Engineering Applications of Artificial Intelligence 177 (2026) 114850

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Research paper

Enhancing quality control in die-casting with ensemble-based computer vision methods☆

Paula Mielgo [a,*], Anibal Bregon [a], Carlos J. Alonso-González [a], Miguel A. Martínez-Prieto [a], Daniel López [b], Belarmino Pulido [a]

[a] Department of Computer Science, University of Valladolid, Paseo de Belén, 15, Valladolid, 47011, Spain
[b] Direcciones Centrales, HORSE, Spain, Av. Madrid, 72, Valladolid, 47008, Spain

ARTICLE INFO

Keywords:
Deep learning
Die casting
Ensemble
Computer vision

ABSTRACT

The transition towards Industry 4.0 has led to a significant increase in the adoption of smart manufacturing, where advanced technologies, such as Artificial Intelligence and Machine Learning, are used to optimize production processes. Quality control in manufacturing presents significant challenges, particularly in detecting non-visible defects. This paper proposes a novel approach to improve quality assurance in die-casting machines for car engine block production through thermographic image analysis. Specifically, we verify whether thermal patterns in the mold, captured immediately after the part is extracted, can serve as an indicator of internal defects in manufactured components, thereby avoiding the need for expensive and time-consuming leak tests. Our approach employs a stacking ensemble as its core. The ensemble integrates Convolutional Neural Networks and Vision Transformers, leveraging their complementary strengths for defect detection. An ensemble and threshold selection process is then carried out to identify optimal classifiers for defective and non-defective parts. Experimental results based on thermographic images from a mold used in the manufacture of 4-cylinder engine blocks demonstrate that the proposed framework can ensure the internal quality of up to 63.3% of components with high confidence. This result enables a significant reduction in reliance on leak tests, illustrating the viability of a real-time, cost-effective decision-making process that reduces bottlenecks and enhances overall manufacturing efficiency.

1. Introduction

Industry 4.0 aims to facilitate a digital transformation of manufacturing and production systems, resulting in the creation of intelligent factories. These factories will rely on data from cyber–physical systems throughout the manufacturing process, aiming to enhance manufacturing performance (Hermann et al., 2016). Cyber–physical systems, a fundamental technology of Industry 4.0, often involve control systems, embedded software, and a substantial amount of data gathered from sensors and actuators. This vast quantity of data must be integrated and analyzed to achieve the designation of "smart factories".

The concept of smart manufacturing was first introduced in the United States to facilitate the deployment of emerging technologies in the manufacturing sector. These include the Industrial Internet of Things (IIoT) and Artificial Intelligence (AI). The concept of smart manufacturing, also known as intelligent manufacturing, is focused on the adoption of advanced information and manufacturing technologies to optimize production processes (Zhong et al., 2017). The prime objective of this methodology is to enhance the quality, traceability, and efficiency of the production process. Machine Learning (ML) and, in particular, Deep Learning (DL) play a pivotal role in the contemporary development of smart manufacturing. The large amount of available data enables the development of data-driven models using ML techniques. These models reduce production time, improve quality, and eliminate unnecessary waste. Some literature reviews demonstrate the use of ML techniques in industrial environments to enhance planning and control procedures (Usuga Cadavid et al., 2020), as well as specific applications in quality control tasks with Support Vector Machine (SVM) algorithms and tree-based models (Peres et al., 2019; R. Li et al., 2021), or Artificial Neural Network (ANN) architectures (Saleh et al., 2022), which improve the performance of classical approaches.

More recently, DL has been a significant contributor to the advancement of computer vision. Architectures such as Convolutional Neural

☆ This article is part of a Special issue entitled: 'AI-Driven Innovations in Cyber-Physical Systems' published in Engineering Applications of Artificial Intelligence.

* Corresponding author.
E-mail addresses: paula.mielgo@uva.es (P. Mielgo), anibal.bregon@uva.es (A. Bregon), calonso@uva.es (C.J. Alonso-González), miguelamp@uva.es (M.A. Martínez-Prieto), daniel.g.lopez@horse.tech (D. López), b.pulido@uva.es (B. Pulido).

https://doi.org/10.1016/j.engappai.2026.114850
Received 13 December 2024; Received in revised form 5 March 2026; Accepted 13 April 2026
Available online 20 April 2026

Networks (CNNs) (Krizhevsky et al., 2012) or Vision Transformers (ViTs) (Dosovitskiy et al., 2021) have made significant advancements in many fields, including manufacturing. Recent proposals in defect detection (Alber et al., 2024) have combined DL and statistical approaches to enhance accuracy and efficiency. Techniques such as preprocessing by overlapping natural texture segments and excluding irrelevant regions have been employed to improve autoencoder models for defect identification on complex surfaces (Mehta and Klarmann, 2023). CNNs have demonstrated good performance, particularly when employing two-stage methods, as in Jin et al. (2025), where a multi-perspective object detection network (YOLO-DSF) was combined with a character recognition module, achieving high accuracy in industrial product classification tasks. Additionally, industrial implementations, such as pretrained models integrated with Programmable Logic Controllers (PLCs), have enabled real-time defect detection in quality control processes (Ozdemir and Koc, 2019). These developments underscore significant improvements in precision, adaptability to varying conditions, and the feasibility of practical application.

In this manuscript, we present a proposal to improve the quality control in a die-casting process through image analysis. In particular, we focus on the die-casting process used in the manufacturing of engine blocks at a car engine manufacturing plant. During the current operation process, various quality control checks are performed to assess the quality of the manufactured parts. First, the die-casting machine performs its own quality control checks on each part and discards those that are considered defective. Second, each part undergoes a visual inspection process that discards parts with visible defects. Hence, those parts with either anomalies in the casting process or with visible defects are rejected in the first stages of the manufacturing process. However, a small percentage of the parts can exhibit internal defects not evident to these two quality control tests. Consequently, even if the number of parts with non-visible defects is minimal, ensuring high product quality requires that all manufactured parts undergo a leak test. The leak test provides a means of assessing the quality of the final product; however, its application can be costly and represents a bottleneck in the overall process.

In the current workflow, computed tomography images are taken from a reduced number of manufactured parts (only one per shift). Since this is a slow and costly process, it cannot be considered for real-time decision-making. Consequently, for affordable and real-time automatic decision-making, we can only count on the thermograms taken after the casting of each part. However, these thermograms are taken from the mold immediately after the part has been extracted and not directly from the part itself. Our hypothesis in this work is that information about thermal patterns in the mold after the casting process provides clues about the quality of the manufactured part, particularly regarding non-visible defects. This leads to the main research question of this work: (*R1*) *can we predict, using thermograms of the mold, the quality of the parts that are considered correct in the first stages before they reach the leak test?* It is important to note that this task can be done either by identifying a part as defective (NK) and sending it for melting down or, more importantly, since most of the manufactured parts are non-defective (OK), by determining, with absolute certainty, that a part is correct. In both cases, the result is the same: the parts will skip the leak test, thus saving the cost of performing such a test and, more importantly, reducing the bottleneck in the testing process, which will eventually lead to an enhanced throughput.

The main contributions of this work are: (1) An end-to-end solution for detecting defective manufactured parts by processing thermal images captured from different cameras. The proposed approach encompasses image preprocessing, the utilization of CNNs and ViTs as backbone architectures, the combination of the best backbone models via ensemble techniques with various classical machine learning methods, and the final selection of the two best ensemble models. (2) An extension of the stacking technique that uses the output confidence score vectors from the meta-learners to identify two models specialized in detecting defective and non-defective parts, respectively. Both models are then jointly applied during inference to generate partial quality decisions, which are subsequently combined to produce the final label for each part. (3) Experimental validation of the proposed architecture in a real industrial scenario using a subset of thermographic images acquired during the manufacturing of 4-cylinder engine blocks. The results demonstrate that this approach can confidently label up to 63.3% of the parts, enabling almost two-thirds of the components that currently undergo the leak test to potentially bypass this inspection stage.

The remainder of the paper is organized as follows. Section 2 presents the case study of the die-casting machine. Section 3 discusses the related work. Section 4 provides a brief introduction to the preprocessing/DL/ensemble techniques employed in this work. Section 5 presents the dataset and the evaluation metrics considered. Section 6 presents the proposed architecture for quality control at the die-casting plant, based on the analysis of thermographic images of the mold. Section 7 discusses the results obtained with the proposed ensemble architecture against the use of individual state-of-the-art computer vision architectures. Finally, Section 8 draws the main conclusions and future directions of this work.

2. Case study

The case study presented in this work is grounded in the context of an industrial manufacturing process. In particular, it focuses on the fabrication of engine blocks through an aluminum die-casting process, which is performed in a factory located in the city of Valladolid (Fig. 1(a)). Fig. 2 shows the layout of the robots, cameras, and other components within the production area. The die-casting process (illustrated in Fig. 3) is conducted as follows. The initial stage of the process is the aluminum melting. Aluminum ingots are melted in a melting tower at a temperature of approximately 720 degrees Celsius. The molten aluminum is stored in holding furnaces to maintain a constant temperature, ensuring it is ready for use. Subsequently, the molten aluminum is placed into a shot chamber using a small ladle. The material is then injected with a force of 22 kN into a steel mold known as a die. The mold consists of a fixed side and a moving side, which remain together during the die-casting process. After that, to reduce the temperature and facilitate the solidification of the part, water and air are introduced into the cooling circuit. This is crucial to ensuring that the engine block is manufactured correctly. An incorrect solidification process could result in non-visible defects, such as porosity, which conventional quality control systems cannot detect. Once this process is finished, the moving side of the mold is separated, and the manufactured part is extracted using a robotic arm. Each part is uniquely identified with an alphanumeric string denoted as datamatrix. Thereafter, a thermographic camera takes a thermogram of each part of the mold. The mold is subsequently sprayed with a release agent to prevent the following part from sticking. Once the part has been extracted from the mold, it undergoes a visual inspection (Fig. 1(b)) and a thermal treatment to release any stresses in the material. As a final stage of the manufacturing process, a series of machining operations are performed on the parts, which then undergo a leak test to ensure the final quality.

Consequently, two thermal images are captured for each manufactured part, one for each side of the mold. For each thermogram, multiple files are generated. First, the source file, which contains the temperature values for each pixel, is only readable with a specific proprietary software. Second, a CSV file containing the different regions of the mold and the maximum temperatures found within them. Finally, an RGB (Red, Green, Blue) image in JPG format.

(a) Aluminum die-casting machine (b) Visual control

Fig. 1. Factory images.

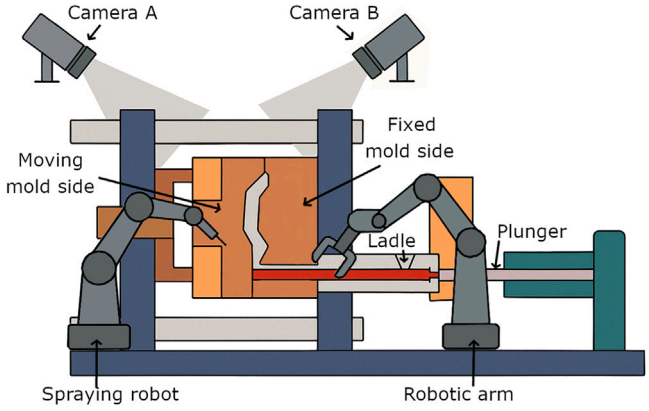


Fig. 2. Layout of components for the die-casting process.

3. Related work

Since the proposal of this work uses thermograms for quality control purposes, this section reviews the current state of the art in quality control, with special emphasis on the computer vision solutions and image acquisition techniques, focusing particularly on thermographic images.

3.1. Quality control

Traditionally, quality control has been approached through the application of statistical techniques, knowledge-based methods, or ML algorithms. The objective of statistical methods is to identify behavioral deviations in process parameters or in the manufactured parts. Several examples in the literature illustrate the monitoring of variance changes, using techniques such as the cumulative sum (CUSUM) (Yashchin, 1994), the moving sum (MOSUM) (Hsu, 2007), or more complex methods like Bayesian Posteriors Updated Sequentially and Hierarchically (BPUSH) (Milo et al., 2015). In knowledge-based methods, expert systems and knowledge bases are employed for decision-making based on established rules and inferential processes. Some examples of these techniques in quality inspection tasks are (Paladini, 2000), which proposes a Decision Supporting Expert System (DSES) for selecting inspection methods, and Perner (1994), which describes a knowledge-based system integrating image processing for defect recognition in offset printing. Finally, ML enables systems to identify patterns and make decisions based on data. Several references address the quality control task using Support Vector Machine (SVM) algorithms and tree-based models (Peres et al., 2019; R. Li et al., 2021), as well as Artificial Neural Network (ANN) architectures (Saleh et al., 2022), enhancing the performance of classical approaches. On the one hand, this improvement can be attributed to the ability of ML techniques to model non-linear and non-dependent relationships of underlying distributions, while managing high-dimensional data. On the other hand, ML architectures are capable of understanding implicit patterns in data without relying on rules. Moreover, ML architectures can adapt to varying operating conditions, making them suitable for industrial applications.

DL represents an emerging area of interest within the field of ML that employs Deep Neural Networks (DNNs). In recent years, the DL revolution has led to substantial improvements in predictive and generative tasks. An advantage of DL is its ability to enhance performance when processing large datasets, which can benefit from the use of GPUs. Furthermore, in contrast to other classical ML models, it does not require manual feature extraction. In the domain of industrial process control, Recurrent Neural Networks (RNNs) have been used for defect detection and texture classification in fabrics (Kumar and Bai, 2023), while Attention Mechanisms have been employed to establish the operating condition of a multi-sensor system (Zhang et al., 2022) and for predictive maintenance in production environments (De Luca et al., 2023). Additionally, DL has made a significant contribution to the field of Computer Vision (CV), where architectures based on autoencoders (Hinton and Salakhutdinov, 2006), CNNs (Krizhevsky et al., 2012), or ViTs (Dosovitskiy et al., 2021) represent the state of the art. In the industrial context, a recent survey (J. Liu et al., 2024) provides a comprehensive overview of DL approaches for image anomaly detection, highlighting current methods, challenges, and future directions.

In Mehta and Klarmann (2023), a defect detection problem is presented using a dataset of melamine-faced board. Images are divided into patches and a K-Means clustering algorithm is employed to determine which of them are relevant. Following this process, smooth surfaces and background patches are dismissed. Furthermore, a data augmentation mechanism is implemented by overlapping natural texture segments from the Describable Textures Dataset (Cimpoi et al., 2014) before training an autoencoder model. In Wang et al. (2018), Wang et al. present a defect detector that is robust to noise and changing conditions, using a texture surface dataset with six different image categories. To this end, a twofold CNN-based architecture is proposed. The first component is a global network that classifies the image. The second component is a specific network for each class that is fed with image patches to determine whether they contain defects. The proposed approach not only improves the state-of-the-art results but also maintains a computational speed that ensures real-time processing. Recent studies have also been focused on deployment-oriented evaluations of DL models in casting environments. In Babawale et al. (2025), the authors compare a custom CNN and a lightweight MobileNetV2 for defect detection in cast submersible pump impellers. Although both models achieve comparable accuracy, the CNN demonstrates faster inference, offering practical benefits for large-scale industrial inspection. An efficient CNN model is proposed in Hu and Wang (2020) for casting defect detection on digital radiography images using image-level labels. The architecture consists of two modules: the Type Classification Module, which extracts relevant features for each type of casting, and the Defect Classification Module, which is focused on local defect detection based on the previously extracted features. The

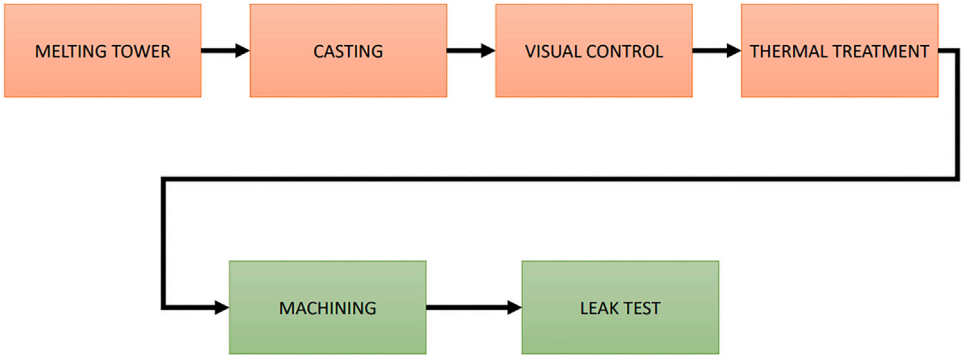


Fig. 3. Die-casting process diagram.

proposal demonstrates improvement in quantitative metrics and real-time efficiency over classical architectures through the employment of an object-level attention mechanism, bilinear pooling, and depthwise separable convolution. Additionally, casting defect recognition has also been addressed in Duan et al. (2020) using an improved YOLOv3-based architecture in digital radiography images. By modifying the network architecture and training strategy, the proposed model achieves better performance and faster convergence, demonstrating the suitability of DL frameworks for real-time industrial inspection.

