adaptive Robotic Inspection Station

Conference Paper · July 2019

DOI: 10.1109/INDIN41052.2019.8972298

CITATIONS

6

READS

2,066

4 authors, including:



[Luis Sousa Variz](#)

Instituto Politécnico de Bragança

1 PUBLICATION 6 CITATIONS

SEE PROFILE



[Luis Piardi](#)

Federal University of Technology - Paraná/Brazil (UTFPR)

35 PUBLICATIONS 151 CITATIONS

SEE PROFILE



[Pedro Rodrigues](#)

Instituto Politécnico de Bragança

13 PUBLICATIONS 48 CITATIONS

SEE PROFILE

Machine Learning Applied to an Intelligent and Adaptive Robotic Inspection Station

Luis Variz*, Luis Piardi*, Pedro João Rodrigues*, Paulo Leitão*

* Research Center in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança,
Campus de Santa Apolónia, 5300-253 Bragança, Portugal
Email: luis.sp.variz@alunos.ipb.pt, {piardi, pjsr, pleitao}@ipb.pt

Abstract—Industry 4.0 promotes the use of emergent technologies, such as Internet of Things (IoT), Big Data, artificial intelligence (AI) and cloud computing, sustained by cyber-physical systems to reach smart factories. The idea is to decentralize the production systems and allow to reach monitoring, adaptation and optimization to be made in real time, based on the large amount of data available at shop floor that feed the use of machine learning techniques. This technological revolution will bring significant productivity gains, resources savings and reduced maintenance costs, as machines will have information to operate more efficiently, adaptable and following demand fluctuations. This paper discusses the application of supervised Machine Learning techniques allied with artificial vision, to implement an intelligent, collaborative and adaptive robotic inspection station, which carries out the quality control of Human Machine Interface (HMI) consoles, equipped with pressure buttons and LCD displays. Machine learning techniques were applied for the recognition of the operator's face, to classify the type of HMI console to be inspected, to classify the state condition of the pressure buttons and detect anomalies in the LCD displays. The developed solution reaches promising results, with almost 100% accuracy in the correct classification of the consoles and anomalies in the pressure buttons, and also high values in the detection of defects in the LCD displays.

Index Terms—Machine learning, Artificial vision, Quality control, TensorFlow, Convolution neural network

I. INTRODUCTION

At the end of the XX century, industries began to automate their processes with the help of computing and automation [1]. Similarly, the quality control (QC) process has been modernized and automated, although in many cases it is still made by humans, which brings several problems, namely mistakes associated with the repetitive execution of tasks, resulting in high costs for the company. Recently, the impressive progresses in Artificial Intelligence (AI), driven by exponential increases in the computing power and by the availability of vast amounts of data, enabled the development of automatic and intelligent QC solutions.

Machine Learning (ML) is a specific strand of AI that trains machines to learn with data that consists in the execution of algorithms that automatically create knowledge representation models based on a database. By letting the algorithm "learn", to iterative adjust the knowledge representation model, it is possible to improve its performance. After this training phase, the model has a potential to make predictions, quality diagnoses in future situations, related to historical patterns with

minimal human intervention, optimizing processes, increasing the reliability and generating savings.

Current QC solutions also use image processing techniques allied with ML algorithms to perform more efficient and faster inspection tasks. The application of automated visual control systems has been able to improve the industry's productivity [2], establishing reliable criteria to control the quality of products and services with reduced costs. Applying ML to this kind of systems, it is possible to make systems more flexible and adaptive, able to learn in an evolutionary way, making systems more capable of achieving higher and higher efficiency levels.

Having this in mind, this paper describes the application of ML techniques to develop an intelligent and adaptive robotic inspection station, capable to perform QC tasks in HMI consoles of different types and configurations without any human intervention. The application of ML algorithms was performed for the tasks related to the operator recognition, classification of the type of HMI under inspection, detection of errors in the LCD display and classification of the push buttons condition. This approach provides a more reliable, flexible, adaptable and robust solution by reducing setup times, in case of condition changes.

