Article | Received 1 Februrary 2026; Revised 9 April 2026; Accepted 21 May 2026; Published 26 June 2026
https://doi.org/10.55092/aias20260006

An ensemble deep learning approach for surface defect detection in aluminum die-cast gas meter lids

Wuyang Qian[1], Olayinka Ayorinde[2], Suhao Chen[1], Lin Guo[1,*] and Dean Jensen[1]

[1] Industrial Engineering, South Dakota School of Mines and Technology, Rapid City, USA
[2] Nevada Gold Mines LLC, Elko, USA

* Correspondence author; E-mail: lin.guo@sdsmt.edu.

Highlights:

- Proposed an ensemble deep learning framework for detecting surface defects in die-cast parts.
- Evaluated CNN, ResNet-18, and ViT models through tuning, validation, and comparison.
- Demonstrated superior accuracy and robustness of the ensemble using real-world data.

Abstract: Aluminum die-cast products often exhibit surface defects that vary in type and severity depending on product function and design requirements. Current defect detection primarily relies on manual inspection, which demands significant expertise, raises health and fatigue concerns, and is prone to human error. Automated defect detection offers a promising solution to reduce costs, improve efficiency, resolve occupational safety and health concerns, and mitigate challenges in labor shortages and training. This paper presents an ensemble deep learning (DL) approach for detecting surface defects in aluminum die-cast lids for residential gas meters, a quality-critical component with stringent safety standards. Specifically, we implement and evaluate three state-of-the-art DL architectures: a convolutional neural network (CNN), residual networks (ResNet-18), and Vision Transformer (ViT). In addition, we develop an ensemble model to further enhance performance. We leverage grid search and cross-validation for hyperparameter tuning and train/test each model ten times for comprehensive performance evaluation. Experiments on a large real-world dataset demonstrate that all models achieve high accuracy, precision, and recall, with CNN and ResNet-18 slightly outperforming ViT. The ensemble model further improves prediction accuracy and robustness. The paired t-tests showed that the ensemble model significantly performed better compared to CNN and ViT model. In summary, this study contributes to the advancement of automated inspection of surface defects in die-cast products by systematically comparing state-of-the-art deep learning methods, discussing model selection criteria, and optimizing ensemble strategies. Centered on CNN, ResNet-18, and ViT architectures, it proposes a rigorous methodological framework for surface defect detection and provides a foundational basis for subsequent research in in-situ quality control. Our codes are available at https://github.com/Alexruoyun/Aluminum-Die-Casting-Surface-Defect-Detection.

Keywords: surface defect detection; aluminum die casting; deep learning; convolution neural networks; residual networks; vision transformer; ensemble

1. Introduction

Die casting is a metallurgical process that forces molten metals such as aluminum, zinc, and magnesium into a steel mold under high pressure, producing parts with complex geometries and excellent surface finish [1]. Aluminum is the most commonly used metal in die casting due to its lightweight nature, corrosion resistance, good strength, and ability to tolerate high operating temperatures [2]. The aluminum die-casting industry is crucial in manufacturing lightweight, strong, and durable components for various sectors such as automotive, aerospace, consumer electronics, and utility equipment, among many others. In recent years, the market has experienced significant growth, driven by increasing demand for lightweight materials in automotive and aerospace applications and the growing emphasis on energy efficiency and sustainability [3]. According to the American Foundry Society (AFS), global aluminum casting production increased from 10.88 million metric tons in 2010 to 16.03 million metric tons in 2020; production in the United States alone increased from 1.13 million metric tons to 1.43 million metric tons during the same period [4,5]. The Aluminum Association estimated that U.S. aluminum die casting shipments in 2020 reached 1.12 million metric tons [6], representing 78.67% of the total aluminum casting production volume by AFS. The North American Die Casting Association (NADCA) estimated aluminum die-casting sales to be $8 billion in 2019 [7].

However, die-cast aluminum parts suffer from many defects. Common defects include surface defects (e.g., cold flow, drag, soldering, blister, deformation, swirls, chill, heat checking, oil stain, sinks, flashes, and inclusions) and internal defects (e.g., inner crack, deformation, and interlayers). Figure 1 illustrates some common surface defects in a die-cast product, a residential gas meter lid. Even minor defects of these four types can compromise the functionality of die-cast parts, make them a hazard to customers, and ultimately jeopardize a foundry's business. For example, defective aluminum brackets in automotive engines may cause engine failures and even lead to accidents. Defects in the lids of natural gas meters may result in gas leaks, threatening human life. Therefore, it is imperative to implement stringent quality inspection in the manufacturing process of die-cast aluminum products. In addition, better quality inspection also benefits foundries economically. A survey conducted by NADCA in 2014 shows that the median scrap rate was as high as 8% of the parts made and the equipment utilization rate was only 68% in high-pressure die-casting [8]. If defective parts can be inspected accurately and swiftly, a foundry may increase the equipment utilization rate and significantly improve efficiency.

Unfortunately, quality inspection is challenging in die casting, even for surface defect detection. Traditional quality inspection in the foundry industry are performed manually by specialized quality inspectors who, over time, have developed visual skills that allow them to detect the majority of surface defects. This practice heavily relies on human expertise. It is also time-consuming, labor-intensive, and prone to human errors [9]. Fatigue and variability in human judgment may result in inspection inconsistencies. The situation is compounded by a significant shortage of skilled workers in the casting industry, one of the top three concerns or priorities based on a survey conducted by AFS in Quarter 3, 2024 [10].

To address these challenges and capitalize on the growing market opportunities, researchers turn to advanced technologies such as machine vision systems and artificial intelligence for surface defect detection and process optimization [11]. These technologies offer opportunities to improve product quality, reduce scrap rates, and increase productivity in the aluminum die-casting industry [12]. Recent advances in deep learning, particularly convolutional neural networks (CNNs), have substantially improved automated surface defect detection in computer vision. However, defect detection using images captured with accessible devices (e.g., smartphones and low-cost cameras) remains understudied. Furthermore, the potential of state-of-the-art architectures—including Vision Transformers (ViT) and ensemble approaches—for this specific application has not been well explored.

In this paper, we propose an ensemble deep learning approach for automatically detecting surface defects in die-cast aluminum parts by analyzing webcam-captured images. We built three state-of-the-art models, including CNN, ResNet-18, and ViT, to analyze the images of the die-cast parts and identify surface defects. An ensemble model is constructed and achieves a more robust performance.

The remainder of this article is organized as follows. Section 2 presents a review of the literature of related studies on surface defect detection for die-cast aluminum alloys using machine learning and deep learning techniques. Section 3 explains the residential gas meter lids dataset that we collect and the pre-processing of the raw images. Section 4 discusses the deep learning architectures that we build, followed by experimental results in Section 5. Section 6 concludes the paper by summarizing its contributions, limitations, and directions for future work.

2. Literature review

Deep learning-based defect detection facilitates real-time monitoring, improves quality, lowers inspection costs, and enhances productivity. It plays a crucial role in supporting cyber-physical systems that enable automation and seamless data exchange, optimizing resource use to drive innovation and uphold manufacturing safety. As a foundational step toward automated manufacturing, deep learning-based automated defect detection impacts not only product and process quality but also significantly reduces processing times, protects laborers' safety and health, and accelerates quality assurance in production. This aligns with the goals of Industry 4.0 and 5.0, which emphasize automation, data integration, human-machine collaboration, and human-centered automation, harnessing the efficiency and data-driven capabilities of deep learning technology.

Table 1 summarizes relevant studies organized chronologically. Our initial focus is on studies employing deep learning techniques for the classification of surface defects in aluminum die-cast alloys through computer vision. However, similar tasks for other types of products (*i.e.,* zinc or magnesium die-cast products) have also been explored in the literature. Considering there are only limited studies in this area, to provide a comprehensive survey, we include studies in recent years involving similar products or tasks that utilize deep learning methods, including different aluminum alloys [13–17], iron or steel products [18–21], and other types of alloys such as Sn-Ag-Cu alloy [22] and titanium alloy [23]. While different products may have distinct features, the defect detection tasks share similarities. These tasks typically include image collection, pre-processing, feature extraction (if necessary), model training,

performing classification or regression on training and testing datasets, and evaluating model performance.

There are various tasks in quality control for die-cast or similar products, including defect detection [24], quality prediction [25], feature extraction [26], training image generation [17], production parameter optimization [27], energy monitoring [28], and human-robot collaboration-based inspection [29]. The most commonly addressed task is classification (for surface defects, internal defects, microstructure types, or alloy types), including binary classification [14–16,19] and multiclass classification [18,30]. Additional tasks include segmentation [31,32], predicting quality features [13,33], and generating training images [17].