In Qu et al. (2023), Component Attention Polyphase Sampling is introduced, which employs component attention polyphase downsampling and component attention polyphase upsampling to improve shift equivalence in CNNs. The approach ensures maximum similarity of downsampled features before and after image translation, leading to improved consistency in industrial defect segmentation models. The performance of ViTs is evaluated in Alber et al. (2024) using the MVTec AD (Bergmann et al., 2021) and the BTAD (Mishra et al., 2021) datasets. In that research, quality control is also approached with a twofold architecture. The first component is the vision backbone, which is based on CNN and ViT models. The second component is the defect detector, which is implemented with statistical models based on underlying distributions. Compared to CNN backbones, ViT backbones demonstrate superior performance, even when employing backbones with a lower number of parameters. Finally, it is possible to identify real prototypes in industrial environments that encompass the entire quality control process. In Ozdemir and Koc (2019), a defect detection problem is presented. The initial stage of the process involves image capture, where a webcam takes pictures of several products. These images are then fed into a pre-trained AlexNet model through a Programmable Logic Controller (PLC). According to the model decision, products could be retired from the production line. The successful integration of such prototypes into production environments reinforces the need to consider not only the performance of the models, but also transversal aspects such as reliability, interoperability, and risk management in systems, as discussed in recent works (Tabassi, 2023; Radziwill and Benton, 2017).

3.2. Image acquisition techniques

The electromagnetic spectrum represents the complete range of electromagnetic waves, including all forms of radiation. The visible spectrum is the region of the electromagnetic spectrum that the human eye can perceive, which includes wavelengths between 0.4 μm and 0.8 μm. However, some devices are capable of representing waves outside of the visible spectrum, enabling the analysis of specific object properties. Thus, different image-capturing techniques are employed in industrial contexts depending on the requirements of the problem. These include:

Standard images capture radiation from the visible spectrum. Their use presents some challenges, including the requirement for appropriate light conditions and limitations in foggy or smoky environments. Moreover, the analysis of standard images only allows for superficial inspections. Some application examples using this kind of image can be found in Hussain et al. (2022) for pallet racking and in C.-L. Li et al. (2021) for anomaly detection and localization using the MVTec AD dataset. Beyond single-image approaches that overlook contextual information and multi-scale characteristics, video-based methods have also been proposed to capture temporal dynamics and fine-grained defect information better, as demonstrated in Zhao et al. (2025), Hu et al. (2024).

X-ray images (Haff and Toyofuku, 2008) are obtained by the penetration of X-rays through objects. The objective is to emit radiation over the material, which may be absorbed or transmitted through it. A detector is positioned behind the object, capturing the rays that pass through it and generating an image. These inspections are fast and non-destructive but only provide superficial information. The principal disadvantage of this technique is the implementation cost. In addition to the acquisition cost, it is also necessary to hire expert staff and implement strict security measures to minimize radiation exposure. Moreover, the use of X-rays is limited by the material density and composition of the object under examination. Yang et al. (2021) propose a defect detection model based on X-ray images for steel pipe weld tasks, while Haff and Toyofuku (2008) focus on the food industry.

A **computed tomography image** (Müller, 2013) is a three-dimensional representation of the object under inspection. The capture process involves acquiring multiple X-ray images from various perspectives and subsequently processing them with a computer to generate cross-sectional images. Thus, internal components of objects can be inspected in a non-destructive manner. However, this technique presents the same challenges as previously discussed in the X-ray images subsection, with the additional issue of slower computational times, which prevents real-time decision-making. Several industrial applications of computed tomography images are presented in Fan and Sun (2019) for fatigue damage evaluation in concrete and in Busch and Hausotte (2022) for surface determination in three-dimensional object representations.

Multispectral images capture a discrete number of spectral bands, typically contained within the visible spectrum and in the infrared region. **Hyperspectral images** (Amigo, 2019) register information in a continuous manner over a wide range of wavelengths. Thus, hyperspectral images enable detailed analysis at the expense of high computational time, whereas multispectral images achieve a balance by capturing fewer spectral bands, resulting in a sacrifice of spectral resolution. In both cases, the required equipment is expensive and dependent on the environmental conditions. Real applications of multispectral and hyperspectral images can be found in Xing and De Baerdemaeker (2005) for bruise detection on apples, in Vila et al. (2005) for estimating chlorophyll in plant leaves, and in Serranti et al. (2011) for quality control in recycling processes.

Light Detection and Ranging (LiDAR) (Wandinger, 2005) technology is a widely used technique. The equipment consists of three

distinct components. First, a laser is employed to emit infrared rays which collide with objects situated in front of it. Second, a sensor is required to collect the reflected rays. The last component is a processor that calculates the time it takes for the light to return to the sensor and generates a point cloud. This information can subsequently be represented as a three-dimensional model or be projected in two dimensions to form a plane picture. LiDAR is capable of making high-precision measurements in real time. However, these systems are expensive and sensitive to dust, fog, and smoke. Several industrial applications are discussed in Wang et al. (2015) for quality control in construction tasks and in Park et al. (2022) for a robotic bin-picking platform.

Thermal images or thermograms are another commonly used technique. Infrared cameras (Vollmer, 2020) are designed to capture infrared radiation (between wavelengths from 0.8 μm to 15 μm), which is located between the visible spectrum and microwaves. Infrared radiation is fundamentally composed of thermal radiation. Therefore, an image generated with an infrared camera is often denominated as a thermal image or thermogram, where each pixel represents a temperature measurement. Consequently, thermal images are monochromatic, although they are typically mapped into a color scale to facilitate human interpretation.

The employment of thermal images offers several advantages. First, it is a non-invasive and non-destructive process. Measurements are taken without direct contact, working at a distance and avoiding disruption or damage to the inspected objects. Furthermore, it is a highly efficient process, thus images can be constructed in real time. Additionally, although the equipment acquisition could be expensive, the operation and maintenance costs are usually low.

Thermal images can be used to obtain supplementary information about thermal patterns or trends, which can be very valuable in specific fields, such as medicine (Ring and Ammer, 2012). In industry, the employment of thermograms can also be useful for anomaly detection tasks (Younus and Yang, 2012; Zhu-Mao et al., 2018), for facilities maintenance (Dai et al., 2017), or for quality control tasks (Grabowski and Cristalli, 2015; Varith et al., 2003; Cruz et al., 2021). Similar approaches based on thermographic imaging have also been explored for defect prediction in die casting processes (Michno et al., 2025).

4. Background

This section presents a detailed description of well-established preprocessing techniques, DL models, and ensemble learning methods that are employed in this work, for the sake of self-containment.

4.1. Preprocessing techniques

The employment of preprocessed images in computer vision tasks is a common approach for enhancing model performance. The following techniques are commonly used in the preprocessing of RGB images:

Channel decomposition is a point-to-point technique that consists of isolating individual channels from a multiple-channel image. In RGB images, three primary channels can be directly extracted: the red channel, the green channel, and the blue channel. This can be achieved by simply selecting their matrix values. Additionally, the complementary channels can be constructed by averaging the primary ones. Let x_i be the value of a pixel x in the channel i. Then, their values in the complementary channels, cyan, magenta, and yellow, can be computed as follows.

$$x_{cyan} = \frac{x_{green} + x_{blue}}{2}$$

$$x_{magenta} = \frac{x_{blue} + x_{red}}{2}$$

$$x_{yellow} = \frac{x_{red} + x_{green}}{2}$$

Channel decomposition has successfully been applied over HSV (Hue, Saturation, Value), CIEL*a*b*[1] or YCbCr[2] images as stated in Sachin et al. (2018).

The grayscale transformation consists of weighting the three RGB channels to obtain a single-channel image. This point-to-point transformation can be performed through the application of different formulas; however, the most popular is the one proposed by the National Television System Committee (NTSC) (Jack, 2011), which is computed as follows.

$$x_{grayscale} = 0.299 \cdot x_{red} + 0.587 \cdot x_{green} + 0.114 \cdot x_{blue}$$

The application of the grayscale transformation before feeding a model has been demonstrated to enhance classification results and reduce computational time (Xie and Richmond, 2018; Hsu et al., 2021).

The gamma correction is a non-linear transformation that modifies the intensity of pixel values. It is a point-to-point operation. For each pixel x, the gamma correction g is defined as follows:

$$g(x) = \left(\left(\frac{x}{255} \right)^{\left(\frac{1}{\gamma}\right)} \right) \times 255$$

where γ is a parameter.

This technique, originally developed to correct the power-law transformations in display devices (Gonzalez and Woods, 2008), has also been employed as an effective preprocessing technique, as evidenced in K. Wang et al. (2021).

The RGB-HOG transformation is the application of the Histogram of Oriented Gradients (HOG) algorithm through the three individual channels, resulting in a three-channel image. The HOG descriptor is a non-point-to-point technique that calculates the gradient orientations in small cells to represent local pattern shapes and textures. HOG is a well-established transformation that has been employed in one-channel image tasks (Dalal and Triggs, 2005; Déniz et al., 2011). However, as demonstrated in Lahmyed et al. (2019), retaining color information may be valuable when working with three-channel images. Consequently, the RGB-HOG transformation is introduced.

The Local Binary Pattern (LBP) is a point-to-point texture descriptor that computes the value of each pixel based on the intensity differences with its neighboring pixels. The image is initially transformed to grayscale, reducing it to a single channel. Then, for each pixel, the neighboring pixels are locally set to 0 or 1, depending on whether its value is greater than or less than the value of the central pixel. Finally, these binary values are combined to determine the central pixel value. LBP has been successfully employed in combination with CNNs for facial recognition tasks (Ravi et al., 2020; Bahroun et al., 2023).

4.2. DL models

DL has made significant contributions to the field of computer vision, mainly through well-established supervised architectures such as CNNs and ViTs. Furthermore, novel approaches, including SSM-based models such as VMamba (Y. Liu et al., 2024), are also contributing to ongoing advancements. Alongside these developments, generative paradigms such as diffusion models (Zhou et al., 2024) have recently emerged as a powerful alternative for anomaly detection and image reconstruction in industrial inspection tasks, highlighting the diversity of current DL strategies.

CNNs (Krizhevsky et al., 2012) are DNNs particularly suited to image processing tasks. Their structure is constructed through the

[1] CIEL*a*b*. Commission Internationale de l'Éclairage. L* represents the lightness, a* represents the chromatic axis from green to red, and b* represents the chromatic axis from blue to yellow.

[2] YCbCr. Y represents the luminance and CbCr the chrominance, representing the blue chroma (Cb) and the red chroma (Cr).

successive application of convolutional and pooling layers. On the one hand, convolutional layers generate feature maps from pixel matrices. This is achieved through a group of kernels that weigh the input values, and also with an activation function that introduces nonlinearity and restricts the output values of the layer. On the other hand, the objective of pooling layers is to reduce the dimensionality of the feature maps by using pooling kernels. This is done to reduce the computational time required for the training process while retaining the most significant information extracted in the convolutional layers. The final layer of a CNN is the fully connected layer, which is employed to transform the network output into a one-dimensional array. This layer is applied after the iterative application of convolutional and pooling layers. CNNs are designed to emulate the visual cortex, making them ideal for extracting patterns in images and videos (Li et al., 2014; Dyrmann et al., 2016). In industrial contexts, CNNs have been successfully employed in manufacturing inspection tasks (Weimer et al., 2016) for detecting surface defects.

A **ViT** (Dosovitskiy et al., 2021) is a DL architecture for image processing based on the Transformers architecture presented by Vaswani et al. in Vaswani et al. (2017). The main component of the ViT is the attention mechanism. This mechanism weighs the strength of relationships between the inputs, which correspond to image patches. Consequently, a weight matrix is constructed with values in the range $[-1, 1]$. Each element $x_{i,j}$ represents the relationship between patch i and patch j. The values 1 and -1 indicate a direct and inverse strong relationship, respectively, while the value 0 represents the absence of a relationship. To enable parallel computation, the input dimension is divided. This allows several attention mechanisms, referred to as attention heads, to operate independently. This is known as the Multi-Head Attention Mechanism.

The structure of a ViT begins with a position embedding, which preserves the positional information of the patches. Subsequently, the architecture employs encoder blocks, which are composed of alternating Multi-Head Attention Mechanisms and Multi-Layer Perceptron blocks. Moreover, the encoder block incorporates a normalization layer before each component and a residual connection after that. Finally, the ViT incorporates a Multi-Layer Perceptron Head to perform the classification. ViTs have also been employed for defect detection tasks in industrial environments, as stated in Smith et al. (2023), where several ViTs were compared with CNNs on a leather dataset.

4.3. Ensemble learning

While ML models can produce satisfactory results, more complex architectures may be capable of producing more robust and accurate models. One method for achieving this is through the use of ensemble techniques (Mohammed and Kora, 2023), which combine the results of multiple individual classifiers to improve overall performance. Several types can be distinguished according to different criteria, including the training method (parallel or sequential), the fusion approach (e.g., weighted voting or meta-learning strategies), and the heterogeneity among base learners. The following are some of the most common approaches:

- **Bagging** (Breiman, 1996), which is based on the resampling of training data with replacement. Each of these subsets will be used to train an individual model, with all models following the same architectural structure. Subsequently, the final prediction is determined by a majority vote among the individual models, with equal weight given to each model. This approach is particularly well-suited for unstable classifiers, where even minor alterations to the training set can lead to significant changes in predictions. Moreover, individual models may be trained in parallel, thereby reducing the computational time. In Liu and Ge (2018), the Random Forest bagging ensemble is presented in combination with a hierarchical clustering model selection in industrial processes.
- **Boosting** (Freund et al., 1999) is a sequential architecture that is based on the influence of previous models on subsequent models, with a particular focus on misclassified instances. All models are identical, and the final prediction is calculated by combining the predictions generated by each model through weighting. Experimental evidence indicates that boosting can reduce bias and loss variance. In Y. Wang et al. (2021) the boosting ensemble XGBoost is employed in conjunction with a Bayesian optimization algorithm to forecast the industrial electricity consumption.
- **Stacking** (Wolpert, 1992) is a technique that combines multiple models to create a more robust and accurate classifier. This approach employs a meta-learner to make the final decision by aggregating the outputs of individual classifiers. Consequently, the ultimate prediction is produced by a model trained on these outputs. The individual models are diverse and potentially complex, while the meta-learner is a simple and smooth model. This combination of classifiers enables more accurate predictions. However, it comes at the expense of reduced interpretability. In Ouyang et al. (2018), an anomaly detection of power consumption is addressed using the stacking ensemble learning approach.

