

## **Industrial Visual Inspection with TinyML for High-Performance Quality Control**

Andrea Albanese, and Davide Brunelli

In industrial processes, predictive maintenance or automated optical analysis of artifacts is fundamental to ensure high-quality products with low costs. However, this step is still done by sophisticated systems or human operators. Automating this process with low-cost solutions by keeping high product quality is one of the most challenging goals of the Industrial Internet of Things (IIoT). IIoT fosters an automation-based production model that uses machine data to enable faster, more flexible, and more efficient production lines [1], leading companies to produce higher-quality goods at lower costs.

Deep Learning (DL) is a powerful solution for implementing high-accuracy artifacts classification or detection; however, such algorithms are computationally intensive. A promising solution for implementing DL systems is edge computing and enabling the so-called tiny Machine Learning (TinyML) paradigm. It consists of performing all the processing near the data source, thus avoiding high-cost cloud infrastructures with their privacy and latency issues. Integrating ML capabilities on edge devices makes near-sensor data processing possible, creating more robust and scalable systems [2].

This article presents an automatic visual inspection system composed of three low-cost cameras to measure the quality of plastic components, as shown in Figure 1. A top camera is responsible for identifying shape defects, and two side cameras are used for color compliance checks. The idea comes from the needs of manufacturing companies that produce plastic products. During the molding process, some pieces may have shape and/or color defects. Thus, it is crucial for companies to promptly detect these artifacts to act on the settings of the plant to restore proper working conditions. The main contribution of this work is the development of a cyber-physical system that can detect defects in plastic molded objects and conduct product quality inspections with tiny NNs (neural networks) trained on ad-hoc datasets. The system inspects one item by running tiny NNs on the edge, and decides to keep or discard it by activating a plunger. The entire process is completed within 0.5 seconds with an impressive 99 % accuracy in defect classification.

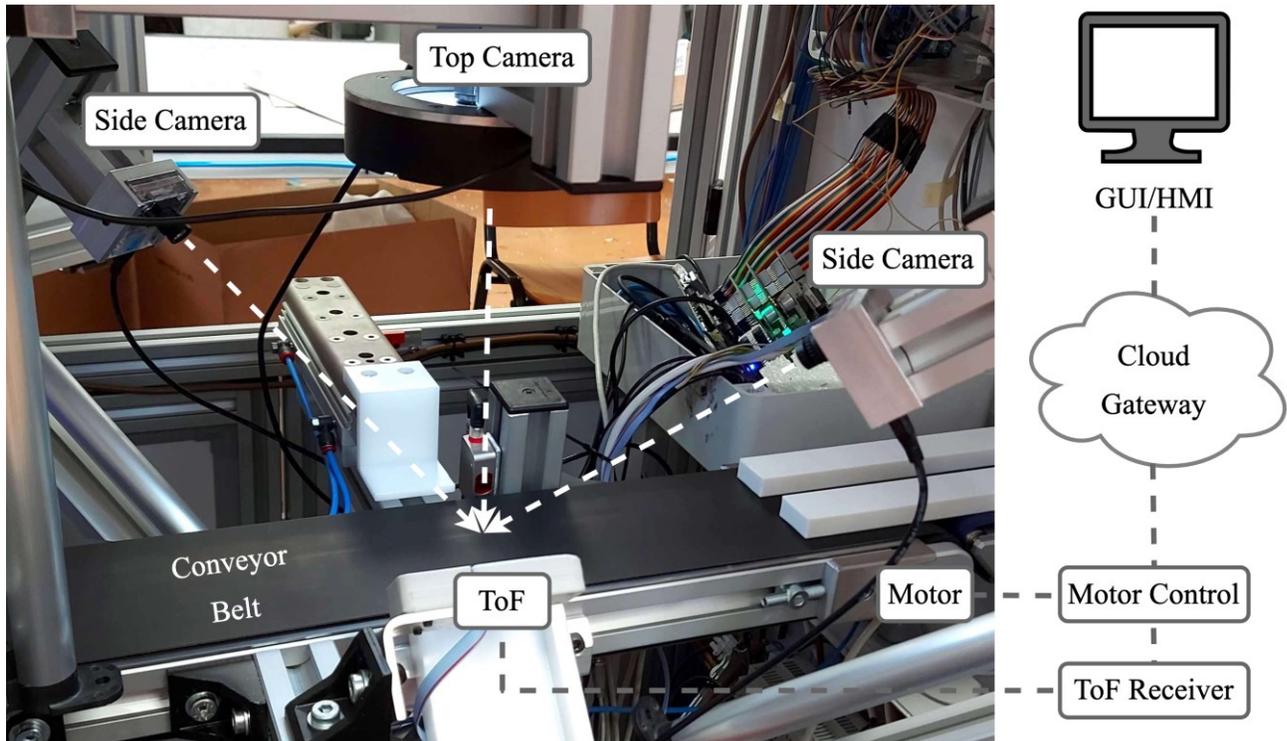
### **Overview of Industrial Visual Inspection**