Table 1. Related studies in literature.

Paper	Products	Tasks	Data	Methods	Results
[22]	solder alloys	CLS[a]: microstructure	microscopy images	SVM[b], KNN[c], RF[d]	Acc.[e]: 97.84% ± 2.65%
[18]	low carbon steel	CLS: microstructure	microscopy images	CNN	Acc.: 93.94%
[19]	iron casting of submersible pump impellers (SPIs)	CLS: surface defects	grayscale images	CNN (VGG *etc.*) + classifiers (SVM, MLP[f], *etc.*)	Acc.: 62%–99.6%
[13]	aluminum die-cast alloys	REG[g]: dendrite arm spacing	microscopy images	CNN	R^2 : 91.5%
[23]	titanium alloy castings	CLS: internal defects	X-ray images	CNN (BX-Net)	Rec.[h]: 99%; Acc.: 99%
[14]	automotive aluminum castings	CLS: surface and internal defects	X-ray images	CNN	Acc.: 94.2%
[20]	steel strips; SPIs; PCB; pistons	CLS: surface defects	images and CCD scans	CNN+LSTM	Pre.[i]/Rec.: 99.1%–100%
[21]	cast SPIs	CLS: surface defects	grayscale images	ResNet50 & CNN ensemble	Pre.: 99.89%; Acc.: 98.18%
[15]	aluminum alloy castings	CLS: alloys based on microstructure	microscopy images	CNN	Acc.: 97.9%
[16]	aluminum alloy castings	CLS: porosity detection	microscopy images	CNN	Acc.: 94%
[17]	magnesium and aluminum alloys	DA[j]	X-ray images	I-DCGAN[k]	SIS[l]: 78.7%, 73.8%
[31]	aluminum alloy casting	CLS, SEG[m]: surface defects	camera images	CNN (U-Net)	Dic.[n]: 81%
[34]	automotive aluminum alloy casting	CLS: surface defect	2D camera images and 3D laser scans	denoising autoencoder	TPR[o]: 96%
[32]	aluminum alloy casting	SEG: surface defects	X-ray images	dual–channel encoder–decoder	Dic.: 75.71%, mIOU[p]: 92%
[30]	aluminum die-cast gas meter lids	CLS: surface defects	camera images	CNN, ResNet, ViT	Acc.: 97.92%–99.11%

[a] CLS: classification; [b] SVM: support vector machine; [c] KNN: K-nearest neighbors; [d] RF: random forest; [e] Acc.: accuracy; [f] MLP: multilayer perceptron; [g] REG: regression; [h] Rec.: recall; [i] Pre.: precision; [j] DA: data augmentation; [k] I-DCGAN: interpolation-deep convolutional generative adversarial network; [l] SIS: similarity to the ideal solution; [m] SEG: segmentation; [n] Dic.: Dice coefficient; [o] TPR: true positive rate; [p] mIOU: mean intersection over union.

Image data types include microscopy (or microstructure) images [13,15,16,18,22], X-ray (or radiographic) images [14,17,23,32], grayscale images [19–21], and camera-based images [30,34]. Camera and grayscale images provide advantages in terms of portability, preservation of intrinsic surface features, and operational efficiency. However, they are not suitable for components exhibiting internal defects. X-ray images are highly effective for detecting internal defects, but can be challenging to implement on production lines and are unnecessary for products with mainly surface defects. Microscopy images, while particularly effective for identifying specific microstructural defects, require extensive preprocessing (e.g., polishing) and are difficult to use for production quality control. A notable trend is the use of publicly available image datasets to train baseline models [20,21]. While these datasets offer the advantage of producing comparable and easily verifiable results, they have limitations regarding the variety of product features, quality requirements, and defect types. Specifically, the findings of these studies may have limited practical applicability to improve the quality of die-cast products different from those in the public datasets. Researchers have employed diffusion models to improve data quality, particularly for image data, by learning to iteratively remove noise from the input data [35].

The predominant method for classification tasks is CNN. Traditional machine learning methods, such as support vector machine, K-nearest neighbor, naive Bayes, and random forest, were often used in earlier studies [22]. However, these methods have largely been replaced by deep learning (DL) models such as CNNs. A growing trend is the use of ensemble or integrated deep learning models to enhance the accuracy and robustness of results[36–38]. Notably, few studies have yet explored the application of ViT, a more recent model, for defect detection in die-casting parts [30]. Recent advances in edge–cloud collaborative computing have highlighted the importance of distributed intelligence and model optimization [39].

Different quality tasks and methods require various evaluation metrics. For classification tasks, accuracy is the most commonly used metric. However, variations in products, tasks, data types, and inspection accuracy requirements across industries can lead to significant differences in model accuracy. Accuracy is also not a good metric for highly imbalanced datasets. Precision and recall are also commonly used [20,21,23]. All three metrics—accuracy, precision, and recall—are widely employed to evaluate classification models, with higher values indicating better performance. For feature prediction using regression, metrics such as R^2 are commonly adopted [13].

Internal defect analysis has also been widely studied in the literature. Yang *et al.* investigated the effects of casting pressure on porosity in large aluminum die-casting components [40], while Bosse *et al.* explored automated porosity characterization using X-ray imaging combined with machine learning techniques [41]. Comprehensive reviews of deep learning-based defect detection in aluminum die-casting can be found in [42]. However, in practical aluminum die-casting production, defective parts are typically remelted and recycled, making defect localization or segmentation unnecessary for downstream decision-making. Consequently, although object detection and segmentation methods are effective for defect localization, they provide limited additional practical value in this context, and this study instead focuses on image-level classification as a more cost-effective solution. Furthermore, while deep learning methods for X-ray inspection exist, such systems are costly and not available in the collaborating manufacturer's environment; therefore, this work focuses on surface inspection using conventional optical images.

In summary, the existing literature exhibits several limitations. First, studies focusing on surface defect detection in aluminum die-cast components using real-world datasets remain limited. Second, the application of advanced architectures, such as ViT, has not been thoroughly investigated for this task. Third, ensemble deep learning approaches incorporating comprehensive model evaluation for robust prediction have yet to be systematically explored.

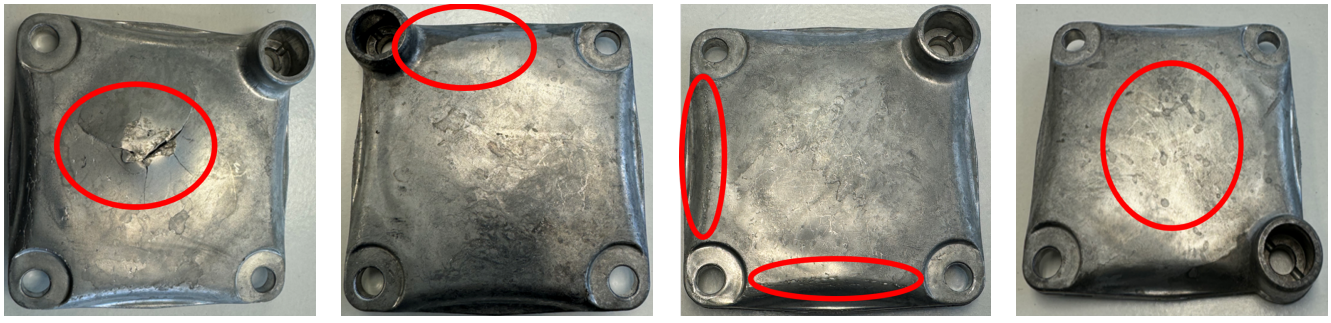
In this paper, we address these gaps by analyzing a production-grade industrial product—residential natural gas meter lid—with surface defects. We investigate state-of-the-art models (ViT, ResNet-18, and CNN) for defect detection and propose an ensemble model to improve performance. This study also aims to provide insights into reducing costs and boosting the efficiency of machine vision systems. The dataset was collected using webcams, offering an inexpensive and easily calibrated approach for data acquisition in real production settings. The resulting quality evaluations can be promptly generalized and adapted to drive quality improvement efforts for different products.