5. Methodology

This section presents a description of the employed dataset, together with the cleaning and partitioning process. Moreover, the selected metrics for the evaluation procedure are detailed.

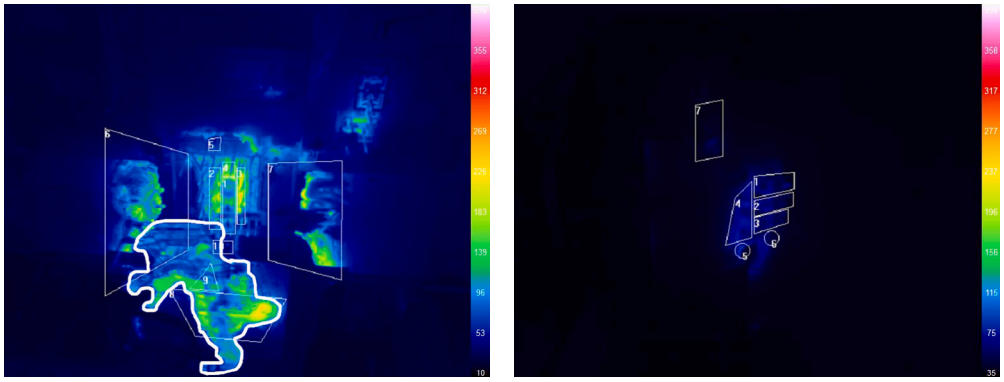
5.1. Raw dataset

The experimental dataset used in this work consists of thermal images captured after the die-casting process. As previously stated in Section 2, a pair of RGB images is captured for each manufactured part. It includes images taken from opposing perspectives with Camera A and Camera B, capturing the moving side and the fixed side of the mold, respectively. The plant has eight aluminum die-casting machines. The images in the dataset belong to one of the machines with a specific mold, which is the one used for fabricating the 4-cylinder engine. Moreover, all the images correspond to parts that have been identified as non-defective by both the quality control system of the die-casting machine and the visual inspection. Consequently, these images are labeled according to the results of the leak test, designated as OK for a successful test and as NK for a failed test.

5.2. Dataset construction

The dataset is highly imbalanced, with a rate of 97.5% of OK parts. As demonstrated by previous research (Buda et al., 2018), the use of these datasets can negatively impact the classifier performance. In cases of extreme imbalance, undersampling can be an effective approach to mitigate the problem. Therefore, a random elimination process was employed to remove OK images until the ratio reached 80% OK images and 20% NK images. There are 2536 raw images for each camera. However, several images contain the robotic arm used for part extraction, which prevents proper thermogram capture. This overlap is exclusively observed on Camera A. It is caused by temporal synchronization issues related to the high production rhythm of the plant (see Fig. 4(a) for an example of this type of object). Therefore, these images were removed. Moreover, several images in Camera B have minor variations in the thermal scale (Fig. 4(b)). Due to the limited number of affected images, these were also removed. This manual data cleaning process resulted in a final dataset comprising 944 images from Camera A and 2217 images from Camera B, referred to as the *cleaned dataset*.

While the cleaned dataset comprises a greater number of images, only 625 of these include images captured by both Camera A and Camera B. This subset of thermograms, designated as the *intersected dataset*, was employed in our preliminary work (Mielgo et al., 2024),



(a) Non-desirable object outlined (b) Thermal scale variation

Fig. 4. Examples of non-usable images.

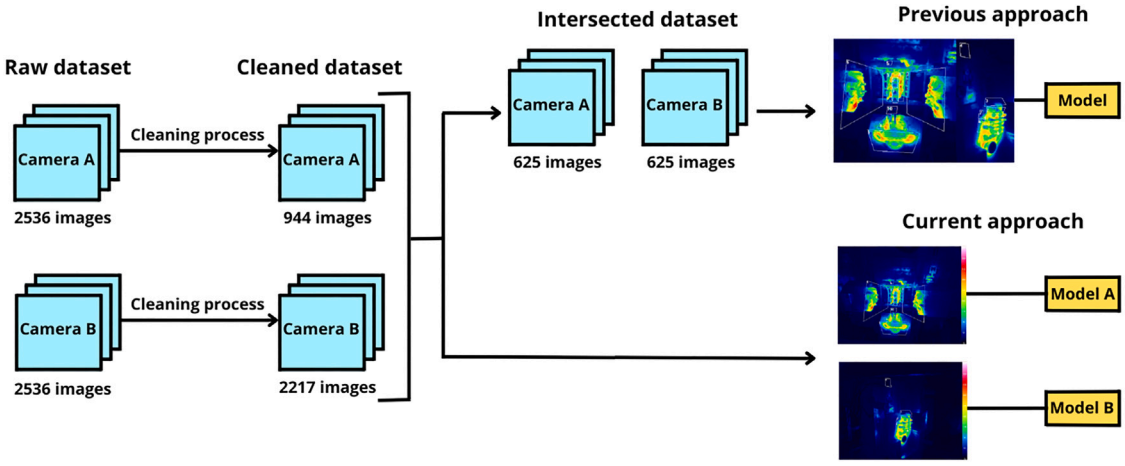


Fig. 5. Datasets considered in the proposal.

where a single thermal image was used to represent the fixed side and the mobile side of the mold. Consequently, both images were merged to create a unified representation of the parts. This results in the discarding of 34% of the images from Camera A and 72% of the images from Camera B. Therefore, to take advantage of the full set of images from both cameras, this work employs an alternative methodology in which specific models are constructed for each camera. An overview of both approaches is presented in Fig. 5.

5.3. Dataset partition

The cleaned dataset is partitioned into three subsets: train, validation, and test. This division is performed based on the datamatrix of the part, rather than on the individual images. This approach ensures that if a given part contains images from both cameras, they will be assigned to the same partition. The subsets are constructed as follows.

- The test subset was created with the same parts used in the previous approach, so it consists of 155 OK images and 33 NK images from both camera A and camera B sets. This decision ensures that evaluation is conducted in a fair and comparable manner across all models.
- The train subset was built by randomly selecting 70% of the remaining parts in a stratified manner, ensuring proportional representation of the OK and NK images. This results in 453 OK images and 81 NK images from the camera A set, and 1169 OK images and 256 NK images from the camera B set.
- The remaining images were incorporated into the validation subset. This includes 188 OK images, and 34 NK images from the camera A set, and 495 OK images and 109 NK images from the camera B set.

The cleaned dataset will be employed for training individual models. However, the meta-learner is designed to work with images from both cameras. Hence, we still need the intersected dataset. For both cameras, three partitions are created as follows.

- The test subset contains the images of the intersected datasets corresponding to the parts in the test subset of the cleaned dataset. This includes 155 OK images and 33 NK images from both camera A and camera B sets.
- The train subset contains the images of the intersected datasets corresponding to the parts in the train subset of the cleaned dataset. This includes 257 OK images and 55 NK images from both camera A and camera B sets.
- The validation subset contains the images of the intersected datasets corresponding to the parts in the validation subset of the cleaned dataset. This includes 103 OK images and 22 NK images from both camera A and camera B sets.

A summary of the datasets partition can be found in Table 1.

Table 1
Dataset partitions.

Dataset	Camera	Train	Validation	Test
Cleaned dataset	Camera A	453 OK & 81 NK	188 OK & 34 NK	155 OK & 33 NK
	Camera B	1169 OK & 256 NK	495 OK & 109 NK	
Intersected dataset	Camera A & B	257 OK & 55 NK	103 OK & 22 NK	

5.4. Evaluation metrics

The performance of classifiers is typically evaluated using a confusion matrix. In binary classification tasks, the confusion matrix is composed of the following elements:

- The True Positive (TP) number, which represents instances where the predicted and actual values are both positive.
- The False Positive (FP) number represents instances in which the predicted value is positive, while the actual value is negative.
- The False Negative (FN) number represents instances in which the predicted value is negative, while the actual value is positive.
- True Negative (TN) number, which represents instances where the predicted and actual values are both negative.

Using these components, some other metrics are computed. For example, the accuracy represents the percentage of correct predictions relative to the total. It is calculated as $accuracy = \frac{TP+TN}{TP+FN+FP+TN}$. However, accuracy is not an appropriate metric to use with imbalanced data because it provides a misleading representation of performance for the minority class. Other well-known metrics are precision and recall. Precision represents the percentage of correct positive predictions, and it is obtained by the formula $precision = \frac{TP}{TP+FP}$. Recall represents the percentage of true positive instances that the model is capable of correctly identifying over the total of positive elements. It is calculated as $recall = \frac{TP}{TP+FN}$. Nevertheless, neither metric alone is appropriate for evaluating imbalanced data. The F1 score metric, which is computed as $F1score = \frac{2 \cdot precision \cdot recall}{precision+recall}$, avoids this limitation by weighting precision and recall. This well-established metric assigns equal weight to both FP and FN values. However, this is not an accurate representation of our work, as the consequences of FN values are more severe than those of FP values. Therefore, the Fβ score metric was selected to address this limitation. It is a high-best metric, and its formula is as follows:

$$\mathrm{F}\beta \text{ score} = \frac{(1+\beta^2) \cdot \text{ precision } \cdot \text{ recall}}{\beta^2 \cdot \text{ precision } + \text{ recall}} = \frac{(1+\beta^2) \cdot \mathrm{TP}}{(1+\beta^2) \cdot \mathrm{TP} + \beta^2 \cdot \mathrm{FN} + \mathrm{FP}}$$

The Fβ score metric is an effective method for evaluating imbalanced datasets, because it allows for the minimization of classification errors in accordance with the specific requirements of the problem.

- If both FP and FN are equally important to prevent, then β should be equal to 1, resulting in the aforementioned F1 score.
- If FN is more important to prevent than FP, then β should be greater than 1.
- If FP is more important to prevent than FN, then β should be less than 1.

Due to the imbalance in the thermal dataset described in Section 5.2, the selected metrics for evaluating models are based on the Fβ score metric, with $\beta = 1$, which corresponds to the F1 score, and with $\beta = 2$, which results in the F2 score. The F1 score will be used for fine-tuning the individual models. The F2 score will be used to select the best individual models within the ensemble architecture and to perform the evaluation. This approach prioritizes the prevention of defective parts classified as OK, which is more important than classifying correct parts as NK, since those would be sent for melting down. Consequently, the defective class (NK) is designated as the positive one because it is the minority.

Moreover, complementary metrics will be used to enhance the understanding of the results. These are TP rate, FP rate, FN rate, and TN rate, which are defined as follows:

$$\text{TP rate} = \frac{TP}{TP+FN}$$

$$\text{FP rate} = \frac{FP}{FP+TN}$$

$$\text{FN rate} = \frac{FN}{FN+TP}$$

$$\text{TN rate} = \frac{TN}{TN+FP}$$

Ideally, TP rate and TN rate should be equal to 1, while FN rate and FP rate should be equal to 0.

6. Architecture

This section provides a detailed description of the proposed architecture. Fig. 6 presents a high-level overview of the architecture. It is comprised of four principal stages, represented by white boxes. The figure is divided into two sections. On the one hand, the offline processes include the architecture construction and training with historical data. On the other hand, the online process can be executed with real-time data. The *individual model training* stage, described in 6.1, involves a fine-tuning process of pretrained CNNs and ViTs using preprocessed images. The *ensemble training* stage, detailed in 6.2, utilizes the individual models trained in the previous stage, together with a meta-learner, to construct a stacked ensemble architecture. The third stage, *ensemble and threshold selection* 6.3, involves the selection of the best ensemble for the detection of OK parts and the best ensemble for the detection of NK parts. Additionally, the OK and NK thresholds are also determined in conjunction with these ensembles. Finally, the *decision procedure* stage, presented in 6.4, determines the classification for new parts.

6.1. Individual model training

This first stage of the architecture comprises two steps, which are executed consecutively. Initially, the image preprocessing step is performed, followed by the fine-tuning process of the pretrained models. Additionally, since there are two different images for each part (one captured by Camera A and the other captured by Camera B), both steps are executed independently for each set (see Fig. 7).

The image preprocessing step comprises the application of the techniques described in Section 3.1. Therefore, for each camera set, these transformations are independently applied. This includes the color plane decomposition (red, green, blue, yellow, cyan, and magenta channels), the grayscale transformation, the gamma correction with parameter optimization ($\gamma \in \{0.1, 0.2, 0.5, 1.5, 2.0, 3.0, 4.0, 5.0\}$), the LBP texture descriptor, and the RGB-HOG transformation. Consequently, a new set of transformed images is generated for each preprocessing technique. All of these, together with the original set, will be used as input in the second step.

For the fine-tuning step, a variety of DL models are selected. First, we have included several CNN architectures, from the classical proposals to more efficient or recent ones, including ResNet-50 (He et al., 2016), SqueezeNet (Iandola et al., 2016), EfficientNet (Tan and Le, 2019), ResNeXt-50 (Xie et al., 2017), ConvNeXt-S and ConvNeXt-L (Liu et al., 2022). Then, some ViTs are also considered, including the

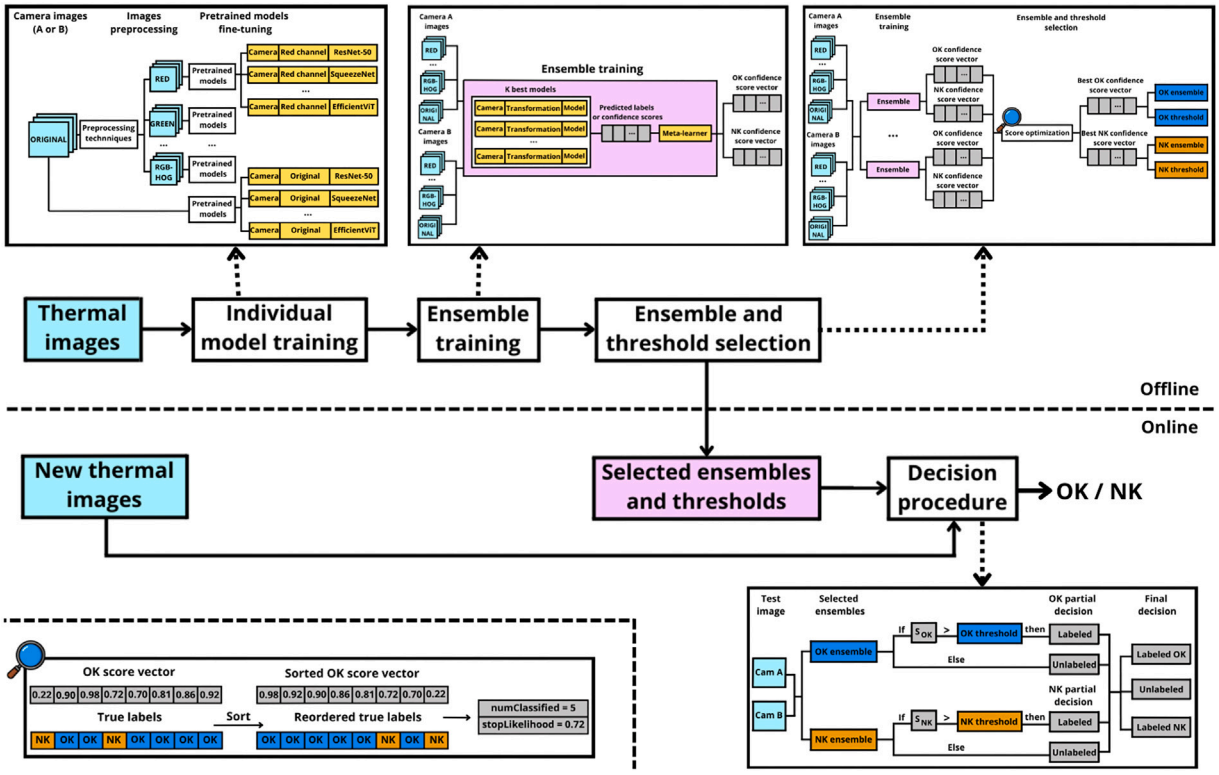


Fig. 6. High-level view of the proposed architecture. Each block is described in the following subsections and can be examined in detail in Figs. 7–11.