The rest of this paper is organized as follows. Section II overviews the state of the art related to industrial quality control using artificial vision and particularly ML algorithms and section III presents the intelligent and adaptive robotic inspection station. Section IV presents the application of ML algorithms to several situations in this robotic inspection station, namely the operator's face recognition, the HMI consoles classification, the LCD display errors detection and the pressure buttons condition classification. Finally, section V rounds up the paper with the conclusions and points out the future work.

II. STATE-OF-THE-ART

Industries are using multiple information and communication technologies (ICT) to perform and automate several production tasks, such as scheduling and planning, process control, tracking and QC. QC is the task of ensuring that products reach a certain standard, either defined by the company or by customers. This field was developed rapidly during the second half of the twentieth century and it is now an

integral part of most manufacturing companies, being currently performed not only at the final stage with conventional techniques, namely statistical quality control and acceptance sampling, but instead along the production process [3], allowing to detect earlier quality deviations, resulting in higher quality standards and lower production cost. However, with the current large amounts of data available at shop floor, there is unexplored potential in the manufacturing industry to develop more automatized and intelligent QC tasks by applying AI and particularly ML techniques. The use of ML algorithms in inspection processes allows to identify earlier defects or deviation as well as perform diagnosis of possible problems.

One of the most dependent industries of the quality management is the food industry, where the lack of quality of at least one ingredient can directly impact the quality of the final product [4]. Kewpie Corporation (a major Japanese food company) is using Google's TensorFlow machine learning libraries in their visual inspection system, to automatically detect anomalies in their diced potatoes [5]. Fujitsu has also developed a solution to identify potential defects in the manufacturing process by performing non-Destructive Testing (NDT) inspection that is combined with image processing and Deep Learning techniques to realize a diagnosis in minutes [6]. Siemens is also using similar AI solutions to identify faults in glass slides up to 75 meters, sweeping each centimeter to identify any type of flaw that could cause manufacturing defects. The inspection process takes now 1 hour and half, when previously it took 6 hours to be completed [7].

These examples show how image processing and AI can be useful applied to improve the QC process. The advent of industry 4.0, and the availability of large amounts of data, allows to use AI to support the development of more adaptive and efficient inspection stations.

III. INTELLIGENT ROBOTIC INSPECTION STATION

In this work an intelligent and adaptive inspection station was developed to perform QC of HMI consoles. For this propose, as illustrated in Figure 1, this inspection station, is composed by an UR3 collaborative robot from Universal Robots, equipped with a force-torque sensor FT 300 of six degrees of freedom (DoF), and an high-resolution industrial camera Mako G-125b for the image acquisition.

A second camera, a Logitech with 720p of resolution, carries out the acquisition of the image to support the operator's face recognition by the system. As human interface, the inspection station is equipped with a Samsung Galaxy Tab, where the operator provides the instructions for operation and visualizes the results of the inspection task. Dealing with a collaborative area where the operator and the robot shares the same work space, e.g., to change the console that will be inspected, some security equipment are installed to improve its safety, namely an industrial barrier sensor, a status lights signal and an emergency stop button.

These hardware apparatus is complemented, and managed, by a software app, codified in Python and running in an industrial PC, which uses ML techniques to perform several tasks

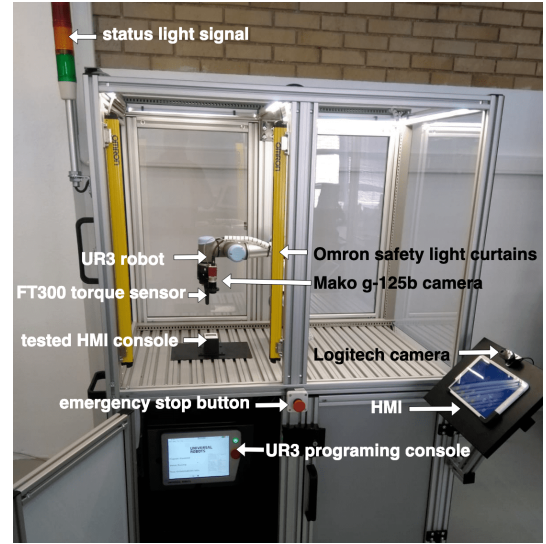


Figure 1. Hardware setup for the robotized inspection station.

during the inspection process, namely the operator recognition, the identification of the HMI console type, the classification of the buttons condition and the detection of errors in the LCD display. This computational app provides flexibility and adaptability during the inspection tests since automatically can perform tests to consoles with different configurations, e.g., different number of buttons and different layout of the LCD display, allowing to reduce the setup time and improve the efficiency and productivity of the inspection process.