3. Dataset and pre-processing

3.1. Dataset

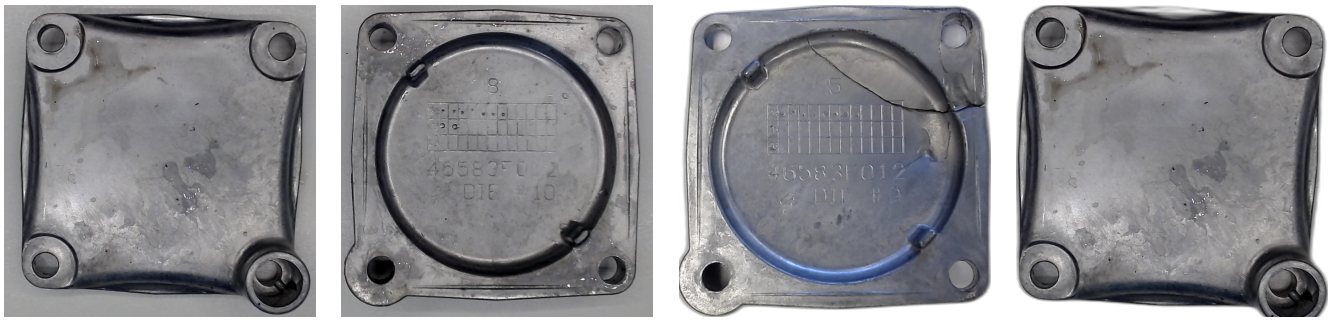
In this pilot study, we focus on aluminum die-cast lids for residential natural gas meters. These meters measure household gas flow and are critical to urban natural gas transmission systems. Given natural gas's flammability and high-pressure combustibility, leak prevention is paramount—a role the lid fulfills. Residential gas meters require frequent maintenance, and their lids must be leak-proof. Aluminum alloy is ideal for this purpose due to its strength and durability in harsh environments. However, as noted in Section 1, aluminum die-cast lids often exhibit surface defects (Figures 1 and 2), necessitating rigorous defect detection during manufacturing.

Table 2 summarizes our dataset. The dataset consists of 3,347 aluminum die-cast residential gas meter lids manufactured by a foundry in the Midwest U.S., each imaged using two Logitech C920 Pro HD webcams with Light Emitting Diode (LED) strip lighting for optimal edge and surface illumination. Expert inspectors categorized the samples as: flawless (n = 2,022; labeled 0), crack (n = 894; labeled 1), and cold flow (n = 431; labeled 2). The uncompressed images (1920 × 1080 pixels, red, green, and blue (RGB) format) represent the raw dataset without augmentation, with each image's defect category explicitly labeled.



(a) (b) (c) (d)

Figure 1. Example surface defects in aluminum die cast parts: **(a)** Crack; **(b)** Cold flow; **(c)** Heat checking; **(d)** Oil stain.



(a) (b) (c) (d)

Figure 2. Gas meter lid dataset examples: **(a)** Original front; **(b)** Original back; **(c)** Processed defective (e.g., crack); **(d)** Processed flawless.

Table 2. Dataset summary.

Dataset Information	Value
Number of Images	3347
Image Dimensions	1920×1080
Image Type	RGB Color Three-channel
Number of Images by Labels	
Flawless	2022
Crack	894
Cold Flow	431

3.2. Pre-processing

The raw images (see Figure 2a,b for example) contain noise (e.g., variations in background color and brightness). They are also too big (in terms of dimensions) and too expensive (in terms of computation) to model, so we pre-process the raw images to remove irrelevant information. Figure 3 shows the flowchart of dataset pre-processing.

First, we remove the background of the raw images to reduce the disturbance of the background using the function remove inside the Python library rembg [43]. Second, we use the function ImageChops in the Python library pillow to crop the image to protrude the parts [44]. Since we have removed the background in the first step and replaced the background color with black, pruning the extra background significantly reduces the size of the images and removes useless background noise. Third, we reduce the resolutions of the raw images to the lowest quality with the function Image.save in the library pillow. This step compresses the images and saves them with a much smaller size for computational efficiency. Lastly, we resize all the images to 224×224 pixels using resize function in opencv-python library [45]. By resizing all the images, we can concatenate them as a four-dimensional matrix. The first dimension is the number of images, the second dimension refers to the three color channels, and the last two dimensions are the height and width of the processed images. After resizing, the images can be concatenated into a unified four-dimensional matrix that is readily used as input for machine learning models. The resolution reduction and resizing steps substantially decrease computational and memory demands. However, these operations inevitably lead to some loss of fine-grained visual information from the original images due

to the constraints of limited hardware resources. Examples of the pre-processed flawless and defective images are shown in Figure 2c,d.

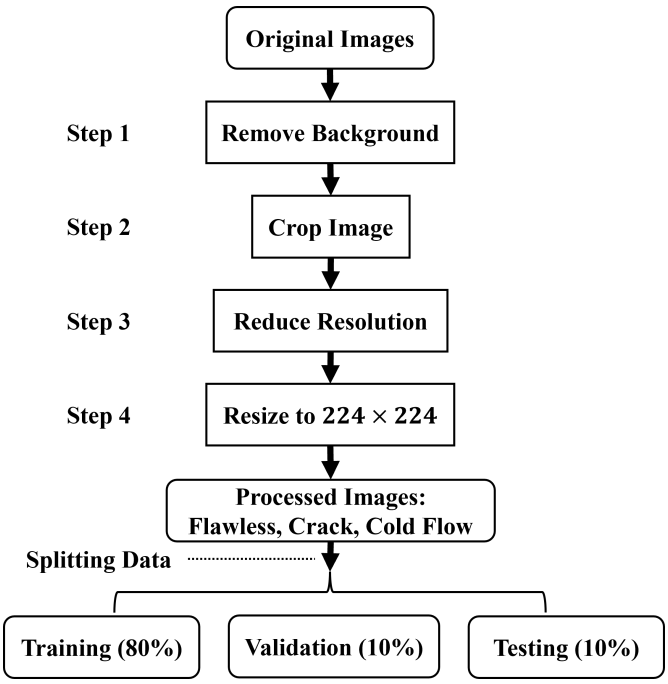


Figure 3. Image pre-processing flowchart.

Before we train the models, we split the dataset into training, validation, and testing subsets with a ratio of 8 : 1 : 1. Since the datasets are imbalanced (*i.e.,* more flawless images than defective images), we further balance the training set by oversampling images of the minority classes, the crack and cold flow. As a result, there are 4854 images in the training set, 606 images in the validation set, and 334 images in the testing set (89 crack, 43 cold flow, and 202 flawless). Oversampling was employed to handle class imbalance while preserving the original data distribution, as synthetic approaches (e.g., Synthetic Minority Over-sampling Technique (SMOTE) or Generative Adversarial Network (GAN)) may introduce distributional artifacts that could compromise model reliability in industrial inspection tasks.

4. Methods

We develop three state-of-the-art deep learning architectures for surface defect detection: a CNN, ResNet-18, and a ViT. Training ResNet-18 and ViT from scratch typically requires a large-scale labeled dataset, which is unavailable in this study. Therefore, the CNN is trained from scratch, while transfer learning is employed to fine-tune pre-trained ResNet-18 and ViT models using the proposed pre-processed dataset. In addition, an ensemble model is constructed by combining the posterior predictions of the three individual models to further improve classification accuracy and robustness.

Let the input image be denoted as $X \in R^{3\times H\times W}$, where 3 represents the RGB channels and $H = W = 224$ after preprocessing. The corresponding ground-truth label is $y \in \{1,2,3\}$, representing the classes flawless, crack, and cold flow. All models learn a mapping $f_\theta : X \to \hat{y}$, where θ denotes trainable parameters and $\hat{y}$ is the predicted class label obtained from class probabilities $\hat{p} \in R^3$.

4.1. Convolutional neural networks

The first architecture developed in this study is a conventional CNN originally proposed in [46]. A CNN consists of convolutional layers for feature extraction, pooling layers for spatial downsampling, and fully connected layers for classification. Convolutional layers learn local feature representations by applying trainable kernels to the input feature maps, while nonlinear activation functions, such as Rectified Linear Unit (ReLU) or its variants, introduce nonlinearity and enable the modeling of complex patterns. Pooling layers reduce spatial dimensionality and computational cost, improve feature robustness, and help mitigate overfitting. The extracted high-level features are subsequently fed into fully connected layers, where softmax or sigmoid functions are applied depending on the classification task.