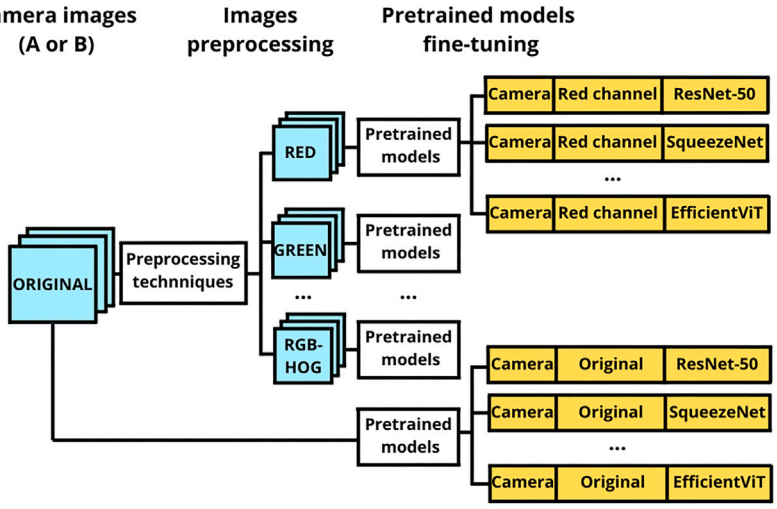


Fig. 7. Individual model training.

original ViT, a hybrid ResNet-ViT (Dosovitskiy et al., 2021), EVA-02 (Fang et al., 2024), and EfficientViT (Cai et al., 2023). For each architecture, a pretrained model fitted on the well-known Imagenet dataset (Deng et al., 2009) is selected. The objective is to fine-tune a model using both transformed and original images for each architecture. To prevent overfitting and reduce computational time, an EarlyStopping mechanism is employed. This form of regularization stops the training process when the validation loss does not demonstrate improvement after a consecutive number of epochs (we considered 20 for this work). At that point, the procedure is concluded and the weights of the epoch presenting the best F1 score on the validation subset are saved. Consequently, the output of the individual model training stage is $(numPrepTechniques + original) \cdot numPretrModels \cdot numCameras = 11 \cdot 10 \cdot 2 = 220$ fine-tuned models.

6.2. Ensemble training

To improve the performance of the individual models, we propose an ensemble architecture. To fully leverage the potential of these models, which were fine-tuned using the cleaned dataset, the most suitable ensemble learning method is stacking. Fig. 8 illustrates this stage, which is composed of three steps.

The initial step is the selection of the K best individual models from the set of models obtained in the previous stage (see Section 6.1). This decision is based on two criteria. The first is the F2 score metric achieved over the validation set, which prioritizes the reduction of false negatives over false positives. Furthermore, it is mandatory to include at least one model from each camera. Otherwise, the classification of OK parts will be incomplete, as it would entail determining a part

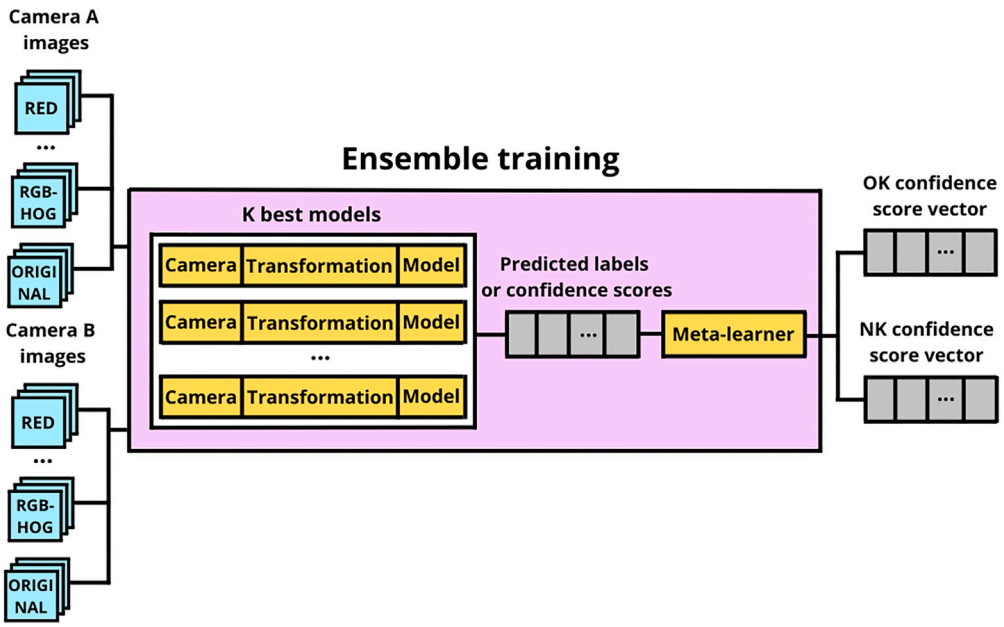


Fig. 8. Detail of the ensemble training stage of the proposed architecture.

as non-defective without considering both sides of the mold, which represent two distinct sides of the part.

The second step involves generating labels and confidence scores as outputs by these models. The concept of a confidence score is a pivotal aspect of this proposal. It denotes a value in the range $[0, 1]$, representing the likelihood that an instance belongs to a specific class. In the context of our binary classification problem, the model will generate a two-dimensional vector where each position is a confidence score, and their sum equals 1. This is, if we denote the score value associated with the NK class as s_{NK}, and the OK score value for the OK class as s_{OK}, then $s_{OK} = 1 - s_{NK}$. The largest value in the confidence score vector is interpreted as the predicted class. Therefore, in addition to generating a predicted label, the model also returns a confidence score associated with it. Hence, a forward pass of the images will be conducted in this step for the K individual models to generate both predicted labels and confidence score vectors.

The third step is the meta-learner training. For that end, the labels and confidence scores generated in the second step are used as input for a simple model. We try different models, including Regularized Linear Regression (RLR), Naive Bayes (NB), Support Vector Machine (SVM), and Multilayer Perceptron (MLP). Moreover, simple voting schemes are also evaluated, using predicted labels for maximum voting (MV), and confidence scores for average (AV) and weighted average voting (WAV). For the remaining meta-learners that accept both types of input, two independent experiments are conducted. On the one hand, the positive confidence values, s_{NK}, obtained with the K models are used as input. On the other hand, the predicted labels (the class with the largest confidence score) are employed as input. Consequently, the number of meta-learners trained is 11. In each case, the meta-learner model is fitted with the outputs generated by the K models over the training set, either in the form of labels or confidence scores. Moreover, some hyperparameters are fitted.

- In MLP, the number and size of the hidden units are optimized. Moreover, the solver and the activation function are also adjusted.
- In SVM, the regularization parameter, the kernel, and the degree of the polynomial kernel function are optimized.
- In RLR, the penalty parameter is adjusted between L1, L2, and elasticnet, as well as without any penalty term. In addition, the inverse of the regularization strength, the solver, and the weights associated with classes (which may be balanced or imbalanced) are also adjusted.
- In NB, several algorithms are tested: the Gaussian and Categorical Naive Bayes algorithms.

The outputs of these ensembles will be used in the next stage to enhance the classification results, specifically by having the validation set pass through the fitted meta-learner. The output of each model for each image will be a confidence score vector with two elements, s_{OK} and s_{NK}, which is processed to construct two complementary vectors. On the one hand, the confidence score vector for the OK class is created, containing in the position i the confidence score s_{OK} for the validation image i. On the other hand, the confidence score vector for the NK class is generated in an analogous manner.

6.3. Ensemble and threshold selection

The third stage of the proposal is the ensemble and threshold selection. The objective is to classify parts only when the prediction confidence score is high enough. To this end, a specific model for OK classification (referred to as the OK ensemble) and a specific model for NK classification (referred to as the NK ensemble) will be selected, in conjunction with two thresholds to establish the minimum confidence score required. The threshold for classification as OK (referred to as the OK threshold) represents the minimum confidence score that a part must have to be considered correct. Similarly, the threshold for classification as NK (referred to as the NK threshold) represents the minimum confidence score that a part must have to be considered defective. Therefore, a total part classification is not performed, and several parts will remain unlabeled. Fig. 9 illustrates this stage.

The stage consists of a vector search algorithm, which takes as input all the confidence score vectors obtained in the previous ensemble training stage, along with the actual labels of the associated parts.

For each of the OK confidence score vectors, the algorithm follows these steps, which are exemplified in Fig. 10:

1. First, the likelihood values are sorted in a descending order, resulting also in the reordering of the associated true labels.
2. The number of correctly classified instances is counted from left to right. Since OK likelihood vectors are being considered:
    - If the actual label of the part is OK, the image i will be correctly classified if its associated score value, $s_{OK}(i)$, is

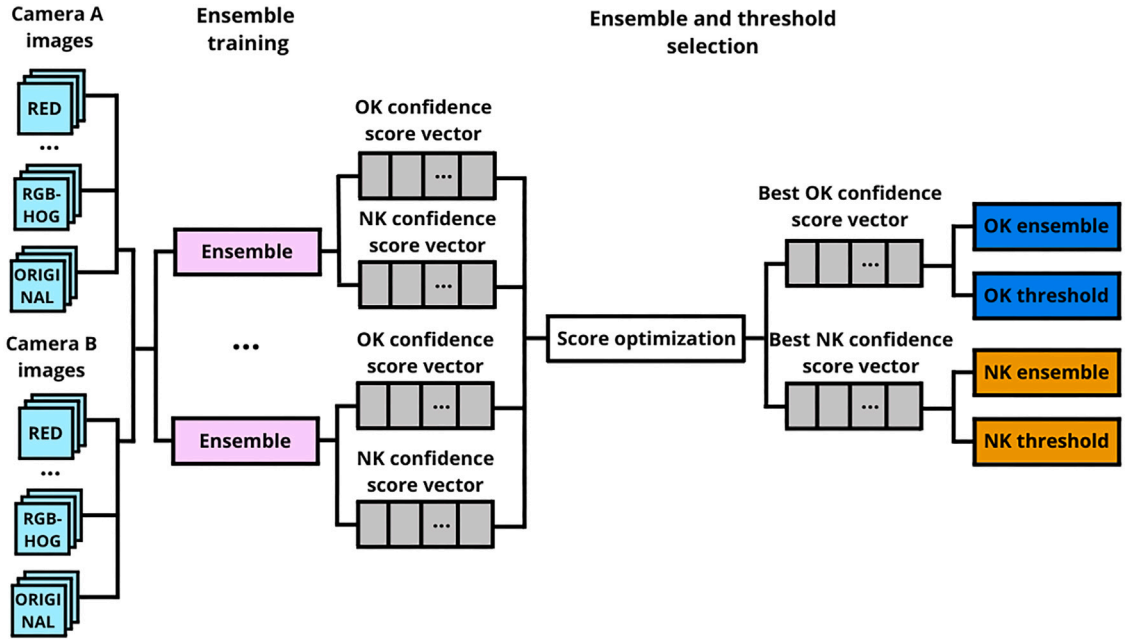


Fig. 9. Detail of the ensemble and threshold selection stage of the proposed architecture.

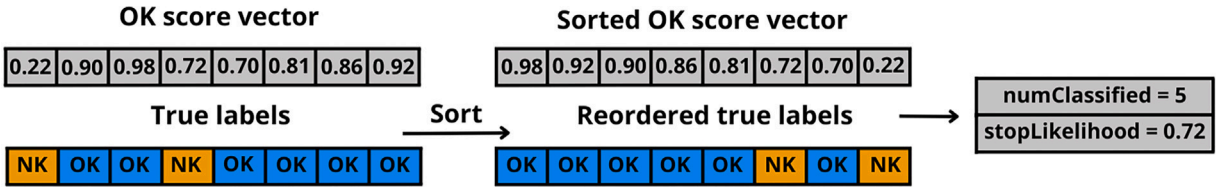


Fig. 10. Algorithm exemplification.

greater than or equal to 0.5. Otherwise, the image will be incorrectly classified.

- If the actual label of the part is NK, the image i will be correctly classified if its associated score value, $s_{OK}(i)$, is less than 0.5. Otherwise, the image will be incorrectly classified.

3. When the first error is committed, the algorithm stops, and the number of elements that have been correctly classified up to that point, denoted as numClassified, is recorded. Additionally, the likelihood associated with the first committed error, denoted as stopLikelihood, is stored.

The aforementioned process is executed for each OK score vector generated by the different ensemble combinations. Subsequently, the vector that has achieved the highest value for numOKclassified within the algorithmic process is selected. This represents the best ensemble for classifying OK images (denoted as the OK ensemble). Furthermore, the associated threshold is set to the first error score value committed in their likelihood vector, which is stopLikelihood.

The algorithm is executed in an analogous manner over the NK score vectors, resulting in the NK ensemble and the NK threshold. Consequently, the output of this stage is composed of two ensembles (the OK ensemble and the NK ensemble), as well as a threshold associated with each ensemble (the OK threshold and the NK threshold).

6.4. Decision procedure

This final stage, illustrated in Fig. 11, presents the procedure for making predictions with the test set. To this end, the two selected ensembles from the previous stage, together with their associated thresholds, are employed. Given a test instance, the likelihood of belonging to the OK class and the likelihood of belonging to the NK class are computed in parallel with each ensemble. This is, the OK ensemble provides the OK confidence score, s_{OK}, and the NK ensemble generates the NK confidence score, s_{NK}. Subsequently, local decisions are made based on both values as follows:

- If the s_{OK} value is greater than the OK threshold, the instance will be classified as labeled in the OK partial decision. Otherwise, the instance will remain unlabeled.
- If the s_{NK} value is greater than the NK threshold, the instance will be classified as labeled in the NK partial decision. Otherwise, the instance will remain unlabeled.

Once partial decisions have been made, the final predicted label is assigned, as outlined in Table 2, according to the XOR gate. Thus, if either the OK partial decision or the NK partial decision is labeled, the final decision will be to label the instance. In this case, the final predicted label will be the one assigned by the partial decision. Otherwise, the instance will remain unlabeled.

Table 2
Final decision.

OK partial decision	NK partial decision	Final predicted label
Labeled	Unlabeled	Labeled OK
Labeled	Labeled	Unlabeled
Unlabeled	Unlabeled	Unlabeled
Unlabeled	Labeled	Labeled NK

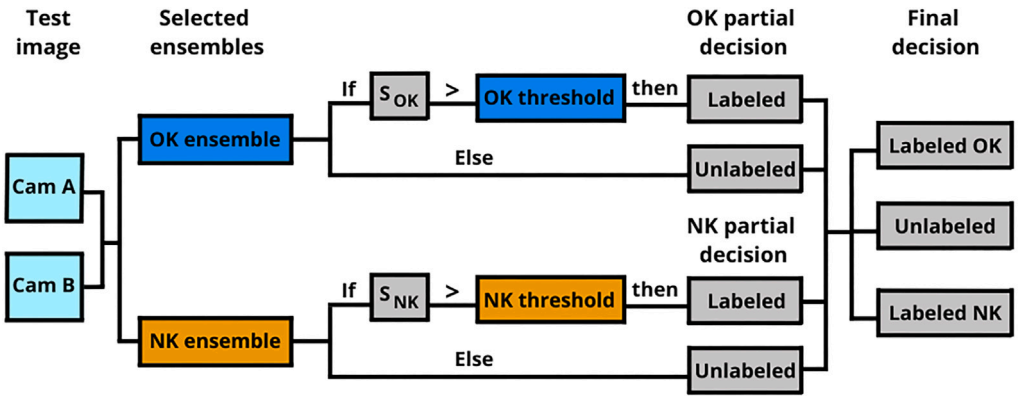


Fig. 11. Inference process.