The developed app to perform the inspection testing is running continuously in cycle, following the fluxogram illustrated in Figure 2.

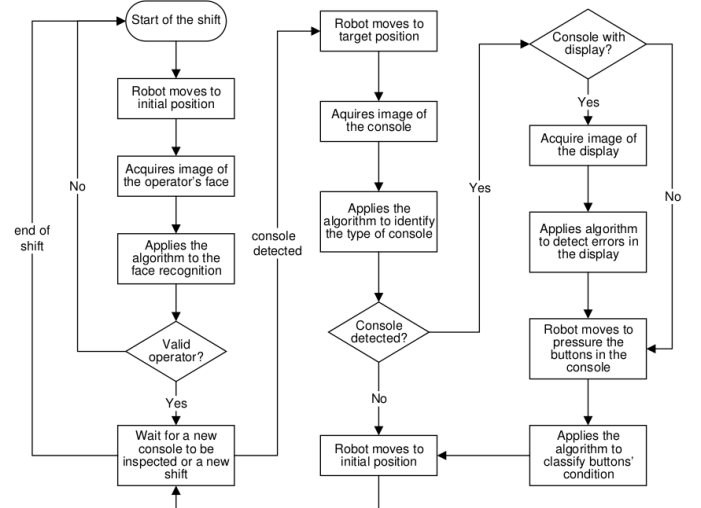


Figure 2. Fluxogram for the execution of an inspection test.

In the beginning of the shift, a face recognition algorithm, using the image acquired by the Logitech camera, identifies the operator in charge by the inspection station. At this stage,

the inspection station is ready to start inspecting the HMI consoles. After detecting a new console in position to be inspected, the robot moves to the console position and the image of the console is acquired to be processed by an ML algorithm that will identify the type of console. This information is used to decide the next inspection actions according to the console configuration. In case of existence of buttons, the robot moves to touch the torque-force sensor in the buttons, being the acquired data feeding the ML technique that classifies the buttons condition. The image acquired by the Mako g-125b camera is used by a ML algorithm, associated to an image processing technique, to detect errors in the LCD display.

During the inspection process, the work space is shared by the robot who moves along different positions to realize the different tasks and the operator who performs the exchange of HMI consoles. This collaborative work is safeguarded by the installed safety systems, namely the robot slows down the movement speed if the security barriers are triggered, and stops its movement if the emergency button is pressed by the operator or the robot safety mechanism is triggered.

New HMI consoles can easily and on-the-fly be added to the inspection catalogue provided by the inspection station, without the need of major changes in the inspection app, only requiring a retraining the ML algorithms.

IV. APPLICATION OF MACHINE LEARNING TECHNIQUES

This section describes the application of image processing and ML algorithms for the automatic and intelligent inspection of HMI consoles.

A. Recognition of the Operator's Face

The automatic recognition of the operator that is associated to the inspection station is performed by applying ML techniques to the acquired image. The applied procedure comprises the face detection and the face recognition.

1) *Face Detection*: The face detection is performed by using the Histogram of Oriented Gradients (HOG) algorithm [8], which is provided in the dlib toolkit [9]. This algorithm looks at every single pixel in the image, one at a time, and for every single pixel searches for the surrounding pixels (see Figure 3). The goal is to figure out how dark the current pixel is compared to the surrounding pixels.



Figure 3. Gradient example obtained from a single pixel.

An arrow is drawn to show in which direction the image is getting darker. If this process is repeated for every single pixel in the image, every pixel will be represented by its gradient direction. Thus, they show the flow from the light to dark across the entire image. But saving the gradient for every single pixel gives too much detail, being necessary to

break up the image into small squares of 16x16 pixels each. In each square, counts up how many gradients point in each major direction then replaces that square in the image with the arrow directions that were the strongest. The end result is a very simple representation that captures the basic structure of a face, as illustrate in Figure 4.