Image sensors are widely spread in many scenarios, such as medical 2D imaging [3], and nanoparticle characterization with transmission electronic microscopy (TEM) images [4]. Imaging is also widely employed in visual inspection and declined in many industrial scenarios. Furthermore, it is crucial to quickly identify and address any issues or abnormalities during the molding process to ensure efficient operation and improve production. Additionally, integrating DL algorithms has significantly improved the robustness and precision of industrial visual inspection. Thus, it is possible to create models that can quickly assess the quality of a product, such as checking for the correct label on a bottle or identifying defects on the surface [5].

Many case studies have demonstrated the potential of industrial visual inspection. For instance, the authors in [6] describe a solution that automatically inspects liquid drug bottles on a production line. It uses an ensemble learning (EL) algorithm for detection based on multiple features and a tunnel structure that allows the bottle inspection to be automated without disrupting original processes and devices. The work presented by [7] shows the ability of CNN models to measure wood quality in timber bundle images to support the Swedish forest industry. So far, the wood quality inspection is made by operators; however, with the proposed automatic visual inspection systems, it is possible to obtain an accurate quality measurement and employ the operators for

more creative tasks. The authors in [8] also provide a comprehensive and organized overview of the existing and emerging automated computer-vision-based methods for classifying defects in three types of flat steel products: con-casting slabs, hot-rolled steel strips, and cold-rolled steel strips. They examine approximately 140 studies on the topic and show that visual inspection is a promising solution in an industrial environment.

However, all the systems presented in the above case studies have a common limitation: the usage of industrial cameras. They are expensive solutions with a high-energy demand. This article addresses this limitation by proposing a system composed of low-cost and low-power embedded cameras, namely the OpenMV Cam H7 Plus<sup>1</sup>. In this way, it is possible to obtain a visual inspection system that can be widely accessible.



**Fig. 1.** System design. It is composed of a conveyor belt, a sensor that detects objects using Time-of-Flight (ToF) technology, three cameras that use microcontroller units to process and classify images, and a cloud-based gateway that accepts or rejects the object based on the classification results. The system shows the top camera in the center of a ring light. Each camera is positioned 25 cm from the working surface to ensure that objects are in focus. The side cameras are at 30° with the vertical to ensure a perspective view.

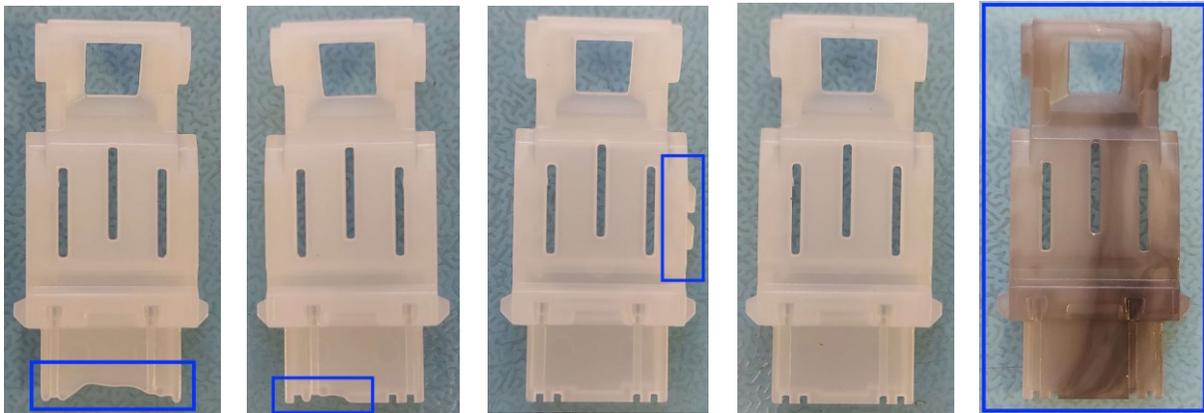
## System Design

The design of the visual inspection system is depicted in Figure 1. The workflow of the system can be organized into four distinct stages.