Table 3
Results for the top five models. M1 denotes the best-performing model, M2 the second best, and so on.

Model	Camera	Prep. Tech.	DL Model	TP	FP	FN	TN	F2 score
M1	Cam A	Magenta ch.	ConvNeXt-S	30	65	3	90	**0.661**
M2	Cam B	RGB-HOG	ConvNeXt-S	27	59	6	96	**0.619**
M3	Cam A	RGB-HOG	EVA-02	30	83	3	72	**0.612**
M4	Cam B	Grayscale	ConvNeXt-S	28	72	5	83	**0.603**
M5	Cam B	Cyan ch.	ResNet-ViT	23	36	10	119	**0.602**

Table 4
Complementary metrics for the top five models.

Model	Accuracy	F1 score	TP rate	FP rate	FN rate	TN rate
M1	0.638	0.469	0.909	0.419	0.091	0.581
M2	0.654	0.454	0.818	0.381	0.182	0.619
M3	0.543	0.411	0.909	0.535	0.091	0.465
M4	0.590	0.421	0.848	0.465	0.152	0.535
M5	0.755	0.500	0.697	0.232	0.303	0.768

7. Results and discussion

This section presents the experimental results and discussion. The results are organized into three different subsections, which correspond to the offline stages identified in Fig. 6, namely, individual models results, ensemble learning architecture results, and ensemble and threshold selection results. Furthermore, the proposed method is compared with other architectures, such as autoencoders and a contrastive learning approach, in the final subsection. The experiments were conducted on an Intel Xeon Silver 4310 CPU at 2.10 GHz with 32 GB RAM. Furthermore, an NVIDIA A40 GPU with 48 GB of RAM is employed to enhance computational efficiency. The model optimization is implemented using the Fast.ai framework[3] (version 2.7.13). Additionally, other libraries are employed, such as PyTorch (version 1.13.1) or Scikit-Learn (version 1.0.2).

7.1. Individual models results

This subsection presents the results of the individual models on the test subset, which serves as the baseline for comparing the results produced by the proposed architecture. As previously stated, the selected metric for evaluation is the F2 score. The positive class corresponds to defective parts. Therefore, TP denotes defective parts correctly identified as defective, while TN denotes normal parts correctly identified as normal. Table 3 presents the top five model results, and Table 4 provides complementary metrics. Fig. 14(a) further illustrates these metrics in a bar chart for clarity.

The results suggest that both cameras exhibit comparable performance in the classification tasks. The preprocessing techniques appear to be diverse, with three techniques employing a single channel and two techniques using two channels. The most effective models are ConvNeXt and ViTs, which represent the current state of the art in computer vision. However, the results obtained with individual models are not satisfactory, as the number of false negatives is not sufficiently low to make their use feasible in a real industrial environment. Furthermore, a considerable number of FPs are present in every model, indicating the misclassification of non-defective parts that subsequently melt down. All of these observations are also reflected in the complementary metrics. Intuitively, these poor results can be attributed to the use of a single camera image. It is illogical to predict an image as OK processing just one side of the mold. The ensemble results, which combine both perspectives of the mold, will be discussed in the following section.

7.2. Ensemble learning results

In the ensemble results, the number of individual models, K, used in the architecture varies from 2 to 20. Models incorporating all possible combinations of the K value, the meta-learner model, and the input type are trained. Once again, the five best models are presented. Table 5 outlines each model, with its characteristics and evaluation metrics, while Table 6 presents the complementary metrics. Once again, Fig. 14(b) provides a visual representation of these results in a bar chart.

The F2 score results suggest that the optimal number of models included in the ensemble architecture is between 15 and 18. Moreover, the AV and WAV methods demonstrate the best performance, while confidence scores constitute the most effective input type. The results achieve a slight improvement in terms of F2 score and FN rate when

[3] https://www.fast.ai/

Table 5
Results for the top five ensembles. E1 denotes the best-performing ensemble, E2 the second best, and so on.

Ensemble	K	Ensemble method	Input type	TP	FP	FN	TN	F2 score
E1	15	WAV	Confidence scores	32	62	1	93	**0.708**
E2	15	AV	Confidence scores	32	68	1	87	**0.690**
E3	18	WAV	Confidence scores	32	68	1	87	**0.690**
E4	8	WAV	Confidence scores	32	70	1	85	**0.684**
E5	18	MV	Labels	33	78	0	77	**0.679**

Table 6
Complementary metrics for the top five ensembles.

Ensemble	Accuracy	F1 score	TP rate	FP rate	FN rate	TN rate
E1	0.670	0.500	0.970	0.400	0.030	0.600
E2	0.647	0.485	0.970	0.439	0.030	0.561
E3	0.647	0.485	0.970	0.439	0.030	0.561
E4	0.635	0.478	0.970	0.452	0.030	0.548
E5	0.587	0.458	1.000	0.503	0.000	0.497

compared to individual models. However, to further enhance these results, ensemble and threshold selection may prove beneficial for performing a partial classification by enabling decisions to be made only when confidence levels are high.

7.3. Ensemble and threshold selection results

This final subsection presents the results of the entire proposed architecture. As previously stated in Section 6.3, the vector search algorithm takes, for each ensemble, a confidence score vector and creates two parameters: the number of elements that have been correctly classified before the first error and the likelihood associated with the first committed error. This process can be represented graphically using a bar chart. Fig. 12 illustrates two examples of the decision made by the algorithm, one for an OK score vector and another for an NK score vector.

In the left-hand case, 49 instances were correctly classified before the first erroneous categorization, whereas in the second case, this number decreased to 7 instances. Once the algorithm has identified the optimal ensembles and thresholds, the final classification is performed. Table 7 presents the results of the five best architectures, together with those of the previous approach (denoted as PA (Mielgo et al., 2024)). It also reports the labeling rate, defined as the ratio between the number of labeled instances and the total number of instances. To enhance comprehension, Figs. 13(a) and 13(b) compare the F2 score metric and the labeling rate using a bar chart and a scatter plot for these five architectures. Examining the results, it is evident that introducing thresholds to limit classification to high-confidence predictions has yielded a significant improvement in the F2 score. In particular, the A1 and A3 architectures are capable of correctly classifying all the labeled samples. However, they only label 11.2% and 5.9% of the instances, respectively. On the other hand, the A3 model is capable of classifying 59% of the parts, and also presents favorable F2 score results. Hence, the A3 model yields the best results, as is also evident in Fig. 13(b), where the A3 model is located in the upper right quadrant, indicating that it optimizes the balance between both metrics. Moreover, when compared with the results of the previous approach (Mielgo et al., 2024), it yields a threefold increase in the number of labeled instances and outperforms it in terms of the F2 score.

Table 8 details the characteristics of each ensemble, including the selected meta-learners and thresholds for the OK and the NK partial classification, while Table 9 includes complementary metrics. Fig. 14(c) presents a graphical representation of these results in a bar chart. Examining these tables, we observe that the ensemble methods selected by the vector search algorithm are primarily SVM for the OK ensemble, and WAV and AV for the NK ensemble. The selected input type appears to be dependent on the classification task: the meta-learner specialized in classifying OK tends to use labels, while the meta-learner specialized in classifying NK tends to use confidence scores. A similar pattern is observed with thresholds, where the OK threshold value is consistently near 1, while the NK threshold exhibits greater variability in its values. The complementary metrics on both models exhibit high values for the TP and TN rates and low values for the FN and FP rates.

Finally, Figs. 14 and 15 provide a graphical comparison between the baseline models (M1–M5 and E1–E5) and the proposed models (A1–A5). The bar charts summarize the F1 and F2 scores, as well as the accuracy, while the heatmap also includes the FN rate and the labeling rate. As can be observed, the proposed architectures consistently outperform the baseline models in terms of accuracy and in their F1 and F2 score metrics. Although the introduction of thresholds significantly decreases the labeling rate for some architectures (particularly A1, A2, A4, and A5), this trade-off results in a notable improvement in overall classification performance. Furthermore, the FN rates of these models remain comparable to those of the E1–E5 architectures, supporting the robustness of the proposed approach in correctly identifying positive instances.

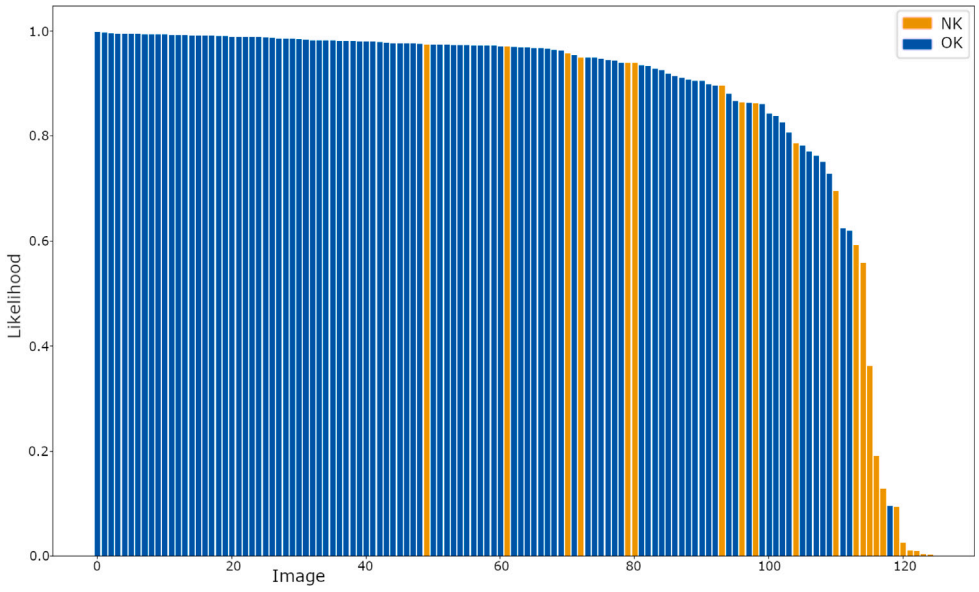
7.4. Comparative evaluation

To compare the performance of the proposed architecture against other integration techniques, three additional experiments were conducted. First, several high-performing boosting ensembles were evaluated to compare their efficiency against the stacking process. Second, alternative methods that modify the objective function and the fusion level were explored to complete the comparison. Specifically, the effectiveness of contrastive learning and autoencoders was assessed. These results provide a more comprehensive evaluation of the proposed approach.

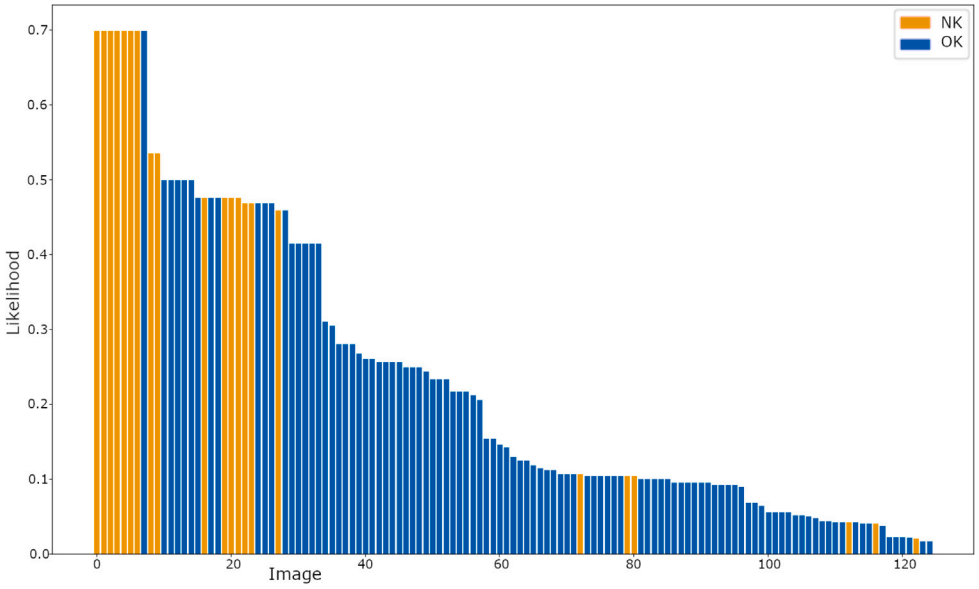
7.4.1. Boosting ensemble

Several experiments were conducted replacing the stacking ensemble with a boosting architecture. In particular, XGBoost, AdaBoost, LightGBM, and CatBoost were employed to analyze the impact of boosting on predictive performance. First, the boosting strategy was applied using the confidence scores from the individual models. However, the achieved results were significantly worse than the baseline performance, as can be seen in Table 10. As this is not the standard way to use boosting ensembles, an additional experiment was conducted in which individual models were used as feature extractors by removing their final layer. In this way, the boosting ensemble takes all extracted features as input. Results are presented in Table 11.

Once again, results were worse than the baseline. Given this, the boosting strategy was confirmed not to improve the stacking ensemble performance.



(a) OK score vector



(b) NK score vector

Fig. 12. Algorithm bar charts.

Table 7
Results for the top five models of the proposed architecture and their comparison with the previous approach. A1 denotes the best-performing architecture, A2 the second best, and so on.

Arch.	TP	FP	FN	TN	Unlabeled NK	Unlabeled OK	Labeling rate	F2 score
A1	10	0	0	11	23	144	0.112	**1**
A2	2	0	0	9	31	146	0.059	**1**
A3	12	0	1	98	20	57	0.590	**0.938**
A4	9	0	1	18	23	137	0.149	**0.918**
A5	9	0	1	15	23	140	0.133	**0.918**
PA (Mielgo et al., 2024)	5	5	1	30	27	120	0.218	**0.735**

7.4.2. Contrastive learning architecture

A contrastive learning approach, previously employed in anomaly detection with satisfactory results (Kopuklu et al., 2021; Mielgo et al., 2025), was selected to simultaneously explore a different loss function and a mid-level fusion of images from both cameras. Contrastive learning is a machine learning technique designed to obtain more structured representations of data. It relies on a contrastive loss function that maximizes the distance between negative pairs and minimizes the

Table 8
Parameters for the top five architectural results.

Arch.	K	OK E. method	NK E. method	OK input t.	NK input t.	OK thr.	NK thr.
A1	12	SVM	WAV	Conf. scores	Conf. scores	0.999	0.729
A2	9	SVM	SVM	Labels	Conf. scores	0.995	0.997
A3	15	MLP	AV	Labels	Conf. scores	0.999	0.722
A4	14	SVM	WAV	Labels	Conf. scores	0.999	0.734
A5	20	SVM	AV	Labels	Conf. scores	0.999	0.738

Table 9
Complementary metrics for the top five models of the proposed architecture and their comparison with the previous approach.

Arch.	Accuracy	F1 score	TP rate	FP rate	FN rate	TN rate
A1	1	1	1	0	0	1
A2	1	1	1	0	0	1
A3	0.991	0.960	0.923	0	0.078	1
A4	0.964	0.947	0.9	0	0.1	1
A5	0.96	0.947	0.9	0	0.1	1
PA (Mielgo et al., 2024)	0.854	0.625	0.833	0.143	0.167	0.857

Table 10
Results for the top five boosting strategies using confidence scores as input. B1 denotes the best-performing ensemble, B2 the second best, and so on.