In this study, the proposed CNN architecture comprises three convolutional blocks. Each block consists of a 3×3 convolutional layer with a stride of 1, followed by a LeakyReLU activation function to improve gradient flow and a max-pooling layer for spatial downsampling. Batch normalization is applied after each block to stabilize training and accelerate convergence. For each convolutional block, given input X, the output is computed as $\hat{P} = BN(MaxPool(LeakyReLU(X * K + b)))$, where $*$ denotes the convolution operation, K represents the learnable convolution kernels, and b is the bias term. The output of the final convolutional block is flattened and passed through a fully connected layer, followed by ReLU activation, batch normalization, and dropout regularization to prevent overfitting: $\tilde{h} = Dropout(BN(ReLU(W_1 * Flatten(\hat{P}) + b_1)))$, where W_1 and b_1 are the weight matrix and bias of the fully connected layer, respectively. The final fully connected layer produces three output nodes, and the class probabilities are obtained using the softmax function: $\hat{p} = Softmax(W_2\tilde{h} + b_2)$ corresponding to the flawless, crack, and cold flow categories. The layer dimensions are determined by the input image size and kernel configuration, as illustrated in Figure 4.

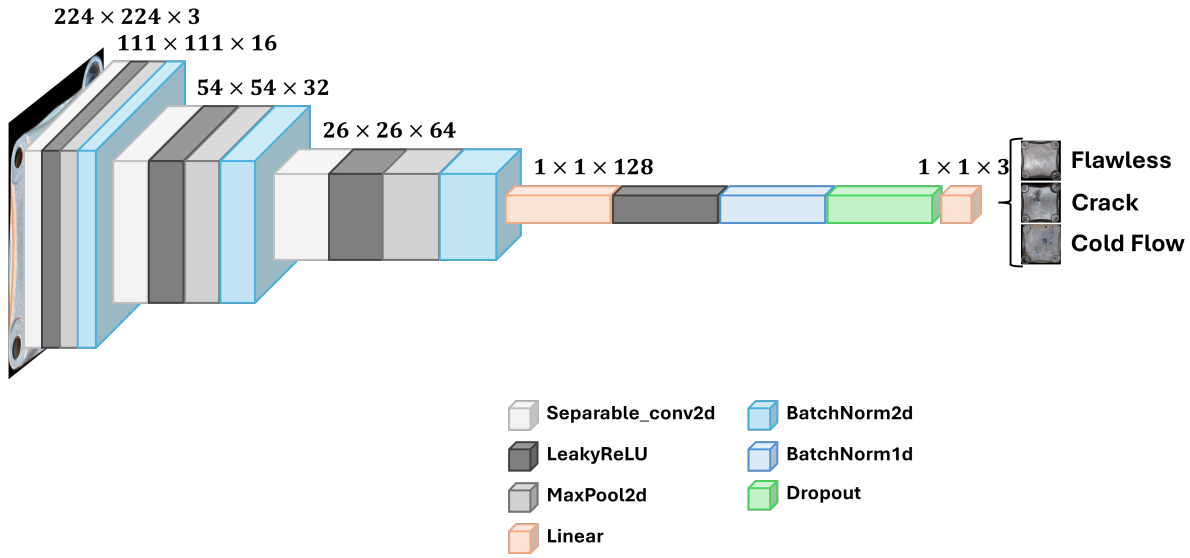


Figure 4. CNN architecture.

4.2. Pre-trained ResNet-18

The second deep learning architecture employed in this study is a pre-trained ResNet-18 model, originally trained on the ImageNet-1K dataset. ResNet-18 is a residual convolutional neural network consisting of

18 layers and is specifically designed to mitigate the vanishing gradient problem and improve training efficiency through the use of residual connections. In conventional CNNs, gradients may gradually diminish as they propagate backward through deep networks, resulting in slow convergence and limited ability to learn complex feature representations. To address this limitation, ResNet introduces shortcut (residual) connections between convolutional layers, as illustrated by the green curved arrows in Figure 5. These connections enable the network to learn residual mappings rather than directly approximating the desired underlying functions, thereby facilitating the training of deeper and more effective models. Since its introduction by He *et al.* [47], ResNet has become a foundational architecture in modern deep learning.

As shown in Figure 5, the ResNet-18 architecture consists of an input layer, a sequence of convolutional layers organized into residual blocks, and a final fully connected classification layer. The network begins with a 7×7 convolutional layer with 64 filters and a stride of 2, followed by batch normalization, ReLU activation, and a 3×3 max-pooling layer. Subsequently, four stages of residual blocks are employed, with the number of filters doubling at each stage from 64 to 512. Each stage contains two residual blocks, and each block comprises two 3×3 convolutional layers, each followed by batch normalization and ReLU activation. The residual connections bypass these convolutional layers, ensuring stable gradient propagation during backpropagation.

Mathematically, each residual block learns a residual function such that the block output is given by $Y = F(X) + X$, where X is the input to the block and F denotes the nonlinear transformation performed by the stacked convolutional layers: $F(X) = ReLU(BN(XK_2)) \circ ReLU(BN(XK_1))$, where $\circ$ is the composition operator. After the final residual stage, a global average pooling layer reduces the spatial dimensions of the feature maps. The pooled features are then passed through a fully connected layer with three output nodes corresponding to the flawless, crack, and cold flow categories. Similar to the CNN model, the class probabilities are obtained using the softmax function: $\hat{p} = Softmax(W * GAP(Y) + b)$, where GAP denotes global average pooling, and W and b are the weight matrix and bias of the final fully connected layer, respectively. In this study, the pre-trained ResNet-18 model is fine-tuned using the pre-processed training dataset to optimize performance for the surface defect detection task.

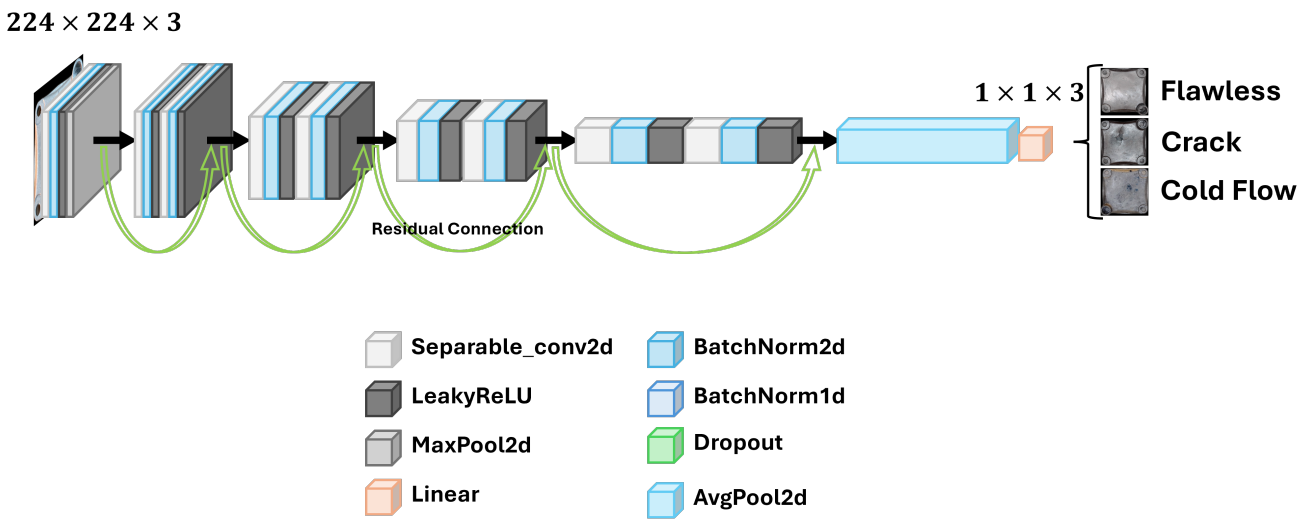


Figure 5. ResNet-18 architecture.

4.3. *Pre-trained vision transformer*

The third architecture employed in this study is a pre-trained ViT model, originally trained on the ImageNet-21K dataset. The ViT architecture is derived from the Transformer model, which was first introduced in 2017 as a groundbreaking approach for natural language processing (NLP) tasks such as text classification and machine translation [48]. A key innovation of the Transformer is the self-attention mechanism, which enables effective modeling of long-range dependencies, as well as the use of positional encodings to preserve the sequential order of input tokens. This architecture was later adapted for image analysis tasks, resulting in the ViT framework 6.

The Transformer follows an encoder–decoder architecture; however, for image classification tasks, only the encoder component is utilized. As illustrated in Figure 6, ViT treats an image as a sequence of visual tokens, enabling global context modeling across the entire image. Specifically, the input image is divided into a set of non-overlapping patches, each of which is flattened and linearly projected into an embedding space, effectively transforming the image into a sequence similar to time-series data.