1. **Item Loading.** The initial step includes placing the item on the conveyor belt. There are no restrictions on the object rotation during the feeding process, which means the parts can be in any random rotation. The color of the conveyor belt was deliberately selected as "matte black" to prevent unwanted light reflections that may cause image distortions and incorrect classifications.

<sup>1</sup> <https://openmv.io/products/openmv-cam-h7-plus>

2. **Item Movement.** The object moves along the conveyor belt, driven by a stepper motor at the roller. The M4 core of the STM32MP1 MPU controls the motor through the X-NUCLEO-IHM03M1 expansion board. Additionally, items on the belt are detected with a Time of Flight (ToF) sensor. The same board also measures the ambient light to determine the lighting conditions when tagging and categorizing items. Both expansion boards are connected to the cloud gateway.
3. **Object classification.** In this step, the system performs object classification using an edge computing approach with three OpenMV Cam H7 Plus cameras (Figure 1). Two different models are used to classify shape and color abnormalities (like those in Figure 2). The position and number of cameras were chosen based on the types of defects to be detected. The cameras are placed in different locations to ensure that all faces of the object are captured. A "Top camera" is positioned vertically to the belt and focuses on the 2D plane of the object. In contrast, the "side cameras" are positioned at an angle to expand the field of view on the remaining four faces. The top camera is best suited for identifying shape defects because shape anomalies occur on the edges; the side cameras are best suited for detecting color defects because they can occur in all the faces of the object. The Deep Neural Networks (DNN) inputs use the captured images to classify shape and color imperfections.
4. **Post-processing.** The cloud gateway uses the information obtained from the three cameras to control the conveyor belt motor. If the objects are determined to be conformant, the conveyor belt motor is enabled. However, if the prediction result shows a non-conformant object, it is removed by activating a plunger.



**Fig. 2.** Possible defects of the items. From left to right: two incomplete items, deformed item, compliant item, and polluted item.

### Tiny Machine Learning

This system uses DNNs to classify color and shape anomalies in plastic objects on the same microcontrollers used to acquire the images. Despite the majority of DNN designs demanding significant computational power, which necessitates the use of specialized high-performance computing units, this project runs the inference on the MCU-based camera, which poses challenges in researching and optimizing DNNs to work within these constraints. Two state-of-the-art DNNs, namely MobileNetV2 and SqueezeNet, have been selected because they satisfy the company

requirements, and other works confirm their potential in such applications [9] [10]. Their structure is optimized for a small memory footprint by combining compression techniques such as parameter optimization, pruning, and quantization.

### ***Model Optimization***

Three different techniques were applied to MobileNetV2 and SqueezeNet to make the trained models compatible with the OpenMV Cam H7 plus.

1. **Parameter optimization.** DNNs have a massive number of parameters, which are the sum of the weights and biases for each layer. We reduced the number of parameters without a negative impact on the overall accuracy. MobileNetV2's blocks were reduced from 17 to 14, and SqueezeNet's blocks were reduced from 8 to 5. An ablation study led to a parameter reduction from 723k to 412k for MobileNetV2, and from 337k to 176k for SqueezeNet without considerably affecting the models' performance.
2. **Pruning.** This technique is applied to reduce the model complexity further. It allows for cutting off weights that are not relevant for making predictions, such as weights that are close to zero or rarely activated. In this case, we applied a mildly aggressive pruning to both models, resulting in a model of only 50% of the original parameters. This threshold was chosen to ensure that the model's accuracy would not be affected.
3. **Quantization.** This operation reduces not only the size of the model but also storage and memory usage during inference by replacing the model's 32-bit float representation with an 8-bit representation. Quantization improves hardware acceleration latency and power efficiency when deploying DNNs on MCUs. A quantized model with 8-bit representation will be 4x smaller and 1.5x-4x faster in computations. In this work, we quantized weights, activations, network inputs, and outputs, resulting in a "full-integer" quantized model.

Consider that the two DNNs for color and shape anomalies classification utilize the same model architecture and input image size for both datasets, resulting in an equal number of parameters and resources needed. Table 1 shows the result of the model optimization procedure considering the RAM usage. SqueezeNet and MobileNetV2 gain a compression factor of 3.8x and 3.9x, respectively.

**Table 1.** The RAM required for the models to perform inference on the MCU-based camera OpenMV Cam H7 Plus. The Float32 model refers to the models trained with the optimized structure obtained with parameter optimization. The Int8 model refers to the further optimized models with pruning and full-integer quantization. The last column shows the achieved compression factor.