Architecture	Boosting strategy	K	TP	FP	FN	TN	F2 score
B1	AdaBoost	17	13	7	20	148	**0.428**
B2	AdaBoost	16	13	9	20	146	**0.422**
B3	AdaBoost	15	12	9	21	146	**0.392**
B4	XGBoost	16	11	2	22	153	**0.379**
B5	XGBoost	17	11	2	22	153	**0.379**

Table 11
Results for the top five boosting strategies using features as input. B6 denotes the best-performing ensemble, B7 the second best, and so on.

Architecture	Boosting strategy	K	TP	FP	FN	TN	F2 score
B6	AdaBoost	6	7	3	26	152	**0.246**
B7	AdaBoost	7	7	6	26	149	**0.241**
B8	AdaBoost	5	7	7	26	148	**0.240**
B9	LightGBM	10	6	0	27	155	**0.217**
B10	LightGBM	12	6	0	27	155	**0.217**

distance between positive pairs. In our setting, positive pairs consist of normal instances, whereas negative pairs combine a normal instance with a defective one. As a result, defective samples should be positioned far from normal ones in the latent space, thereby facilitating the classification task.

A contrastive loss function oriented toward anomaly detection (Kopuklu et al., 2021) was selected. To this end, the training batches were constructed to contain a proportional number of elements from both classes, reflecting the dataset distribution. Let $n_r,\ r \in 1, 2, \ldots, N$ denote the normal instances in a batch, with N the total number of normal instances, and $d_s,\ s \in 1, 2, \ldots, D$ denote the defective instances in the batch, with D the total number of defective instances. Then, the contrastive loss function is given by Eq. (2).

$$\mathcal{L}_{i,j} = -\log \frac{\exp\left(sim(n_i, n_j)/\tau\right)}{\exp\left(sim(n_i, n_j)/\tau\right) + \sum_{k=1}^{D} \exp\left(sim(n_i, d_k)/\tau\right)} \tag{1}$$

$$\mathcal{L} = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbb{1}_{i \neq j} \mathcal{L}_{i,j} \tag{2}$$

where $sim(\cdot, \cdot)$ denotes the cosine similarity function between two vectors, τ is a positive temperature parameter, and $\mathbb{1}$ is the indicator function.

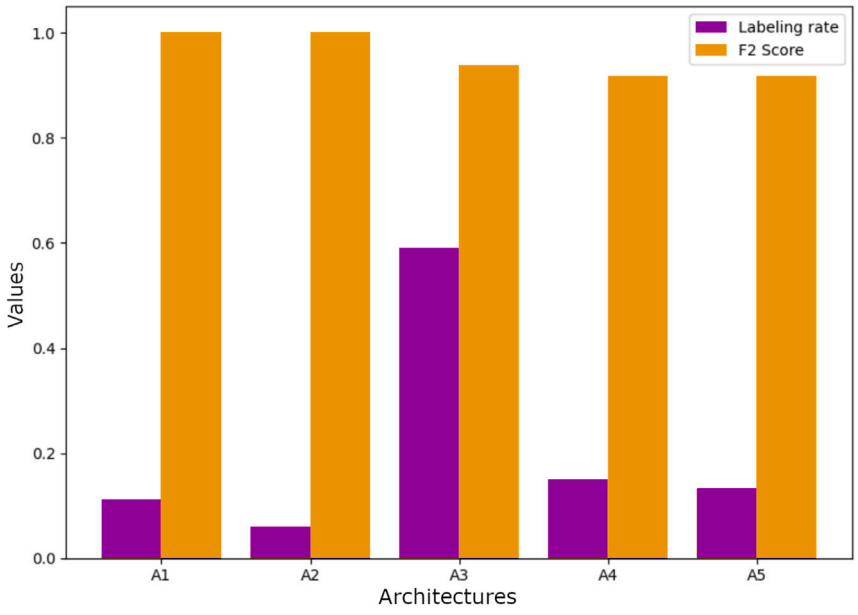
To effectively integrate information from both cameras, we first used two pretrained DL models as feature extractors and then merged the outputs through concatenation. Contrastive loss was subsequently applied to shape the representation space, followed by a projection head that performed the final classification. As feature extractors, we employed the same CNNs and ViTs considered in the main proposal, applying identical preprocessing techniques for consistency. The projection head consisted of two fully connected layers with a ReLU activation applied between them. The training, validation, and test sets were identical to those in the main proposal. The training procedure was also similar, including the EarlyStopping mechanism based on the validation loss, with the temperature parameter τ fixed at 0.1. The results of the five best models are presented in Table 12.

The results demonstrate the limited performance of the method, particularly when compared to the baseline models, which achieve an F2 score of 0.661.

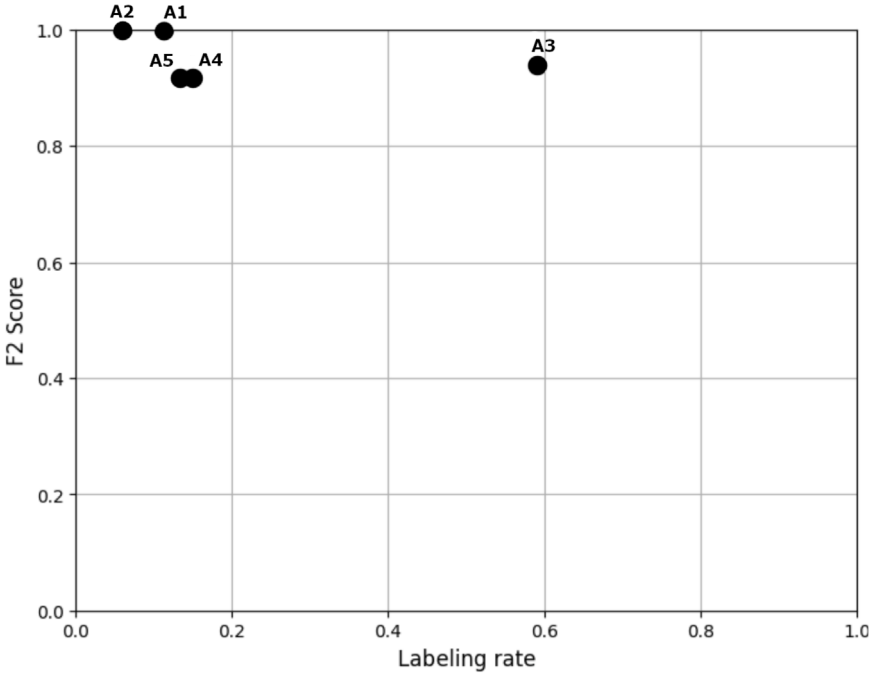
7.4.3. Autoencoder

Finally, an autoencoder approach is evaluated to enhance the completeness of the comparison with a semi-supervised method. An autoencoder is a neural network architecture designed to learn meaningful data representations through reconstruction. It consists of an encoder, which maps the input into a latent representation, and a decoder, which reconstructs the original input from this representation. The loss function penalizes discrepancies between the input and its reconstruction. In most cases, the latent space has a lower dimensionality than the input, which encourages the network to preserve the most relevant features in a compact representation.

For evaluation, the autoencoder architecture is trained on non-defective instances to model its distribution in the latent space (Tsai and Jen, 2021). Specifically, two autoencoders are trained in parallel, one for each camera. The validation set, which includes both



(a) Bar chart



(b) Scatter plot

Fig. 13. Comparison of F2 score and labeling rate for models based on the proposed architecture.

Table 12
Results for the top five models using the contrastive learning approach. CL1 denotes the best-performing model, CL2 the second best, and so on.

Model	Prep. Tech.	DL Model	TP	FP	FN	TN	F2 score
CL1	Gamma 0.5	SqueezeNet	7	10	26	145	**0.235**
CL2	Gamma 0.5	ResNet-50	6	6	27	149	**0.208**
CL3	Red ch.	SqueezeNet	6	7	27	148	**0.207**
CL4	Red ch.	ResNeXt-50	4	8	29	147	**0.139**
CL5	Green ch.	Resnet-50	3	1	30	154	**0.110**

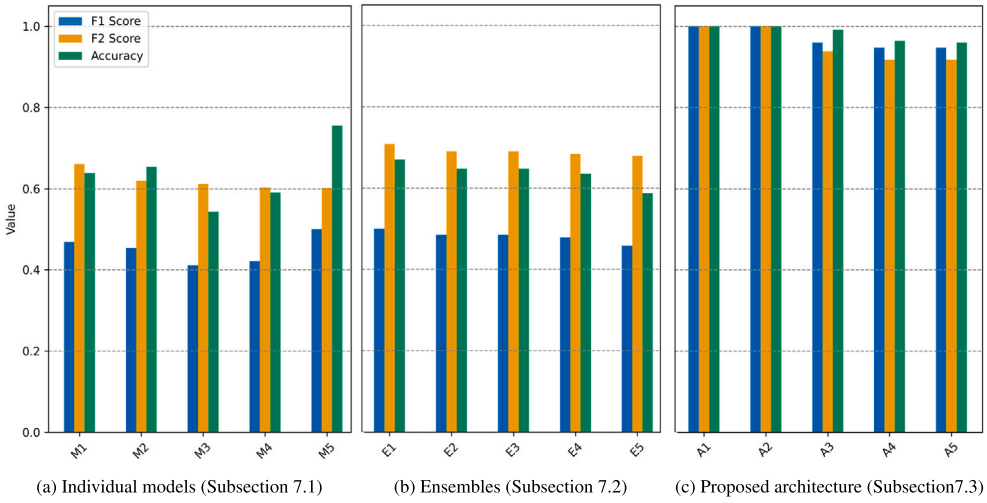
defective and non-defective instances, is then processed through the autoencoders to determine a threshold based on the reconstruction error. Following the same procedure adopted in the proposed method, instances are sorted by their reconstruction error, and the threshold is set to the error value corresponding to the first defective image. For the inference stage, an instance is classified as defective if the

Table 13
Results for the top five autoencoders. AE1 denotes the best-performing model, AE2 the second best, and so on.

Model	Prep. Tech.	Autoencoder	LS dim	TP	FP	FN	TN	F2 score
AE1	Gamma 2.0	VAE	64	33	152	0	3	**0.521**
AE2	Blue ch.	CVAE	256	33	153	0	2	**0.519**
AE3	Grayscale	VAE	32	33	154	0	1	**0.517**
AE4	Original	VAE	256	33	154	0	1	**0.517**
AE5	Gamma 1.5	CVAE	64	33	154	0	1	**0.517**

Table 14
Best five autoencoder results obtained with tolerance levels of 15% and 30%. AET1 denotes the best-performing model, AET2 the second best, and so on.

Model	Prep. Tech.	Autoencoder	LS dim	Tolerance	TP	FP	FN	TN	F2 score
AET1	Yellow ch.	VAE	32	30%	111	152	3	44	**0.549**
AET2	Yellow ch.	VAE	32	15%	32	134	1	21	**0.537**
AET3	Red ch.	CVAE	32	15%	33	145	0	10	**0.532**
AET4	Gamma 0.1	VAE	64	15%	33	145	0	10	**0.532**
AET5	Gamma 5.0	VAE	32	30%	29	113	4	42	**0.529**



(a) Individual models (Subsection 7.1) (b) Ensembles (Subsection 7.2) (c) Proposed architecture (Subsection7.3)

Fig. 14. Performance comparison of all models based on F1 score, F2 score, and accuracy.

reconstruction error of at least one autoencoder exceeds its threshold. Multiple autoencoder variants were evaluated in combination with all preprocessing techniques, including sparse autoencoder (SAE), denoising autoencoder (DAE), convolutional autoencoder (CAE), variational autoencoder (VAE), and convolutional variational autoencoder (CVAE). For consistency, network hyperparameters were set to values similar to those in the proposed method. The experiments were conducted with latent space dimensions of 32, 64, and 256. Table 13 presents the results.

The results show a lower performance of this approach compared with both the proposed method and the baseline. The reduced number of FN and TN instances might suggest that a very strict threshold was applied, potentially influenced by an outlier in the validation set. Therefore, the experiments were repeated, setting the thresholds with a tolerance of 15% and 30% for defective instances misclassified in the validation set. The results are presented in Table 14.

The selection of the threshold with the tolerance parameter was found to help the models achieve slightly better results; however, their performance remains worse than that of the proposed approach. Analyzing the reconstruction error distribution in the test set for both cameras (Fig. 16 shows the values for the AET1 model), it can be observed that there are no significant differences between the defective and non-defective classes. Therefore, it is unlikely that a more suitable threshold can be identified.

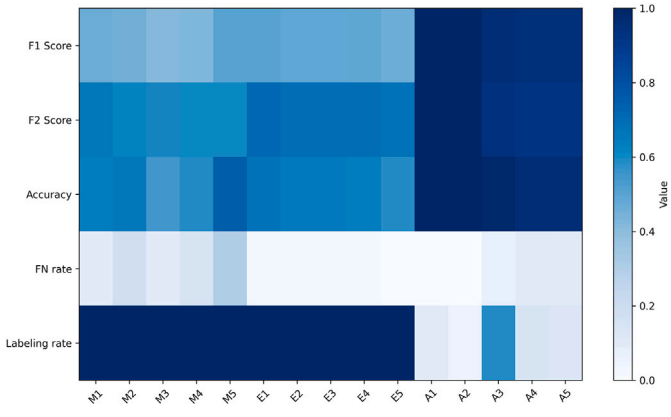
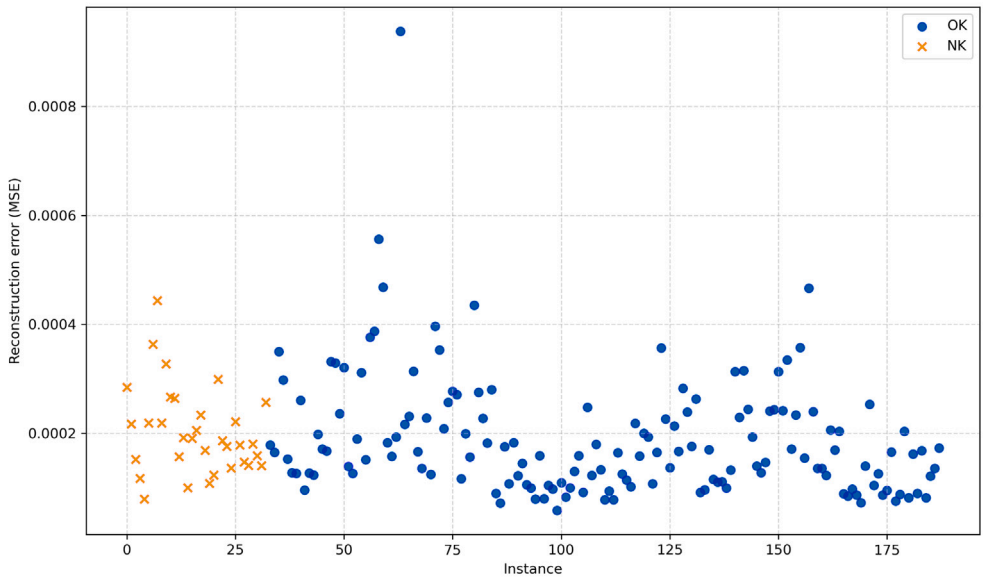


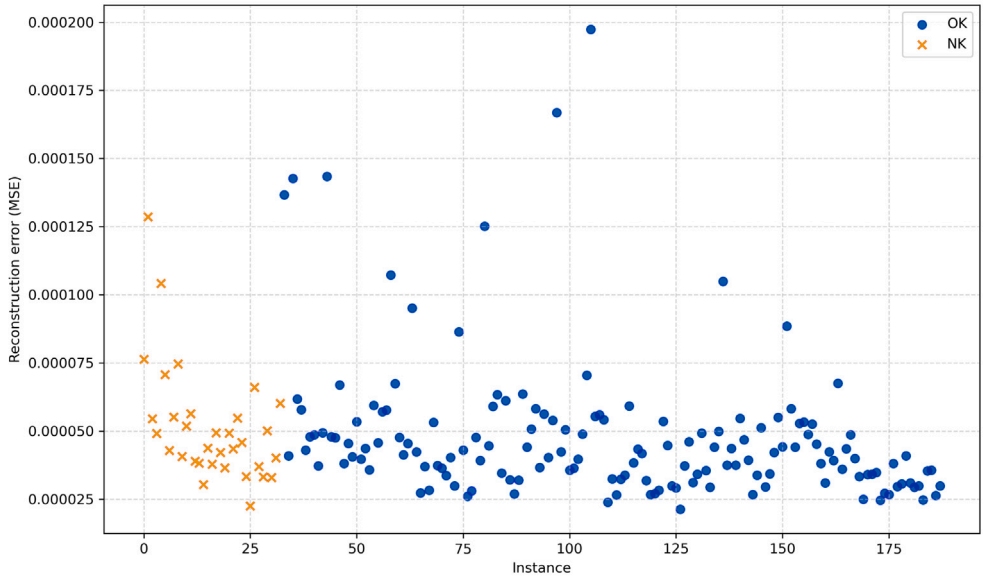
Fig. 15. Heatmap comparing evaluation metrics across models.