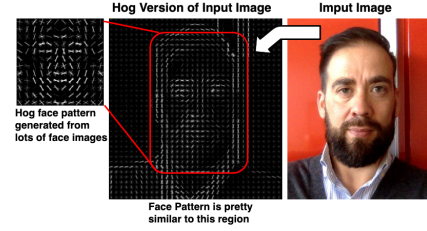


Figure 4. Face detection using the HOG algorithm.

In this work, a previous trained linear support vector machine (SVM) classifier was used to find the part of our image that looks the most similar to a known hog pattern, by sliding a window to identify and locate faces in an image [8].

2) *Face Recognition*: The face recognition is performed by using the deep metric learning technique, also provided by the dlib library [9], which provides as output the face characteristics. This technique is based on a Deep Convolution Neural Network to learn distance features to assimilate different face samples [10], [11]. For the dlib facial recognition network, the output feature vector is a list of 128 real-valued numbers, that is used to quantify the face previously detected. The training process works by looking at 3 face images at a time, loading two images of a known person and a picture of a totally different person [12], as illustrated in Figure 5. Repeating this step enough number of times, the neural network learns to generate 128 features that describe the face reliably.

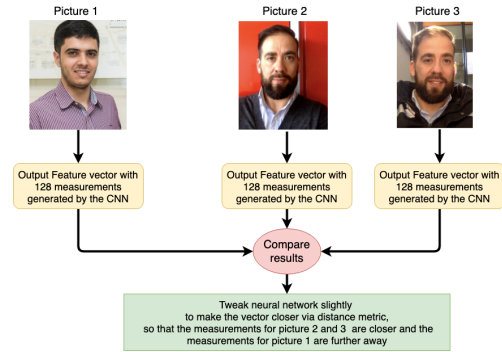


Figure 5. Single triplet training step

The network architecture for the face recognition is based on the ResNet-34 network [13], with a few layers removed and the number of filters per layer reduced by half. In this application, it is used a pre-trained network, which has an accuracy of 99.4% on the standard Labeled Faces in the Wild benchmark [14]. To create our dataset of features, each face is mapped by the pre-trained neural network to the corresponding

feature vector. During the classification, it was used a simple k-Nearest Neighbours (kNN) algorithm to find the person in the database which has the closest feature vector to the probe face (mapped to its feature vector). As illustrated in Figure 6, the confidence is around 68%, which combined with the number of necessary matches (>10 to avoid a false recognition) for each operator, increases the face recognition reliability and safety to use in a real industrial environment.



Figure 6. Results of the face recognition procedure.

B. Classification of the HMI Consoles

The classification of the HMI consoles, to detect different typologies, as illustrated in Figure 7. Since building an image classifier from scratch is a colossal and daunting task, the TensorFlow library was used in this work. TensorFlow is an open-source library created by Google that specializes in ML applications [15].

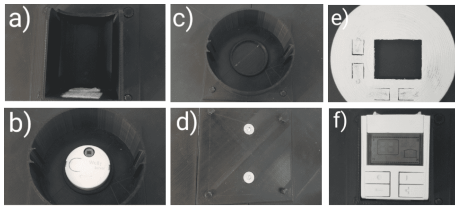


Figure 7. Examples of HMI console types: a) HMI with LCD display stand, b) one button HMI, c) one button HMI stand, d) empty object, e) 4 button round HMI and f) HMI with LCD display.

Inception is a deep convolution neural network trained for the image classification, which was trained on a staggering millions of images from the ImageNet database with thousand different categories [16]. Note that building a custom deep learning model demands extensive computation resources and a significant amount of training data. In this way, the proposed approach of using a previous trained network, brings several advantages, e.g., saving time, some reuse of the parameters that CNN has already learned allows to have a very precise classifier with much less training data. Reusing the previous trained models on different but related tasks is known as Transfer Learning, which reuse the initial and middle layers of the previous trained model to re-train only the final layers, as illustrated in Figure 8.