160x160px	<b>Float32 model</b> (Parameter optimization)	<b>Int8 model</b> (Pruning and full-integer quantization)	Compression
	RAM (KB)	RAM (KB)	
<b>SqueezeNet</b>	1780.00	455.25	<b>3.80 x</b>
<b>MobileNetV2</b>	1490.00	381.22	<b>3.90 x</b>

## System Development

The development of the system can be divided into three steps: image preprocessing, dataset collection, and DNN training.

1. **Image Preprocessing.** An image preprocessing algorithm is developed to analyze the images before feeding them to the DNN models. This fundamental step in DL permits the adjustment of the images to fulfill the DNN's input requirements and emphasize relevant features useful for classification. In this case, the image preprocessing algorithm removes the background as it is irrelevant for classifying object anomalies. Moreover, it permits considering the object's position and avoids incorrect clipping of the image. It consists of a combination of three state-of-the-art computer vision methods. The “Canny” algorithm is first applied to the image to find component contours. This method highlights all the contours in the image and not only the object's contours. Then, blob detection is applied to the extracted contours to find the object's exact location. It follows the blob's merging and blob center computation. The blob center is used as a reference point to crop the image (size 160 x 160) and avoid clipping. Finally, Otsu's method is used to find the optimal threshold for background removal.
2. **Dataset Collection.** The dataset for training the DNN models is collected with the setup shown in Figure 1. Collecting a big and heterogenous enough dataset is fundamental to ensure good system performance. In this case, two different datasets were organized: one for color defects and one for shape defects. The color-defected dataset suitable for the side cameras consists of 6800 images equally divided between conformant and color-defected. The shape-defected dataset consists of 10800 images equally divided between conformant and shape-defected.
3. **Training.** MobileNetV2 and SqueezeNet are trained by using RGB images with a size of 160 x 160 pixels, an initial learning rate of  $10^{-5}$ , the Adam optimizer, and a binary cross-entropy loss function. The training starts with weights initialized to random values, while the number of epochs is set by using the early stopping callback by validation accuracy. This approach stops the training at the optimal point where the model has gained sufficient knowledge of the problem and avoids overfitting. In this case, a validation accuracy threshold is set to 99.5%; thus, the training stops at the epoch that reaches that required accuracy. Considering the top camera, the training needs 25 and 38 epochs for MobileNetV2 and SqueezeNet, respectively. The training for side cameras needs 12 and 100 epochs for MobileNetV2 and SqueezeNet, respectively. The top camera is employed for classifying shape defects; thus, color information is not necessary. In this case, the images are binarized to highlight the object's perimeter but still in the RGB color space to avoid the model's incompatibility in producing the prediction. On the contrary, side cameras oversee identifying color defects; thus, it is fundamental to keep the color information. In this case, the images are in the RGB color space and pixels can take values from 0 to 255.

## Experimental Results

DNNs are assessed by using a specific dataset never used during the training. In this way, it is possible to ensure the generalization capability of the network and its performance. The models' performance for the top camera and side cameras is evaluated by considering accuracy, precision, recall, f-score, and the loss in accuracy due to the optimization process. The top and side camera models are trained with two different datasets (i.e., one for shape defects and one for color defects); thus, their performance is different. Moreover, the platform is characterized by measuring the execution time of the selected algorithms.

### Top Camera

The top camera is responsible for classifying conformant and shape-defected items. In this case, 860 images are used for testing, where 410 images represent conformant objects, and the remaining 450 images represent shape-defected objects. The test results are summarized in Table 2. Notice that even though the model is compressed, the loss in performance is negligible. Moreover, the minimization of false negatives (FN) is fundamental in an industrial visual inspection setting. Analyzing the recall defined in (1), it is possible to choose the best-performing model. In this case, MobileNetV2 achieves a recall of 100%, which means that any FN is detected during the test. Thus it is preferable to SqueezeNet.

$$recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (1)$$

**Table 2.** Models' performance for the top camera. “Base” refers to the baseline model without optimization, and “Optimize” refers to the optimized models with parameter optimization, pruning, and quantization. “Loss” highlights the loss in accuracy during the optimization process.