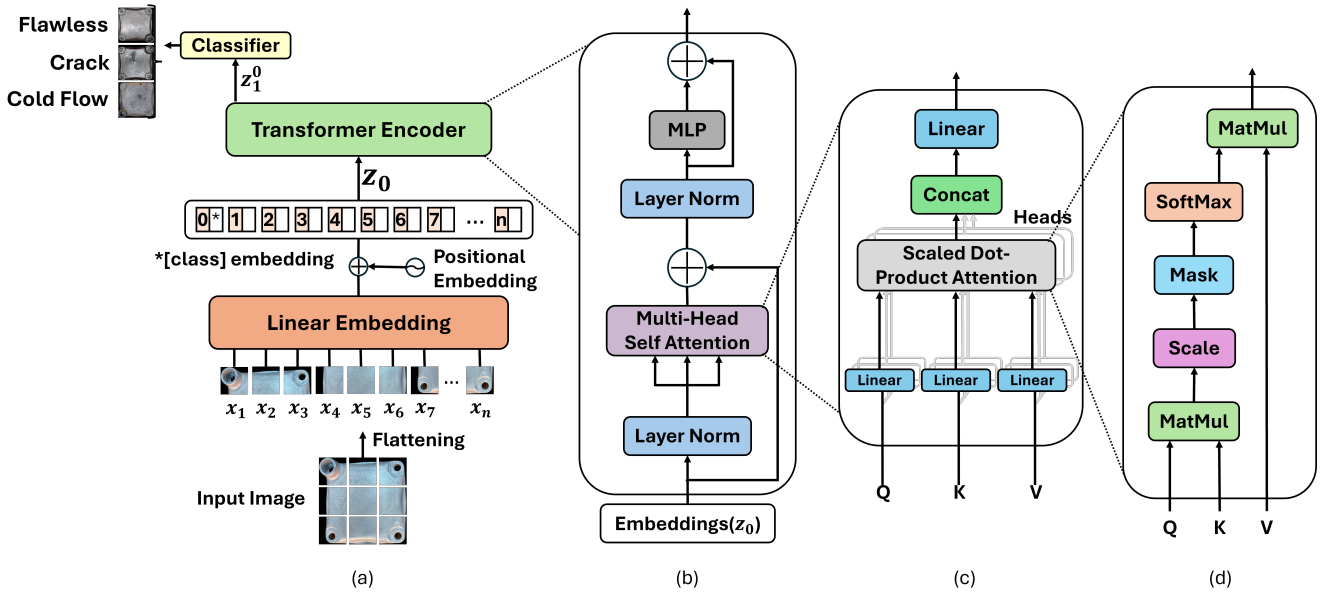


Figure 6. ViT architecture: **(a)** Mainframe; **(b)** Transformer encoder module; **(c)** Multi-head self-attention; **(d)** Self-attention.

Formally, let the input image be decomposed into N patches $x_1, x_2, \ldots, x_N$. Each patch is projected using a learnable linear mapping E, and a learnable classification (CLS) token cls is prepended to the sequence. Positional embeddings E_{pos} are added to retain spatial information. The resulting input to the encoder is given by $z_0 = [x_{cls}, x_1E, x_2E, \ldots, x_NE] + E_{pos}$, where E denotes the projection matrix and E_{pos} represents positional embeddings. The encoder consists of multiple Transformer encoder blocks, each composed of a Multi-Head Self-Attention (MHSA) module followed by a Feed-Forward Neural Network (FFN), as shown in Figure 6b. The self-attention mechanism captures relationships between different image patches. For each attention head, the input embeddings are linearly transformed into query (Q), key (K), and value (V) matrices. The attention operation is defined as $Attention(Q,K,V) = Softmax(\frac{QK^T}{\sqrt{d}})V$, where d is the embedding dimension. The outputs from all attention heads are concatenated and projected back into the original embedding space. To

stabilize training and facilitate deeper architectures, residual connections and layer normalization are applied around both the MHSA and FFN modules: $z' = LN(z + MHSA(z)), z^{l+1} = LN(z' + FFN(z'))$. After passing through the final encoder block, the output corresponding to the classification token is extracted and fed into a linear classification layer, followed by a softmax activation to obtain class probabilities: $\hat{p} = Softmax(Wz_{cls} + b)$, where W and b are the learnable weight matrix and bias of the final linear layer, respectively. The resulting probabilities correspond to the flawless, crack, and cold flow categories.

4.4. Ensemble

An ensemble model combines the predictions or predicted probabilities of multiple individual models to produce a single, consolidated output. By leveraging the complementary strengths of its constituent models, an ensemble approach can achieve higher accuracy and greater robustness than any single model alone. A key component of ensemble methods is the voting mechanism, which determines how individual model outputs are aggregated to generate the final prediction. Let the predicted class labels from the CNN, ResNet-18, and ViT models be denoted as $\hat{y}^{(1)}, \hat{y}^{(2)}, \hat{y}^{(3)}$, respectively. Two common voting strategies are soft voting and hard voting. Soft voting averages the predicted class probabilities from all models and assigns the final label to the class with the highest averaged probability. In contrast, hard voting adopted in this study selects the final prediction based on the majority class label predicted by the individual models. Formally, the hard voting ensemble prediction is given by $\hat{y}_{ens} = mode(\hat{y}^{(1)}, \hat{y}^{(2)}, \hat{y}^{(3)})$. In our study, both voting strategies are employed. The voting strategy reduces the influence of misclassifications from any single model and enhances prediction stability across different data samples. Moreover, by effectively aggregating the biases and variances of the individual models, the ensemble approach mitigates overfitting and improves generalization performance. Overall, the ensemble framework provides a systematic means of capitalizing on the strengths of multiple architectures while compensating for their individual weaknesses, resulting in more reliable and accurate classification outcomes.

5. Results and discussion

The four deep learning architectures explained in Section 4. are implemented in Python v3.13.7 using PyTorch 2.8.0 [49]. Experiments are conducted on a workstation that runs the Windows 11 Education operating system and is equipped with an AMD Ryzen 7 9700X 8-Core 3.8 GHz CPU, 128 GB RAM, and an NVIDIA GeForce RTX 5090 32 GB GPU.

We begin by identifying the optimal model for each architecture through an evaluation of different combinations of hyperparameter values—a process known as grid search. Grid search is a widely used technique in machine learning for fine-tuning model performance, offering advantages such as automation of the tuning process and flexibility in parameter configuration. Unlike random search, which randomly samples hyperparameter combinations within a predefined time limit, grid search systematically explores the entire defined parameter space, often leading to more consistent and superior performance outcomes.

When conducting the grid search, we focus on three key hyperparameters: batch size, number of epochs, and learning rate. The batch size defines the number of training samples processed before updating the model's

internal parameters (weights and biases). Due to random-access memory (RAM) limitations, batch sizes of 32 and 64 are evaluated. The epoch represents one complete iteration over the entire training dataset, determining how many times the model revisits all training examples; we test 20, 30, and 40 epochs. The learning rate controls the magnitude of updates to the model's parameters during training, influencing how quickly or slowly the model learns. We examine four learning rate values: 0.1, 0.01, 0.001, and 0.0001. The grid search results, summarized in Table 3, indicate that all architectures achieve their best performances with a batch size of 64 and a learning rate of 0.0001. The optimal number of epochs is 30 for the CNN and ResNet architectures, and 40 for the ViT.

Table 3. Hyperparameters with best performance for each architecture.

Architecture	Batch Size	Epochs	Learning Rate
CNN	64	30	0.0001
ResNet-18	64	30	0.0001
ViT	64	40	0.0001

Using the optimized hyperparameters, we performed ten independent training–testing runs for each model and evaluated performance using the area under the Receiver Operating Characteristic curve (AUC). Figure 7 summarizes the results with box plots across CNN, ResNet, and ViT. Figure 7 a–c present one-vs-rest AUC distributions for the "flawless," "crack," and "cold flow" classes, enabling class-specific comparison, while Figure 7d,e reports the micro-average and macro-average AUC results to reflect overall performance across categories. Across all classes and evaluation metrics, ResNet consistently demonstrates the highest and most stable AUC performance, with tightly clustered results and minimal variability across runs. CNN achieves similarly strong performance, though with slightly greater spread in AUC values. In contrast, the ViT model exhibits lower median AUC scores and noticeably higher variability, particularly for the "crack" and "cold flow" classes, indicating less stable and less effective classification behavior. The micro- and macro-average AUC results further confirm this trend: ResNet achieves the best overall and most reliable performance, CNN follows closely, and ViT trails behind in both accuracy and consistency. Overall, these results highlight the robustness and reliability of ResNet and CNN for defect classification under repeated experimental conditions.