In a feedforward neural network, the neurons are organized in layers, with different layers performing different kinds of transformations on their inputs. The signals travel from the

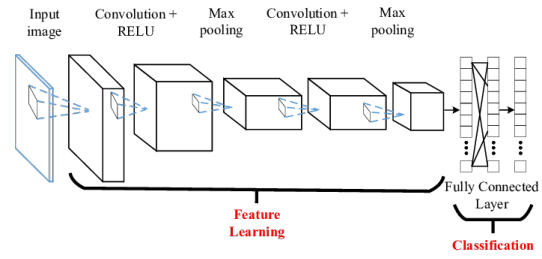


Figure 8. Deep CNN for classification, Figure adapted from [17].

first layer (input), to the last one (output), possibly after traversing the layers multiple times. As the last hidden layer, the “bottleneck” has enough summarized information obtained from the previous layers, to provide the next layer which does the actual classification task. The reason why final layer retraining can work on new classes is that it turns out the kind of information needed to distinguish between all the 1000 classes in the ImageNet databases often also useful to distinguish between new kinds of objects.

The training task for the image classifier is performed by using a database of 100 images of each type of HMI console, obtained by applying the dataset augmentation technique to a single image of each object using image processing algorithms [18]. The data augmentation is an automatic way to boost the number of different images in the database to train the Deep learning algorithms. The transformation functions for the data augmentation are the random rotation, the random noise, the horizontal flip and the blur, as shown in Figure 9.

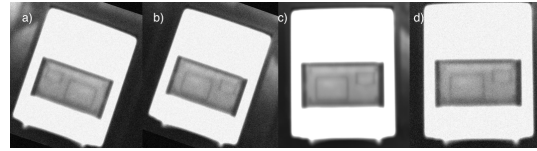


Figure 9. Example of the data augmentation image transformation: a) rotate, b) flip, c) blur and d) noise.

The achieved experimental results shown an accuracy higher than 97% in the correctness of the identification of the different typologies showed in Figure 7.

C. Detection of Defects in the LCD Display

The detection of defects in the LCD displays is performed by applying the template matching technique from the OpenCv libraries to the acquired image. This technique tries to find areas in a image that are similar to a pre-defined template, as illustrated in Figure 10.

To increase the reliability of the defects detection system, the image processing techniques have considered exposure control (average of gray pixel information from image histogram to approximate the similarity of both images) and Gaussian adaptive binarization to simplify the image, by eliminating the noise and highlighting the objects to be detected. A database was created with all the objects to be detected in the display, as represented in Figure 11.

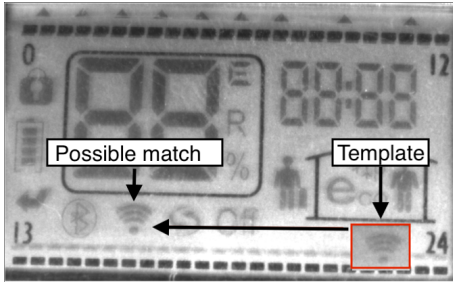


Figure 10. Example of applying the template matching technique to the LCD display.

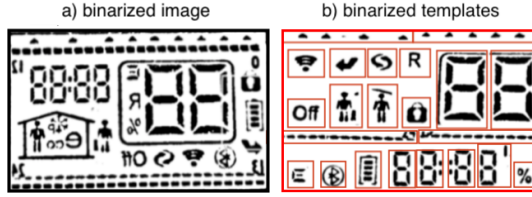


Figure 11. a) Binarized input image and b) 22 templates for matching.

The equation 1 represents the "CV_TM_SQDIFF" method, provided by the OpenCV library, to verify the matching between the binarized templates and the binarized image regions. Each template, represented by T , is slid over the source image, represented by I , and is compared against the overlapped image regions to find the match. Note that by sliding, we mean moving the patch one pixel at a time at each location.

$$R(x, y) = \sum x', y' (T(x', y') - I(x + x', y + y'))^2 \quad (1)$$

The calculated metric represents how similar the template is to that particular area of the source image. Each pixel location (x, y) in the result matrix R contains the match metric of moving T over I .

The Figure 12 shows the result R of sliding the template object and applying the referred function, where the brightest locations indicate the highest matches. The location marked by the red circle is an example of one with the highest value, representing the location that is considered the match [19].