	MobileNetV2			SqueezeNet		
	Base	Optimize	Loss	Base	Optimize	Loss
Accuracy	99.5%	98.9%	<b>0.6%</b>	98.6%	98.4%	<b>0.2%</b>
Precision	99.0%	98.0%	<b>1.0%</b>	99.0%	99.0%	<b>0.0%</b>
Recall	100.0%	100.0%	<b>0.0%</b>	98.0%	98.0%	<b>0.0%</b>
F-score	99.5%	99.0%	<b>0.5%</b>	98.0%	98.0%	<b>0.0%</b>

### Side Cameras

The side cameras classify conformant and color-defected objects, and the same tests for the top camera are conducted. In this case, 540 test images are split equally between the two classes. Table 3 summarizes the test results for the side cameras. Also, the optimization process does not generate a significant performance loss in this case. Furthermore, MobileNetV2 is still the best-performing model considering the minimization of FNs.

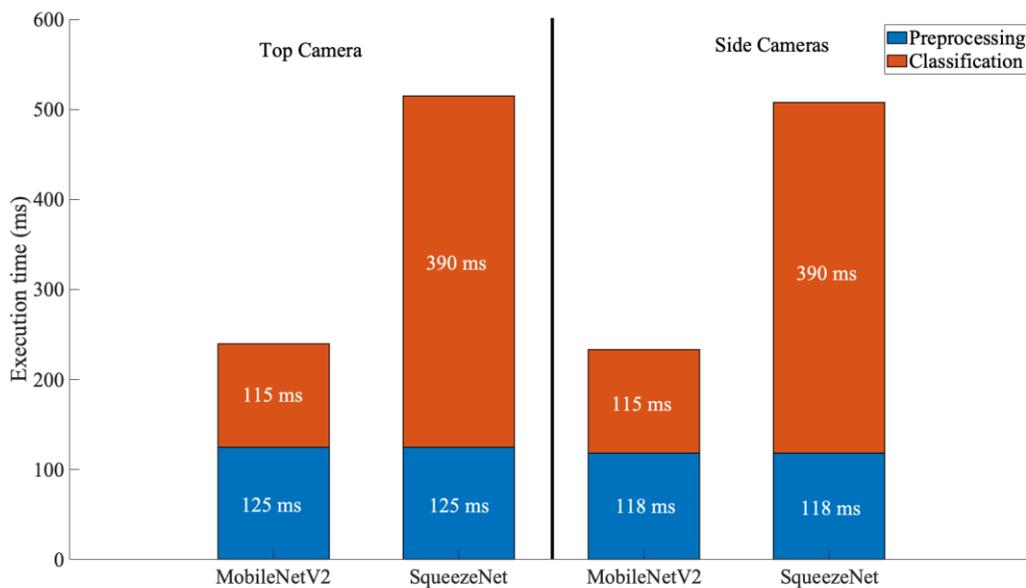
**Table 3.** Models' performance for the side cameras. “Base” refers to the baseline model without optimization and “Optimized” refers to the optimized models with parameter optimization, pruning, and quantization. “Loss” highlights the loss in accuracy during the optimization process.

	MobileNetV2			SqueezeNet		
	Base	Optimized	Loss	Base	Optimized	Loss
Accuracy	100.0%	99.6	<b>0.6%</b>	98.4%	98.2%	<b>0.2%</b>
Precision	100.0%	100.0%	<b>0.0%</b>	100.0%	100.0%	<b>0.0%</b>
Recall	100.0%	99.0%	<b>1.0%</b>	96.0%	96.0%	<b>0.0%</b>
F-score	100.0%	99.5%	<b>0.5%</b>	98.0%	98.0%	<b>0.0%</b>

### Execution Performance

The system performance is evaluated by considering the computational time of image preprocessing and DNNs classification inference. Furthermore, given that the nominal power consumption of the OpenMV H7 Plus in active mode is 0.8 W, it is possible to estimate the energy consumption of the camera system. Figure 3 shows the comparison of the execution time for MobileNetV2 and SqueezeNet. Considering both top and side cameras, MobileNetV2 outperforms SqueezeNet taking only 240 ms and 233 ms to process one image, respectively. Thus, the total energy required by the system is equal to 192 mJ for the top camera and 186 mJ for the side cameras. This system could also be battery-powered, thus deployable in positions where an unlimited energy source is unavailable.

Furthermore, the experimental results confirm the system's suitability for inspecting plastic components. Considering the components in Figure 2, the molding process takes about 20 seconds to produce two moles of 8 items. Consequently, 240 ms for processing one object is enough to guarantee the continuous operation of the production line.



**Fig. 3.** Execution time comparison for MobileNetV2 and SqueezeNet. The left side of the graph is related to the top camera, while the right side is related to the side cameras.