7.5. Computational efficiency

This work proposes an effective architecture for quality control in engine-block manufacturing through die casting, aiming to reduce the bottleneck in the leak test. Since the die-casting process produces an engine block every 90 s, computational complexity is not a constraint in this context. However, in another industrial scenario, it could be



(a) Camera A



(b) Camera B

Fig. 16. Reconstruction error for test set instances of the AET1 model.

crucial. Therefore, the computational efficiency of the developed models is analyzed. To this end, the average inference time per instance is recorded, together with the number of parameters. Both metrics are computed over all the presented models (M1–M5, E1–E5, A1–A5) and summarized in Table 15. The inference process was conducted on an Intel Xeon Silver 4310 CPU with an NVIDIA A40 GPU with 48 GB of RAM. As classification performance was not evaluated in this experiment, 1000 random images from the intersected dataset were selected, allowing repetitions if necessary. For each model, five independent experiments were conducted and averaged to obtain representative, robust measurements. Furthermore, to accurately reflect the operating conditions of the system in the factory during real-time decision-making, model loading time was excluded from the measurements. Thus, the reported inference time includes both image loading and preprocessing, as well as the forward-pass inference computation. Inference was performed using a batch size of one.

For both metrics, the value is proportional to the number of individual models considered (K), with an average inference time per image of less than 0.360 s in all cases.

7.6. Limitations

Despite the promising results reported in this work, several limitations should be acknowledged. First, the dataset employed for evaluation was obtained from a single die-casting machine operating with a specific mold used to manufacture 4-cylinder engine blocks. While this reflects a realistic industrial deployment scenario, it constrains the immediate generalizability of the findings to other molds, machines, or component types. Variations in cooling dynamics, mold geometry, or production cadence may produce different thermal signatures. Therefore, although the proposed architecture is model-agnostic and

Table 15
Computational efficiency measurements and number of parameters per model.

Model	K	Average time per instance (s)	Number of parameters
M1	–	0.040	50.20M
M2	–	0.038	50.20M
M3	–	0.041	85.87M
M4	–	0.039	50.20M
M5	–	0.039	22.28M
E1	15	0.294	729.20M
E2	15	0.305	729.20M
E3	18	0.345	830.94M
E4	8	0.163	375.94M
E5	18	0.353	830.94M
A1	12	0.233	570.93M
A2	9	0.174	376.76M
A3	15	0.301	729.20M
A4	14	0.271	643.04M
A5	20	0.360	833.44M

adaptable, cross-machine validation remains an important direction for future work, as we discuss in the following section.

Second, the dataset size (1690 usable thermograms after intersection and cleaning) aligns with typical industrial data availability but is moderate by deep learning standards. Class imbalance was addressed through controlled undersampling, and the methodology includes several robustness-enhancing features, including multi-view modeling, diverse preprocessing pipelines, and meta-learning aggregation.

Third, image acquisition challenges inherent to high-throughput industrial environments introduce additional constraints. The synchronization issues described in Section 5.2 arise from real production conditions and were conservatively handled during dataset construction, as they occasionally led to partial occlusions (e.g., robotic arm interference) and minor thermal scale variations that required manual filtering. While the proposed architecture accommodates incomplete views via independent camera models and confidence-based decision thresholds (Section 6.3), fully automating these preprocessing steps remains an open challenge.

Finally, although inference latency is below 0.36 s per image, the current validation focuses on offline decision support rather than full closed-loop real-time integration. Practical deployment would require seamless synchronization between sensing, prediction, and actuation systems.

8. Conclusions and future work

The application of DL techniques enables the development of a classification model for the identification of defective and correct parts using thermographic images captured during the manufacturing process. The proposal combines the state-of-the-art DL models with classical preprocessing techniques and ensemble learning methods to enhance quality control in die casting processes. Our novel approach, which employs two different ensemble models, one for identifying OK parts and another for detecting NK parts, along with two thresholds, significantly improves the results of the previous approach.

The partial classification performed in the proposal provides promising results that would enable its implementation in a real manufacturing environment. On the one hand, a very strict model is obtained, which classifies a small number of images with total certainty. On the other hand, a slightly more relaxed model is attained, which classifies nearly two-thirds of parts with very high confidence. This approach minimizes the number of defective parts classified as OK, thanks to the use of the F2 score metric. Therefore, only one-third of the manufactured parts need to undergo the leak test, thereby improving factory productivity by reducing the bottleneck of additional testing.

An important aspect to consider for real-world deployment is how the proposed architecture would handle potential industrial instability and imperfect synchronization. As stated before, a manual data cleaning process was carried out to remove all the non-usable images caused mainly by synchronization issues with the robotic arm. However, that process cannot be performed in real time once the system is deployed. Furthermore, perfect synchronization between image capture and the injection process could not be assumed, as there is no direct signal from the die-casting machine indicating when to capture the thermographic images. The die casting machine is a black box that only the manufacturer can manipulate, so making ad-hoc programming for it is highly expensive. Thus, the only feasible solution is to use the signal of the start of the whole injection cycle using a fixed time for image capturing. As a future work, a YOLO-based model is planned to replace the current manual data cleaning process. The model will be used to extract regions of interest by identifying and drawing bounding boxes around the six faces present in the molds. This will allow the automatic detection of missing regions caused by robotic arm misalignment; in such cases, the corresponding images from both cameras will be discarded.

Beyond these deployment-related challenges, it is also important to consider the broader applicability of the architecture. Although the proposed approach is evaluated with a thermographic dataset in a die casting manufacturing of engine blocks, it can be straightforwardly extended to other quality control use cases. For a different binary image dataset with defective and non-defective classes, all the training processes described in Section 6 should be repeated. For individual model training, transfer learning can be considered by reusing models trained on the presented dataset. The subsequent ensemble training and ensemble/threshold selection stages should be performed as described in 6.2 and 6.3.

Future work will focus on processing images by regions of interest to study each portion individually. This approach will facilitate the reduction of the considerable complexity present in the image and also make models robust to changes in perspective and distance across images. Furthermore, it is planned to extend the experiments to additional molds and die-casting machines, as well as to other components like gearboxes, to achieve a broader validation of the proposed approach.

CRediT authorship contribution statement

Paula Mielgo: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation. **Anibal Bregon:** Writing – review & editing, Writing – original draft, Project administration, Methodology, Investigation, Conceptualization. **Carlos J. Alonso-González:** Writing – review & editing, Validation, Project administration. **Miguel A. Martínez-Prieto:** Writing – review & editing, Methodology. **Daniel López:** Writing – review & editing. **Belarmino Pulido:** Writing – review & editing, Validation.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used Grammarly and ChatGPT to improve the English language and readability of the manuscript. The tool was used solely for language editing purposes. After using this service, the author(s) reviewed and edited the content as needed and take full responsibility for the scientific content of the published article.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Daniel Lopez reports a relationship with HORSE Spain that includes: employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

P. Mielgo's work has been supported by the 2023 Call for Predoctoral Contracts granted by the University of Valladolid and Banco Santander. P. Mielgo, A. Bregon, C.J. Alonso-Gonzalez, M.A. Martinez-Prieto, and B. Pulido's work has been partially supported by Spanish Ministerio de Ciencia e Innovación under Grant PID2021-126659OB-I00. The authors acknowledge HORSE Powertrain for granting us access to the thermographic images dataset and Javier Moral Blanco for the help and support provided.

Data availability

The data that has been used is confidential.

References

Alber, M., Hönes, C., Baier, P., 2024. Evaluating vision transformer models for visual quality control in industrial manufacturing. In: Proc. Jt. Eur. Conf. Mach. Learn. Knowl. Discov. Databases. Springer, pp. 116–132. http://dx.doi.org/10.1007/978-3-031-70381-2_8.

Amigo, J.M., 2019. Hyperspectral and multispectral imaging: setting the scene. In: Data Handling in Science and Technology, vol. 32, Elsevier, pp. 3–16. http://dx.doi.org/10.1016/b978-0-444-63977-6.00001-8.

Babawale, S.D., Adebimpe, O.A., Oladokun, V.O., 2025. Evaluation of computer vision techniques for quality inspection in casting manufacturing process. Int. J. Adv. Manuf. Technol. 1–16. http://dx.doi.org/10.1007/s00170-025-16665-7.

Bahroun, S., Abed, R., Zagrouba, E., 2023. Deep 3D-LBP: CNN-based fusion of shape modeling and texture descriptors for accurate face recognition. Vis. Comput. 39 (1), 239–254. http://dx.doi.org/10.1007/s00371-021-02324-x.

Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., Steger, C., 2021. The MVTec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection. Int. J. Comput. Vis. 129 (4), 1038–1059. http://dx.doi.org/10.1007/s11263-020-01400-4.

Breiman, L., 1996. Bagging predictors. Mach. Learn. 24, 123–140. http://dx.doi.org/10.1007/bf00058655.

Buda, M., Maki, A., Mazurowski, M.A., 2018. A systematic study of the class imbalance problem in convolutional neural networks. Neural Netw. 106, 249–259. http://dx.doi.org/10.1016/j.neunet.2018.07.011.

Busch, M., Hausotte, T., 2022. Application of an edge detection algorithm for surface determination in industrial X-ray computed tomography. Prod. Eng. 16 (2), 411–422. http://dx.doi.org/10.1007/s11740-021-01100-z.

Cai, H., Li, J., Hu, M., Gan, C., Han, S., 2023. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In: Proc. IEEE/CVF Int. Conf. Comput. Vis.. pp. 17302–17313. http://dx.doi.org/10.1109/iccv51070.2023.01587.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A., 2014. Describing textures in the wild. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. CVPR, pp. 3606–3613. http://dx.doi.org/10.1109/cvpr.2014.461.

Cruz, S., Paulino, A., Duraes, J., Mendes, M., 2021. Real-time quality control of heat sealed bottles using thermal images and artificial neural network. J. Imaging 7 (2), 24. http://dx.doi.org/10.3390/jimaging7020024.

Dai, D., Wang, X., Zhang, Y., Zhao, L., Li, J., 2017. Leakage region detection of gas insulated equipment by applying infrared image processing technique. In: Proc. 9th Int. Conf. Meas. Technol. Mechatronics Autom.. ICMTMA, IEEE, pp. 94–98. http://dx.doi.org/10.1109/icmtma.2017.0030.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. CVPR, In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., vol. 1, IEEE, pp. 886–893. http://dx.doi.org/10.1109/cvpr.2005.177.

De Luca, R., Ferraro, A., Galli, A., Gallo, M., Moscato, V., Sperli, G., 2023. A deep attention based approach for predictive maintenance applications in IoT scenarios. J. Manuf. Technol. Manag. 34 (4), 535–556. http://dx.doi.org/10.1108/jmtm-02-2022-0093.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. CVPR, IEEE, pp. 248–255. http://dx.doi.org/10.1109/cvpr.2009.5206848.

Déniz, O., Bueno, G., Salido, J., De la Torre, F., 2011. Face recognition using histograms of oriented gradients. Pattern Recognit. 32 (12), 1598–1603. http://dx.doi.org/10.1016/j.patrec.2011.01.004.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2021. An image is worth 16x16 words: transformers for image recognition at scale. In: Proc. Int. Conf. Learn. Represent.. ICLR, http://dx.doi.org/10.48550/arXiv:2010.11929v2.

Duan, L., Yang, K., Ruan, L., 2020. Research on automatic recognition of casting defects based on deep learning. IEEE Access 9, 12209–12216. http://dx.doi.org/10.1109/access.2020.3048432.

Dyrmann, M., Karstoft, H., Midtiby, H.S., 2016. Plant species classification using deep convolutional neural network. Biosyst. Eng. 151, 72–80. http://dx.doi.org/10.1016/j.biosystemseng.2016.08.024.

Fan, Z., Sun, Y., 2019. Detecting and evaluation of fatigue damage in concrete with industrial computed tomography technology. Constr. Build. Mater. 223, 794–805. http://dx.doi.org/10.1016/j.conbuildmat.2019.07.016.

Fang, Y., Sun, Q., Wang, X., Huang, T., Wang, X., Cao, Y., 2024. EVA-02: A visual representation for neon genesis. Image Vis. Comput. 149, 105171. http://dx.doi.org/10.1016/j.imavis.2024.105171.

Freund, Y., Schapire, R., Abe, N., 1999. A short introduction to boosting. J. Jpn. Soc. Artif. Intell. 14 (1612), 771–780. http://dx.doi.org/10.11517/jjsai.14.5_771.

Gonzalez, R.C., Woods, R.E., 2008. Digital Image Processing. Pearson Education.

Grabowski, D., Cristalli, C., 2015. Production line quality control using infrared imaging. Infrared Phys. Technol. 71, 416–423. http://dx.doi.org/10.1016/j.infrared.2015.06.002.

Haff, R.P., Toyofuku, N., 2008. X-ray detection of defects and contaminants in the food industry. Sens. Instrum. Food Qual. Saf. 2, 262–273. http://dx.doi.org/10.1007/s11694-008-9059-8.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. CVPR, pp. 770–778. http://dx.doi.org/10.1109/cvpr.2016.90.

Hermann, M., Pentek, T., Otto, B., 2016. Design principles for Industrie 4.0 scenarios. In: Proc. 49th Hawaii Int. Conf. Syst. Sci.. HICSS, IEEE, pp. 3928–3937. http://dx.doi.org/10.1109/hicss.2016.488.

Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. Science 313 (5786), 504–507. http://dx.doi.org/10.1126/science.1127647.

Hsu, C.-C., 2007. The MOSUM of squares test for monitoring variance changes. Financ. Res. Lett. 4 (4), 254–260. http://dx.doi.org/10.1016/j.frl.2007.09.003.

Hsu, C.-M., Hsu, C.-C., Hsu, Z.-M., Shih, F.-Y., Chang, M.-L., Chen, T.-H., 2021. Colorectal polyp image detection and classification through grayscale images and deep learning. Sensors 21 (18), 5995. http://dx.doi.org/10.3390/s21185995.

Hu, C., Wang, Y., 2020. An efficient convolutional neural network model based on object-level attention mechanism for casting defect detection on radiography images. IEEE Trans. Ind. Electron. 67 (12), 10922–10930. http://dx.doi.org/10.1109/tie.2019.2962437.

Hu, C., Zhao, C., Shao, H., Deng, J., Wang, Y., 2024. TMFF: Trustworthy multi-focus fusion framework for multi-label sewer defect classification in sewer inspection videos. IEEE Trans. Circuits Syst. Video Technol. http://dx.doi.org/10.1109/tcsvt.2024.3433415.