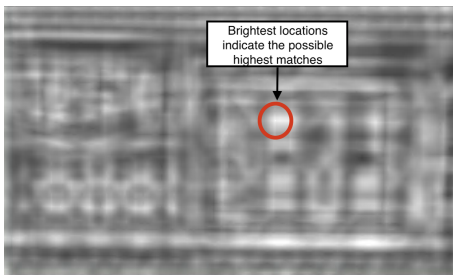


Figure 12. $R(x, y)$ results of applying the template matching technique.

In case of the lower bars of the console illustrated in Figure 13 (lower selection), when a defect occurs, the difference proportion between the input image and the bars template is

very tiny, making very difficult to detect this kind of fail with such low variation, which means that sometimes this algorithm is not able to detect this object. For all the other objects, the application of this technique has an accuracy of 100%.

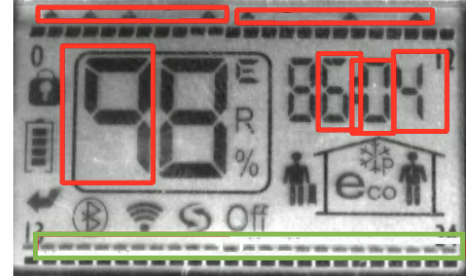


Figure 13. Matching results of the LCD display (areas surrounding by the rectangles are possible defects).

D. Classification of the Buttons' Condition

The detection of errors in the buttons of the HMI console was performed by using a kNN algorithm to the signal obtained from the force sensor, when the robot is pressing the button (see Figure 14).



Figure 14. Robot pressing the buttons during the inspection task.

This algorithm is a supervised learning method that consists in calculating the distances between the variable to be classified and the elements in the dataset. These elements are labelled and the predicted class is given by the more voted class inside the k nearest elements subset. The algorithm learned the differences between these button's conditions with a dataset X with 700 inputs, through the test of different buttons, with and without anomalies to perform the classification between "Ok", "Not Ok" and "Rigid" of the new data obtained by the torque-force sensor. Each input data is obtained through the action of pressing each button four times, being extracted and analyzed 7 different features, namely the maximum value (i.e. the highest force recorded), mean (i.e. the average value of the force), root mean square (RMS) (i.e. the magnitude of applied force), mean of derivative (i.e. the average value of the variation), RMS gradient (i.e. the magnitude of the applied force variation), maximum gradient (i.e. the largest positive variation) and minimum gradient (i.e. the greatest negative variation).

Figure 15 represents the data obtained from the force sensor Y in which it can be seen that the behavior has 4 peaks,

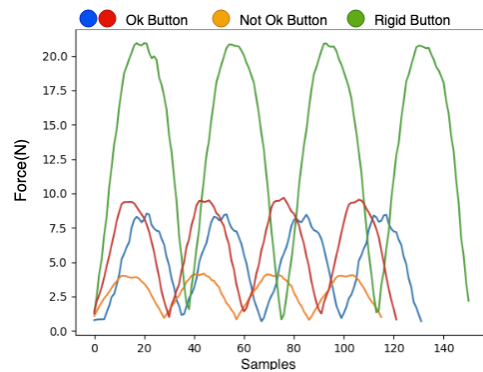


Figure 15. Real force data from the pressure action over buttons.

representing the 4 actions of pressing the buttons. From these curves, the features to classify the button condition can be extracted. For the classification task was used a KNN classifier, that reached an accuracy of 100%.

V. CONCLUSIONS

This paper describes the application of ML algorithms in a robotized artificial vision inspection station, to increase the flexibility and adaptability of the system to accomplish the QC tasks to a diversified set of HMI consoles. The adopted solution uses a deep CNN algorithm to perform the operator's facial recognition and the console detection, which allows to have a adaptable solution, with possibility to add new operators or different kind of HMI consoles to be inspected, in a simple and fast way, being only necessary to retrain final layer of the CNN.

The proposed approach shows the application of IA techniques to robotized inspection systems, supported by artificial vision systems, with applicability in several tasks of QC, increasing the reliability and optimization of the processes, reducing the costs and allowing to perform these kind of tasks with no human interference. With the use of larger amounts of data, the retraining of the ML algorithms will contribute to increase the accuracy of the face recognition and consoles detection algorithms, to values near of 100%.