We further compare the ROC curves of CNN, ResNet, and ViT using multiple AUC evaluation methods. For each method, the testing run that achieved the highest AUC is selected to represent the model's best performance, ensuring a fair and consistent comparison of its optimal discriminative capabilities. The ROC curve illustrates the trade-off between the true positive rate and the false positive rate, while the AUC, ranging from 0 to 1, quantifies overall classification effectiveness. A larger AUC, therefore, indicates stronger discriminative performance. As shown in Figure 8, all three models achieve strong classification performance across the "flawless," "crack," and "cold flow" classes, as well as in the micro- and macro-average evaluations. CNN and ResNet consistently produce ROC curves that closely approach the top-left corner, with AUC values near one, indicating excellent and stable performance across categories. CNN shows a slight advantage in several cases, particularly for the "flawless" and "crack" classes and in the macro-average evaluation, while ResNet demonstrates nearly identical and highly reliable results throughout. In contrast, the ViT model exhibits comparatively lower AUC values

and less steep ROC curves, especially for the “crack” and “cold flow” classes, suggesting weaker and less consistent discriminative ability. Overall, the ROC analysis confirms that CNN and ResNet outperform ViT in both class-specific and aggregated evaluations, with CNN achieving the best overall performance and ResNet showing exceptional stability.

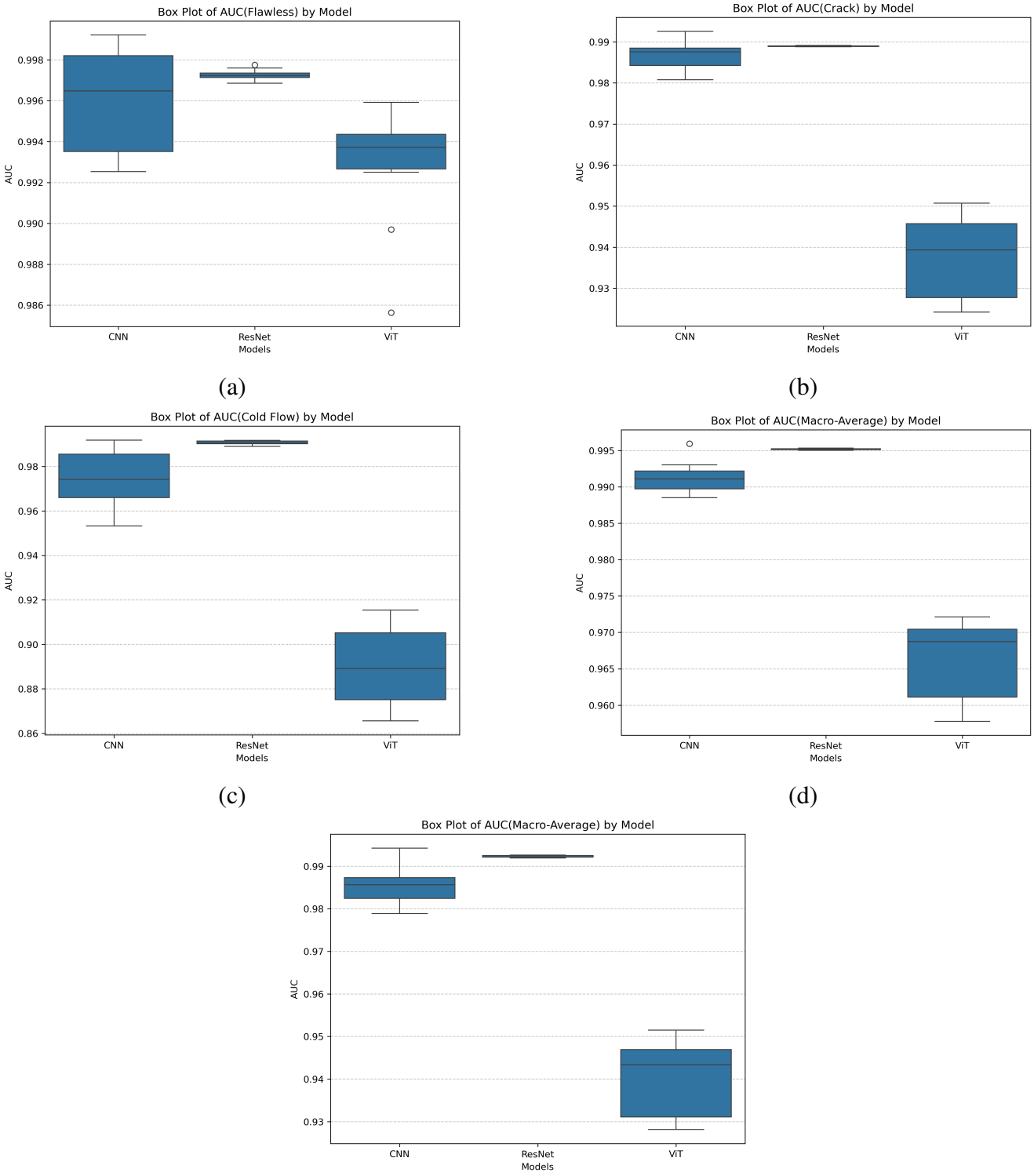


Figure 7. Box plots of AUC by models: **(a)** Flawless; **(b)** Crack; **(c)** Cold flow; **(d)** Micro-average; **(e)** Macro-average.

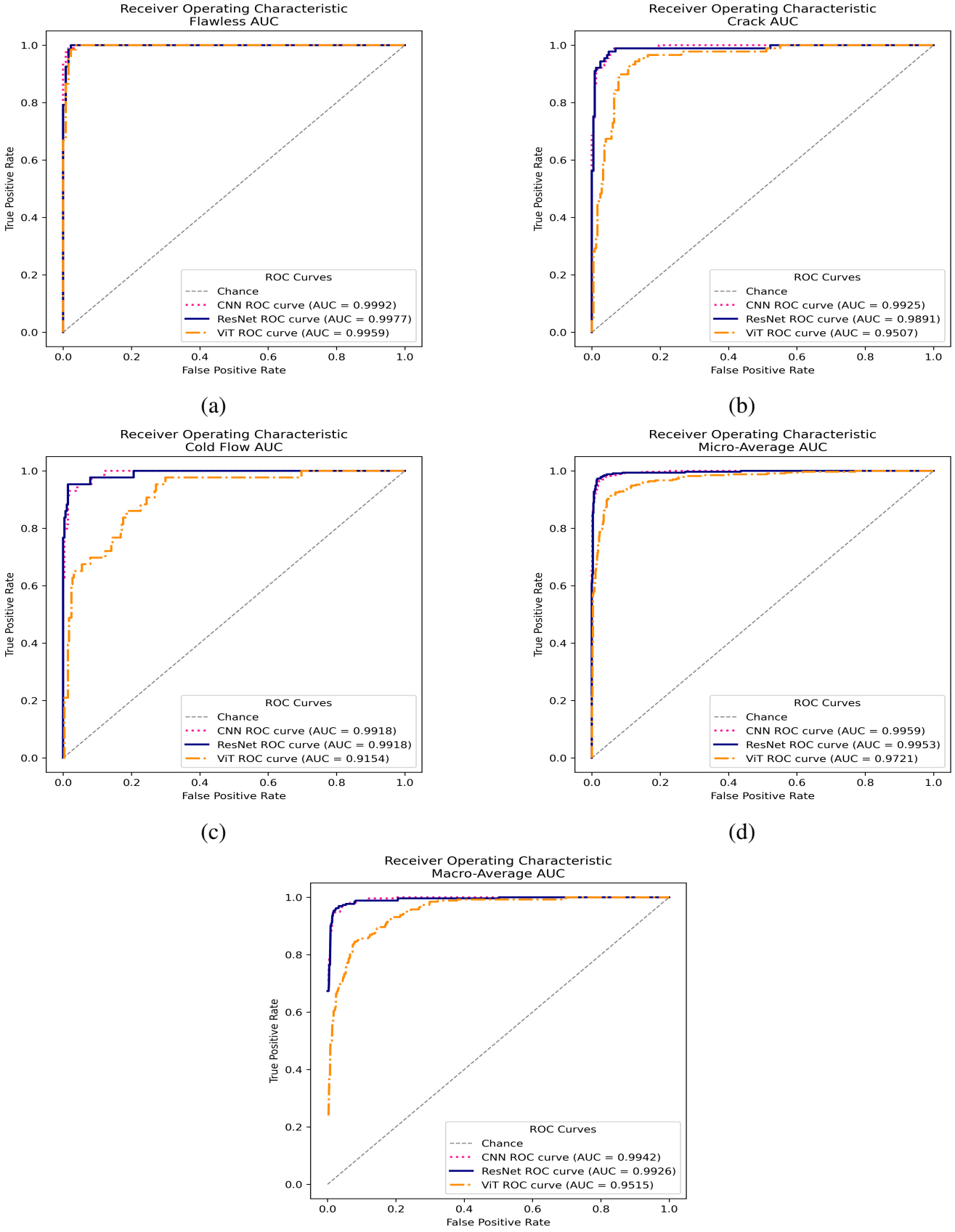


Figure 8. ROC curves by models: **(a)** Flawless; **(b)** Crack; **(c)** Cold flow; **(d)** Micro-average; **(e)** Macro-average.