Hussain, M., Chen, T., Hill, R., 2022. Moving toward smart manufacturing with an autonomous pallet racking inspection system based on MobileNetV2. J. Manuf. Mater. Process. 6 (4), 75. http://dx.doi.org/10.3390/jmmp6040075.

Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K., 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5mb model size. http://dx.doi.org/10.48550/arXiv.1602.07360v4, ArXiv Prepr..

Jack, K., 2011. Video Demystified: a Handbook for the Digital Engineer. Elsevier, http://dx.doi.org/10.1016/b978-0-7506-8395-1.x5000-7.

Jin, S., Cao, Z., Yu, C., 2025. Two-stage vision system: Application of multi-perspective object detection network and character recognition network in industrial product classification. Eng. Appl. Artif. Intell. 156, 111190. http://dx.doi.org/10.1016/j.engappai.2025.111190.

Kopuklu, O., Zheng, J., Xu, H., Rigoll, G., 2021. Driver anomaly detection: A dataset and contrastive learning approach. In: Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.. WACV, pp. 91–100. http://dx.doi.org/10.1109/wacv48630.2021.00014.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25, http://dx.doi.org/10.1145/3065386.

Kumar, K.S., Bai, M.R., 2023. LSTM based texture classification and defect detection in a fabric. Meas. Sens. 26, 100603. http://dx.doi.org/10.1016/j.measen.2022.100603.

Lahmyed, R., El Ansari, M., Ellahyani, A., 2019. A new thermal infrared and visible spectrum images-based pedestrian detection system. Multimedia Tools Appl. 78, 15861–15885. http://dx.doi.org/10.1007/s11042-018-6974-5.

Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D.D., Chen, M., 2014. Medical image classification with convolutional neural network. In: Proc. 13th Int. Conf. Control Autom. Rob. Vis.. ICARCV, IEEE, pp. 844–848. http://dx.doi.org/10.1109/icarcv.2014.7064414.

Li, R., Jin, M., Paquit, V.C., 2021. Geometrical defect detection for additive manufacturing with machine learning models. Mater. Des. 206, 109726. http://dx.doi.org/10.1016/j.matdes.2021.109726.

Li, C.-L., Sohn, K., Yoon, J., Pfister, T., 2021. Cutpaste: self-supervised learning for anomaly detection and localization. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. CVPR, pp. 9659–9669. http://dx.doi.org/10.1109/cvpr46437.2021.00954.

Liu, Y., Ge, Z., 2018. Weighted random forests for fault classification in industrial processes with hierarchical clustering model selection. J. Process Control 64, 62–70. http://dx.doi.org/10.1016/j.jprocont.2018.02.005.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S., 2022. A ConvNet for the 2020s. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. CVPR, pp. 11976–11986. http://dx.doi.org/10.1109/cvpr52688.2022.01167.

Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Jiao, J., Liu, Y., 2024. Vmamba: Visual state space model. Adv. Neural Inf. Process. Syst. 37, 103031–103063. http://dx.doi.org/10.52202/079017-3273.

Liu, J., Xie, G., Wang, J., Li, S., Wang, C., Zheng, F., Jin, Y., 2024. Deep industrial image anomaly detection: A survey. Mach. Intell. Res. 21 (1), 104–135. http://dx.doi.org/10.1007/s11633-023-1459-z.

Mehta, D., Klarmann, N., 2023. Autoencoder-based visual anomaly localization for manufacturing quality control. Mach. Learn. Knowl. Extr. 6 (1), 1–17. http://dx.doi.org/10.3390/make6010001.

Michno, T., Holom, R., Schmalzer, S., Meyer-Heye, P., Scampone, G., Riegler, E., Hartmann, M., Repanšek, U., Košir, N., Šifrer, P., Poczęta, K., 2025. Thermal imaging-based defects prediction in high-pressure die casting using hybrid neural networks and fuzzy cognitive maps. In: Proc. 11th Int. Conf. Nat. Lang. Comput. (NATL 2025). 15, (22), pp. 153–180. http://dx.doi.org/10.5121/csit.2025.152212.

Mielgo, P., Bregon, A., Alonso-González, C.J., López, D., Martínez-Prieto, M.A., Pulido, B., 2024. A deep learning solution for quality control in a die casting process. In: Proc. Ann. Conf. PHM Soc., vol. 16, (1), http://dx.doi.org/10.36001/phmconf.2024.v16i1.3973.

Mielgo, P., Bregon, A., Alonso-Gonzalez, C.J., Martínez-Prieto, M.A., Pulido, B., 2025. A contrastive learning approach for anomaly detection in multi-view scenarios. In: Proc. Ann. Conf. PHM Soc., vol. 17, (1), http://dx.doi.org/10.36001/phmconf.2025.v17i1.4562.

Milo, M.W., Roan, M., Harris, B., 2015. A new statistical approach to automated quality control in manufacturing processes. J. Manuf. Syst. 36, 159–167. http://dx.doi.org/10.1016/j.jmsy.2015.06.001.

Mishra, P., Verk, R., Fornasier, D., Piciarelli, C., Foresti, G.L., 2021. VT-ADL: A vision transformer network for image anomaly detection and localization. In: Proc. IEEE 30th Int. Symp. Ind. Electron.. ISIE, IEEE, pp. 01–06. http://dx.doi.org/10.1109/isie45552.2021.9576231.

Mohammed, A., Kora, R., 2023. A comprehensive review on ensemble deep learning: Opportunities and challenges. J. King Saud Univ. Comput. Inf. Sci. 35 (2), 757–774. http://dx.doi.org/10.1016/j.jksuci.2023.01.014.

Müller, P., 2013. Coordinate metrology by traceable computed tomography (Ph.D. thesis). Technical University of Denmark.

Ouyang, Z., Sun, X., Chen, J., Yue, D., Zhang, T., 2018. Multi-view stacking ensemble for power consumption anomaly detection in the context of industrial internet of things. IEEE Access 6, 9623–9631. http://dx.doi.org/10.1109/access.2018.2805908.

Ozdemir, R., Koc, M., 2019. A quality control application on a smart factory prototype using deep learning methods. CSIT, In: Proc. 2019 IEEE 14th Int. Conf. Comput. Sci. Inf. Technol., vol. 1, IEEE, pp. 46–49. http://dx.doi.org/10.1109/stc-csit.2019.8929734.

Paladini, E., 2000. An expert system approach to quality control. Expert Syst. Appl. 18 (2), 133–151. http://dx.doi.org/10.1016/s0957-4174(99)00059-7.

Park, J., Jun, M.B., Yun, H., 2022. Development of robotic bin picking platform with cluttered objects using human guidance and convolutional neural network (CNN). J. Manuf. Syst. 63, 539–549. http://dx.doi.org/10.1016/j.jmsy.2022.05.011.

Peres, R.S., Barata, J., Leitao, P., Garcia, G., 2019. Multistage quality control using machine learning in the automotive industry. IEEE Access 7, 79908–79916. http://dx.doi.org/10.1109/access.2019.2923405.

Perner, P., 1994. A knowledge-based image-inspection system for automatic defect recognition, classification, and process diagnosis. Mach. Vis. Appl. 7, 135–147. http://dx.doi.org/10.1007/bf01211659.

Qu, Z., Tao, X., Shen, F., Zhang, Z., Li, T., 2023. Investigating shift equivalence of convolutional neural networks in industrial defect segmentation. IEEE Trans. Instrum. Meas. 72, 1–17. http://dx.doi.org/10.1109/tim.2023.3325514.

Radziwill, N.M., Benton, M.C., 2017. Cybersecurity cost of quality: Managing the costs of cybersecurity risk management. http://dx.doi.org/10.48550/arXiv.1707.02653, ArXiv Prepr.

Ravi, R., Yadhukrishna, S., et al., 2020. A face expression recognition using CNN & LBP. In: Proc. 4th Int. Conf. Comput. Methodol. Commun.. ICCMC, IEEE, pp. 684–689. http://dx.doi.org/10.1109/iccmc48092.2020.iccmc-000127.

Ring, E., Ammer, K., 2012. Infrared thermal imaging in medicine. Physiol. Meas. 33 (3), R33. http://dx.doi.org/10.1088/0967-3334/33/3/r33.

Sachin, R., Sowmya, V., Govind, D., Soman, K., 2018. Dependency of various color and intensity planes on CNN based image classification. In: Proc. 3rd Int. Symp. Signal Process. Intell. Recognit. Syst.. SIRS, Springer, pp. 167–177. http://dx.doi.org/10.1007/978-3-319-67934-1_15.

Saleh, R.A., Konyar, M.Z., Kaplan, K., Ertunç, H.M., 2022. Tire defect detection model using machine learning. In: Proc. 2nd Int. Conf. Emerg. Smart Technol. Appl. (ESmarTA). IEEE, pp. 1–5. http://dx.doi.org/10.1109/esmarta56775.2022.9935140.

Serranti, S., Gargiulo, A., Bonifazi, G., 2011. Characterization of post-consumer polyolefin wastes by hyperspectral imaging for quality control in recycling processes. Waste Manage. 31 (11), 2217–2227. http://dx.doi.org/10.1016/j.wasman.2011.06.007.

Smith, A.D., Du, S., Kurien, A., 2023. Vision transformers for anomaly detection and localisation in leather surface defect classification based on low-resolution images and a small dataset. Appl. Sci. 13 (15), 8716. http://dx.doi.org/10.3390/app13158716.

Tabassi, E., 2023. Artificial intelligence risk management framework (AI RMF 1.0). http://dx.doi.org/10.6028/nist.ai.100-1.

Tan, M., Le, Q., 2019. Efficientnet: rethinking model scaling for convolutional neural networks. In: Proc. Int. Conf. Mach. Learn.. ICML, PMLR, pp. 6105–6114.

Tsai, D.-M., Jen, P.-H., 2021. Autoencoder-based anomaly detection for surface defect inspection. Adv. Eng. Inf. 48, 101272. http://dx.doi.org/10.1016/j.aei.2021.101272.

Usuga Cadavid, J.P., Lamouri, S., Grabot, B., Pellerin, R., Fortin, A., 2020. Machine learning applied in production planning and control: a state-of-the-art in the era of Industry 4.0. J. Intell. Manuf. 31, 1531–1558. http://dx.doi.org/10.1007/s10845-019-01531-7.

Varith, J., Hyde, G., Baritelle, A., Fellman, J., Sattabongkot, T., 2003. Non-contact bruise detection in apples by thermal imaging. Innov. Food Sci. Emerg. Technol. 4 (2), 211–218. http://dx.doi.org/10.1016/s1466-8564(03)00021-3.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30, http://dx.doi.org/10.5555/3295222.3295349.

Vila, J., Calpe, J., Pla, F., Gómez, L., Connell, J., Marchant, J., Calleja, J., Mulqueen, M., Muñoz, J., Klaren, A., et al., 2005. SmartSpectra: applying multispectral imaging to industrial environments. Real-Time Imag. 11 (2), 85–98. http://dx.doi.org/10.1016/j.rti.2005.04.007.

Vollmer, M., 2020. Infrared thermal imaging. In: Computer Vision: A Reference Guide. Springer, pp. 666–670. http://dx.doi.org/10.1007/978-3-030-63416-2.

Wandinger, U., 2005. Introduction to Lidar. In: Lidar: Range-Resolved Optical Remote Sensing of the Atmosphere. Springer, pp. 1–18. http://dx.doi.org/10.1007/0-387-25101-4_1.

Wang, T., Chen, Y., Qiao, M., Snoussi, H., 2018. A fast and robust convolutional neural network-based defect detection model in product quality control. Int. J. Adv. Manuf. Technol. 94, 3465–3471. http://dx.doi.org/10.1007/s00170-017-0882-0.

Wang, Y., Sun, S., Chen, X., Zeng, X., Kong, Y., Chen, J., Guo, Y., Wang, T., 2021. Short-term load forecasting of industrial customers based on SVMD and XGBoost. Int. J. Electr. Power Energy Syst. 129, 106830. http://dx.doi.org/10.1016/j.ijepes.2021.106830.

Wang, J., Sun, W., Shou, W., Wang, X., Wu, C., Chong, H.-Y., Liu, Y., Sun, C., 2015. Integrating BIM and LiDAR for real-time construction quality control. J. Intell. Robot. Syst. 79, 417–432. http://dx.doi.org/10.1007/s10846-014-0116-8.

Wang, K., Zhang, J., Ni, H., Ren, F., 2021. Thermal defect detection for substation equipment based on infrared image using convolutional neural network. Electronics 10 (16), 1986. http://dx.doi.org/10.3390/electronics10161986.

Weimer, D., Scholz-Reiter, B., Shpitalni, M., 2016. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. CIRP Ann. 65 (1), 417–420. http://dx.doi.org/10.1016/j.cirp.2016.04.072.

Wolpert, D.H., 1992. Stacked generalization. Neural Netw. 5 (2), 241–259. http://dx.doi.org/10.1016/s0893-6080(05)80023-1.

Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K., 2017. Aggregated residual transformations for deep neural networks. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. CVPR, pp. 1492–1500. http://dx.doi.org/10.1109/cvpr.2017.634.

Xie, Y., Richmond, D., 2018. Pre-training on grayscale imagenet improves medical image classification. In: Proc. Eur. Conf. Comput. Vis. Workshops. pp. 476–484. http://dx.doi.org/10.1007/978-3-030-11024-6_37.

Xing, J., De Baerdemaeker, J., 2005. Bruise detection on 'jonagold' apples using hyperspectral imaging. Postharvest Biol. Technol. 37 (2), 152–162. http://dx.doi.org/10.1016/j.postharvbio.2005.02.015.

Yang, D., Cui, Y., Yu, Z., Yuan, H., 2021. Deep learning based steel pipe weld defect detection. Appl. Artif. Intell. 35 (15), 1237–1249. http://dx.doi.org/10.1080/08839514.2021.1975391.

Yashchin, E., 1994. Monitoring variance components. Technometrics 36 (4), 379–393. http://dx.doi.org/10.1080/00401706.1994.10485844.

Younus, A.M., Yang, B.-S., 2012. Intelligent fault diagnosis of rotating machinery using infrared thermal image. Expert Syst. Appl. 39 (2), 2082–2091. http://dx.doi.org/10.1016/j.eswa.2011.08.004.

Zhang, Z., Farnsworth, M., Song, B., Tiwari, D., Tiwari, A., 2022. Deep transfer learning with self-attention for industry sensor fusion tasks. IEEE Sens. J. 22 (15), 15235–15247. http://dx.doi.org/10.1109/jsen.2022.3186505.

Zhao, C., Hu, C., Shao, H., Dunkin, F., Wang, Y., 2025. Trusted video-based sewer inspection via support clip-based Pareto-optimal evidential network. IEEE Signal Process. Lett. 32, 356–360. http://dx.doi.org/10.1109/lsp.2024.3480830.

Zhong, R.Y., Xu, X., Klotz, E., Newman, S.T., 2017. Intelligent manufacturing in the context of Industry 4.0: a review. Engineering 3 (5), 616–630. http://dx.doi.org/10.1016/j.eng.2017.05.015.

Zhou, Z., Wang, L., Fang, N., Wang, Z., Qiu, L., Zhang, S., 2024. R3d-ad: Reconstruction via diffusion for 3d anomaly detection. In: Proc. Eur. Conf. Comput. Vis.. Springer, pp. 91–107. http://dx.doi.org/10.1007/978-3-031-72764-1_6.

Zhu-Mao, L., Qing, L., Tao, J., Yong-Xin, L., Yu, H., Yang, B., 2018. Research on thermal fault detection technology of power equipment based on infrared image analysis. In: Proc. 3rd IEEE Adv. Inf. Technol. Electron. Autom. Control Conf.. IAEAC, IEEE, pp. 2567–2571. http://dx.doi.org/10.1109/iaeac.2018.8577908.