For the detection of defects in the LCD displays, the adopted solution uses image processing techniques to perform the image correction and to reduce the impact that luminosity changes have in the image, combined with the template matching technique, which finds the areas of the image that are similar to the pre-defined template. For the correct classification of the buttons condition, was used a kNN algorithm, being previously trained with several anomalies examples, to be compared to the data obtained by the torque-force sensor. The applied algorithms showed promising results, with high accuracy values in the QC tasks, demonstrating the robustness and efficiency of these solutions to be applied in a real industry context.

Future work is devoted to improve the detection of errors in the LCD displays by adopting another ML techniques, combined with the improvement of the image acquisition

conditions, e.g., controlling the luminosity and the noise in the acquired image, which could eventually help to solve the problem of non-detection of small faults in the LCD displays.

ACKNOWLEDGMENT

The work reported in this paper was supported by Norte2020 - Programa Operacional Regional do Norte programme through the I40@TMAD project.

REFERENCES

- [1] G. Chrysosouris, D. Mavrikios, N. Papakostas, D. Mourtzis, G. Michalos, and K. Georgoulas, "Digital manufacturing: History, perspectives, and outlook," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 223, no. 5, pp. 451–462, may 2009.
- [2] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent Manufacturing in the Context of Industry 4.0: A Review," *Engineering*, vol. 3, no. 5, pp. 616–630, oct 2017.
- [3] J. Barbosa, P. Leitão, A. Ferreira, J. Queiroz, C. Geraldès, and J. P. Coelho, "Implementation of a multi-agent system to support zdm strategies in multi-stage environments," in *Proceedings of the IEEE 16th International Conference on Industrial Informatics (INDIN'18)*, 2018, pp. 822–827.
- [4] C. L. Paiva, "Quality Management: Important Aspects for the Food Industry," in *Food Industry*, I. Muzzalupo, Ed. Rijeka: IntechOpen, 2013, ch. 9.
- [5] Jon Fingas, "Google AI could keep baby food safe," 2017. [Online]. Available: <https://www.engadget.com/2017/07/25/google-ai-helps-make-safer-baby-food/>
- [6] Fujitsu Laboratories of Europe Ltd, "Fujitsu Develops State-of-the-Art AI Solution to Revolutionize Non-Destructive Testing Manufacturing Inspection - Fujitsu CEMEA&I," 2017. [Online]. Available: <http://www.fujitsu.com/fts/about/resources/news/press-releases/2017/emeai-20171002-fujitsu-develops-state-of-the-art-ai.html>
- [7] Fujitsu Forum Munich, "Artificial Intelligence Solution from Fujitsu Helps Siemens Gamesa Significantly Accelerate Quality Assurance Procedures - Fujitsu CEMEA&I," 2017.
- [8] O. Déniz, G. Bueno, J. Salido, and F. De la Torre, "Face recognition using Histograms of Oriented Gradients," *Pattern Recognition Letters*, vol. 32, no. 12, pp. 1598–1603, sep 2011.
- [9] Davis E. King, "dlib C++ Library." [Online]. Available: <http://dlib.net/>
- [10] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1875–1882.
- [11] J. Lu, J. Hu, and J. Zhou, "Deep metric learning for visual understanding: An overview of recent advances," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 76–84, 2017.
- [12] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International Workshop on Similarity-Based Pattern Recognition*. Springer, 2015, pp. 84–92.
- [13] 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.
- [14] M. Kawulok, M. E. Celebi, and B. Smolka, Eds., *Advances in Face Detection and Facial Image Analysis*. Cham: Springer International Publishing, 2016.
- [15] TensorFlow, "TensorFlow." [Online]. Available: <https://www.tensorflow.org/>
- [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [17] P. Kamencay, M. Benco, T. Mizdos, and R. Radil, "A New Method for Face Recognition Using Convolutional Neural Network," *Advances in Electrical and Electronic Engineering*, vol. 15, 2017.
- [18] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.
- [19] T. Mahalakshmi, R. Muthaiah, and P. Swaminathan, "An overview of template matching technique in image processing," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, pp. 5469–5473, 2012.