In the end, we compare the performances of the three architectures using the four metrics below.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

where accuracy represents the correct prediction rate among all categories, TP (short for true positive) is the number of defective cases that are correctly predicted as defective, TN (true negative) is the number of flawless cases that are correctly predicted as flawless, FP (false positive) is the number of flawless cases that are incorrectly predicted as defective, and FN (false negative) is the number of defective cases that are incorrectly predicted as flawless. F1-score describes the harmonic mean of precision and recall and measures the reliability of the model from both angles equally. The ranges of four metrics are $[0, 1]$, and each metric indicates better performance when it is closer to 1.

In multi-class classification, most models estimate the conditional probability of each category given the input features and assign the label with the highest probability. Table 4 summarizes the performance of CNN, ResNet-18, ViT, and the Ensemble model using accuracy, precision, recall, and F1-score, where the bracketed values denote the minimum and maximum across runs and the parenthesized values indicate the mean. Consistent with the AUC and ROC analyses, CNN and ResNet-18 demonstrate the highest and most stable performance across all metrics, with ResNet-18 showing a slight advantage in recall and F1-score, reflecting better sensitivity and balanced classification ability. In contrast, the ViT model yields noticeably lower scores and greater variability, indicating weaker generalization and less reliable detection performance. The Ensemble model achieves the best overall results, outperforming all individual models in every metric, which highlights the benefit of combining complementary model predictions to enhance robustness and predictive reliability.

To further assess statistical significance, two-sample t-tests were conducted between the accuracies obtained from ten runs of each individual model and those of the ensemble model. The resulting p-values for CNN, ResNet-18, and ViT compared to the ensemble (hard voting) are 0.000136, 0.322121, and 0, respectively. The p-values for the three models compared to the ensemble (soft voting) are 0.001742, 0.193422, and 0, respectively. At a significance level of $\alpha = 0.05$, these results indicate that the ensemble model performs statistically significantly better than both CNN and ViT, while its performance is not significantly different from that of ResNet-18.

The comparatively weaker performance of ViT can be attributed to its architectural characteristics. Vision Transformers rely heavily on large-scale training data to effectively learn global self-attention patterns and lack the strong inductive biases of convolutional networks, such as locality and translation invariance. In datasets of limited size, such as defect images with subtle local features, ViT may struggle to capture fine-grained spatial details that CNN-based architectures inherently model more effectively

through convolutional kernels. As a result, CNN and ResNet-18 are better suited to extracting localized defect patterns, leading to more stable and accurate classification. The ensemble approach further mitigates individual model weaknesses by aggregating their predictions through hard voting, reducing bias and variance, and ultimately producing the most reliable performance across all evaluation metrics.

Due to differences in architectural complexity, the training time varies substantially across models. As shown in Table 5, the CNN requires the least computational time, completing one training run in 4 min 49 s. ResNet-18 requires nearly twice as long, with a training time of 8 min 20 s per run. In contrast, the ViT model is considerably more computationally demanding, requiring 1 h 10 min 57 s for a single run, approximately 18 times longer than the CNN. While ViT incurs significantly higher training cost, training is performed offline, and real-time deployment depends on inference, which remains efficient. Despite lower standalone performance, ViT contributes architectural diversity to the ensemble, improving robustness without prohibitive inference overhead.

Table 4. Model classification performance.

Method	Accuracy	Precision	Recall	F1-Score
CNN	[0.9581, 0.9731] (0.9665)[a]	[0.9382, 0.9624] (0.9528)	[0.9256, 0.9543] (0.9435)	[0.9317, 0.9582] (0.9480)
ResNet-18	[0.9611, 0.9701] (0.9692)	[0.9343, 0.9509] (0.9492)	[0.9454, 0.9545] (0.9536)	[0.9384, 0.9519] (0.9506)
ViT	[0.8503, 0.9042] (0.8784)	[0.7421, 0.8468] (0.7918)	[0.7395, 0.8118] (0.7788)	[0.7449, 0.8178] (0.7824)
Ensemble (Hard)	[0.9641, 0.9760] (0.9707)	[0.9518, 0.9646] (0.9577)	[0.9350, 0.9620] (0.9505)	[0.9432, 0.9631] (0.9539)
Ensemble (Soft)	[0.9671, 0.9760] (0.9710)	[0.9518, 0.9646] (0.9582)	[0.9388, 0.9620] (0.9511)	[0.9451, 0.9631] (0.9545)

[a] mean value within the parenthesis.

Table 5. Model computational performance.

Method	Running Time for 1 Run	Running Time for 10 Runs
CNN	4 min 49 s	47 min 37 s
ResNet-18	8 min 20 s	1 h 21 min 57 s
ViT	1 h 10 min 57 s	11 h 52 min 49 s

6. Conclusion

In this paper, we propose a framework for automating the detection of surface defects in aluminum die-cast gas meter lids. Specifically, we implemented three state-of-the-art deep learning models, including a conventional CNN, a pre-trained ResNet-18, and a pre-trained ViT. An ensemble of these models is also constructed. We collected a large real-world dataset, pre-processed the images, and used grid search for hyperparameter tuning. Ten experiments were performed for each architecture. An extensive comparison shows that all the architectures achieved high performance in accuracy, precision, recall, and F1-score. The fine-tuned ResNet-18 model performs slightly better than the CNN model on precision, recall,

and F1-score, while the fine-tuned ViT model underperformed compared to CNN and ResNet-18. The ensemble model takes advantage of all three models and generates better accuracy. The paired t-tests indicate that the ensemble model performs significantly better than both the CNN and ViT models. The superior performances indicate that it is feasible to implement the framework for automatically detecting surface defects in aluminum die-cast lids for gas meters. We believe these state-of-the-art models have the potential to be broadly studied for surface defect detection in other aluminum die-cast products.

We acknowledge the following limitations in this study. First, this study only tackles the detection of surface defects without considering internal defects. As discussed in Section 1, internal defects such as porosity may also jeopardize the functionalities of many aluminum die-cast alloys and should be inspected. However, surface defects are the major types of defects in aluminum die-cast gas meter lids, so this study is still promising in improving the current inspection practice for gas meter lids. In addition, X-ray and ultrasonic scans can be utilized to identify internal defects [14,23,41]. The framework and the models developed in this study may be extended to such applications, contingent upon the availability of corresponding X-ray or ultrasonic image data. Second, this study addresses a multiclass classification task with flawless and two types of surface defects (*i.e.,* crack and cold flow), while there are possibly other types of surface defects, particularly for other aluminum die-cast products. For alternative applications, such as surface defects of aluminum die-cast automotive parts, the framework and architectures developed in this study can be adapted; however, the collection and appropriate labeling of a new dataset may be needed. Third, this study belongs to acceptance sampling quality control efforts, while online quality control solutions may better address the problems and improve productivity. The approach in this study is accessible and could be easily implemented as an in-situ quality control application to realize online quality control.

Our future work includes collecting a larger and more diverse dataset (labeled with more different types of surface defects) and revising our models. In addition, we plan to implement deep learning models as an in-situ quality control application to realize online quality control for real-time cause-and-effect analysis and closed-loop manufacturing systems.