## Conclusions

Image sensor systems with DNNs can successfully inspect the quality of objects in an industrial process. The proposed system uses two DNN architectures, MobileNetV2 and SqueezeNet, which achieved a classification accuracy of 99% and 98%, respectively. The system can detect and classify two different anomalies in the objects, which are color and shape defects, and is designed to work in real-time with a frame rate of 5 FPS and 2 FPS for MobileNetV2 and SqueezeNet, respectively. Future improvements will use new training techniques such as neural architecture search, integrating continual learning capabilities, and improving the system's efficiency with more optimized DNNs.

## Acknowledgment

This work was supported by STMicroelectronics, Italy, and Rosa Micro srl, Ceggia, Italy. Funded by the European Union under NextGenerationEU (Project: Interconnected Nord-Est Innovation Ecosystem - iNEST). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or The European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

## References

- [1] F. Pilati, A. Sbaragli, M. Nardello, L. Santoro, D. Fontanelli and D. Brunelli, "Indoor positioning systems to prevent the COVID19 transmission in manufacturing environments," *Procedia CIRP*, vol. 107, pp. 1588-1593, 2022.
- [2] R. Sanchez-Iborra and A. F. Skarmeta, "TinyML-Enabled Frugal Smart Objects: Challenges and Opportunities," *IEEE Circuits and Systems Magazine*, vol. 20, no. 3, pp. 4-18, 2020.
- [3] V. A. Rani and S. Lalithakumari, "Efficient Medical Image Fusion Using 2-Dimensional Double Density Wavelet Transform to Improve Quality Metrics," *IEEE Instrumentation & Measurement Magazine*, vol. 24, no. 4, pp. 35-41, 2021.
- [4] A. Lay-Ekuakille, M. A. Ugwiri, M. Di Paola, F. Conversano, S. Casciaro, C. Liguori and V. Bhateja, "Functionalized Nanoparticles for Characterization Through TEM Images: Comparison Between Two Innovative Techniques," *IEEE Instrumentation & Measurement Magazine*, vol. 24, no. 4, pp. 29-34, 2021.
- [5] J. Tao, Y. Zhu, F. Jiang, H. Liu and H. Liu, "Rolling Surface Defect Inspection for Drum-Shaped Rollers Based on Deep Learning," *IEEE Sensors Journal*, vol. 22, no. 9, pp. 8693-8700, 2022.
- [6] L. Ma, X. Wu and Z. Li, "High-Precision Medicine Bottles Vision Online Inspection System and Classification Based on Multifeatures and Ensemble Learning via Independence Test," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-12, 2021.
- [7] M. Carratù, V. Gallo, C. Liguori, A. Pietrosanto, M. O'Nils and J. Lundgren, "A CNN-based approach to measure wood quality in timber bundle images," *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1-6, 2021.
- [8] Q. Luo, X. Fang, J. Su, J. Zhou, B. Zhou, C. Yang, L. Liu, W. Gui and L. Tian, "Automated Visual Defect Classification for Flat Steel Surface: A Survey," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, pp. 9329-9349, 2020.
- [9] W. Xu, S. Zhang, F. Bai and T. Zhao, "Research on Improved Residual Network Classification Method for Defect Recognition of Thermal Battery," *IEEE Access*, vol. 10, pp. 113234-113248, 2022.
- [10] D. K. Jain, Y. Li, M. J. Er, Q. Xin, D. Gupta and K. Shankar, "Enabling Unmanned Aerial Vehicle Borne Secure Communication With Classification Framework for Industry 5.0," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5477-5484, 2022.

**Andrea Albanese** (Student member IEEE) received the M.S. degree in mechatronics engineering with robotics and electronics specialization in 2020 from the University of Trento, Italy. He is currently a PhD student at the Department of Industrial Engineering at the same university. His research interests are machine learning optimization techniques and online learning systems for resource-constrained devices such as microcontrollers.

**Davide Brunelli** (Senior Member, IEEE) received the M.S. (cum laude) and PhD degrees in electrical engineering from the University of Bologna, Italy, in 2002 and 2007, respectively. He is an Associate Professor of electronics with the Department of Industrial Engineering, University of Trento, Italy and Head of the Laboratory of Embedded electronics and measurement systems. His current research interests include new techniques of energy scavenging for IoT and embedded systems, the optimization of low-power and low-cost consumer electronics, and the interaction and design issues in embedded personal and wearable devices.