Data availability statement

The data or datasets generated or analyzed in this study are available in GitHub at https://github.com/Alexruoyun/Aluminum-Die-Casting-Surface-Defect-Detection.

Declaration of generative AI and AI-assisted technologies

During the preparation of this manuscript, the authors used generative AI tools only to improve language and readability. Specifically, the authors used ChatGPT for language polishing only in the entire manuscript. The authors take full responsibility for the content of the manuscript.

Acknowledgments

This work was funded by the Competitive Research Grant Program from South Dakota Board of Regents.

Authors' contribution

Wuyang Qian: methodology, writing—review and editing, writing—original draft preparation; Olayinka Ayorinde: writing—review and editing, writing—original draft preparation; Suhao Chen: conceptualization, methodology, writing—original draft preparation; Lin Guo: conceptualization, methodology, writing—review and editing, writing—original draft preparation, funding acquisition; Dean Jensen: conceptualization, methodology, writing—original draft preparation. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Kaye A, Street A. Die casting metals and alloys. In *Die Casting Metallurgy: Butterworths Monographs in Materials*, 1st ed. London: Butterworth Scientific, 1982. pp. 10–16.

[2] Kapranos P, Brabazon D, Midson S, Naher S, Haga T. Advanced casting methodologies: inert environment vacuum casting and solidification, die casting, compocasting, and roll casting. In *Comprehensive Materials Processing*, 1st ed. Amsterdam: Elsevier, 2014, pp. 3–37.

[3] Zheng P, Wang H, Sang Z, Zhong RY, Liu Y, *et al.* Smart manufacturing systems for Industry 4.0: conceptual framework, scenarios, and future perspectives. *Front. Mech. Eng.* 2018, 13(2):137–150.

[4] American Foundry Society. Census of world casting production: fewer castings made in 2020. *Mod. Cast.* 2021, 111(12):26–28.

[5] American Foundry Society. Census of world casting production. *Mod. Cast.* 2021, 121(1):27.

[6] NADCA. State of the industry archive. 2021. Available: https://www.diecasting.org/Web/About/State_of_the_Industry_Archive.aspx (accessed on 31 January 2026).

[7] Folk J. U.S. aluminum casting industry—2019. 2019. Available: https://www.diecasting.org/archive/dce/71916.pdf (accessed on 31 January 2026).

[8] Midson S. Report on the 2014 die casting benchmarking survey part 2 of 3: operations, in report on the 2014 die casting benchmarking survey. 2014, pp. 1–3. Available: https://education.diecasting.org/education/online/enrol/index.php?id=237 (accessed on 31 January 2026).

[9] Frayman Y, Zheng H, Nahavandi S. Machine vision system for automatic inspection of surface defects in aluminum die casting. *J. Adv. Comput. Intell. Intell. Inform.* 2006, 10(3):281–286.

[10] American Foundry Society. Metalcasters quarterly outlook survey. 2024. Available: https://www.afsinc.org/metalcasters-quarterly-outlook-survey (accessed on 31 January 2026).

[11] Wang J, Ma Y, Zhang L, Gao RX, Wu D. Deep learning for smart manufacturing: methods and applications. *J. Manuf. Syst.* 2018, 48:144–156.

[12] Tao F, Qi Q, Liu A, Kusiak A. Data-driven smart manufacturing. *J. Manuf. Syst.* 2018, 48:157–169.

[13] Nikolic F, Stajduhar I, Canadija M. Casting microstructure inspection using computer vision: dendrite spacing in aluminum alloys. *Metals* 2021, 11(5):756.

[14] García Pérez A, Gómez Silva M, de La Escalera Hueso A. Automated defect recognition of castings defects using neural networks. *J. Nondestr. Eval.* 2022, 41(1):11.

[15] Moon G, Seo HI, Seo DH, Lee E. Application of the cnvolutional neural network for classification of the aluminum alloys based on their microstructural characteristics. *JOM* 2023, 75(11):4858–4867.

[16] Nikolic F, Stajduhar I, Canadija M. Casting defects detection in aluminum alloys using deep learning: a classification approach. *Int. J. Metalcast.* 2023, 17(1):386–398.

[17] Hou M, Dong H, Ji X, Zou W, Xia X, *et al.* I-DCGAN and TOPSIS-IFP: a simulation generation model for radiographic flaw detection images in light alloy castings and an algorithm for quality evaluation of generated images. *China Foundry* 2024, 21(3):239–247.

[18] Azimi SM, Britz D, Engstler M, Fritz M, Mücklich F. Advanced steel microstructural classification by deep learning methods. *Sci. Rep.* 2018, 8(1):2128.

[19] Habibpour M, Gharoun H, Tajally A, Shamsi A, Asgharnezhad H, *et al.* An uncertainty-aware deep learning framework for defect detection in casting products. *arXiv* 2021, arXiv:2107.11643.

[20] Stephen O, Madanian S, Nguyen M. A robust deep learning ensemble-driven model for defect and non-defect recognition and classification using a weighted averaging sequence-based meta-learning ensembler. *Sensors* 2022, 22(24):9971.

[21] Gupta R, Anand V, Gupta S, Koundal D. Deep learning model for defect analysis in industry using casting images. *Expert Syst. Appl.* 2023, 232:120758.

[22] Chowdhury A, Kautz E, Yener B, Lewis D. Image driven machine learning methods for microstructure recognition. *Comput. Mater. Sci.* 2016, 123:176–187.

[23] Wu B, Zhou J, Yang H, Huang Z, Ji X, *et al.* An ameliorated deep dense convolutional neural network for accurate recognition of casting defects in X-ray images. *Knowledge-Based Syst.* 2021, 226:107096.

[24] Cavaliere G. Comparative use of systems to detect surface defects in die-cast components using advanced vision systems applying artificial intelligence. Doctoral Thesis, Free University of Bozen-Bolzano, 2024.

[25] Liu D, Du Y, Chai W, Lu C, Cong M. Digital twin and data-driven quality prediction of complex die-casting manufacturing. *IEEE Trans. Ind. Inf.* 2022, 18(11):8119–8128.

[26] Zhang S, Li H, Ren P, Peng T, Meng X. A detection method for small casting defects based on bidirectional feature extraction. *Sci. Rep.* 2025, 15(1):6362.

[27] Uyan TC, Otto K, Silva MS, Vilaça P, Armakan E. Industry 4.0 foundry data management and supervised machine learning in low-pressure die casting quality improvement. *Int. J. Metalcast.* 2023, 17(1):414–429.

[28] Liu W, Tang R, Peng T. An IoT-enabled approach for energy monitoring and analysis of die casting machines. *Procedia CIRP* 2018, 69:656–661.

[29] Gao Y, Gao L, Li X. A two-stage focal transformer for human–robot collaboration-based surface defect inspection. *J. Manuf. Sci. Eng.* 2023, 145(12):121004.

[30] Ayorinde O, Qian W, Chen S, Jensen D, Guo L. Surface defect classification in aluminum die-cast gas meter lids using machine vision system and deep learning techniques. In *Proceedings of the ASME 2025 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2025)*, Anaheim, USA, August 17–20, 2025, pp. 1–12.

[31] Yousef N, Sata A. Intelligent inspection for evaluating everity of surface defects in investment casting. *J. Adv. Manuf. Syst.* 2024, 23(1):215–225.

[32] Jiang H, Zhang X, Tao C, Ai S, Wang Y, *et al.* Casting defect region segmentation method based on dual-channel encoding–fusion decoding network. *Expert Syst. Appl.* 2024, 247:123254.

[33] Shahane S, Mujumdar S, Kim N, Priya P, Aluru NR, *et al.* Simulations of die casting with uncertainty quantification. *J. Manuf. Sci. Eng.* 2019, 141(4):041003.

[34] Cavaliere G, Lanz O, Borgianni Y, Savio E. Deep learning-supported machine vision-based hybrid system combining inhomogeneous 2D and 3D data for the identification of surface defects. *Prod. Manuf. Res.* 2024, 12(1):2378199.

[35] Liu Y, Liu J, Li C, Xi R, Li W, *et al.* Anomaly detection and generation with diffusion models: a survey. *arXiv* 2025, arXiv:2506.09368.

[36] Schueller A, Saldaña C. Indirect tool condition monitoring using ensemble machine learning techniques. *J. Manuf. Sci. Eng.* 2023, 145(1):011006.

[37] Mo W, Luo X, Zhong Y, Jiang W. Image recognition using convolutional neural network combined with ensemble learning algorithm. *J. Phys. Conf. Ser.* 2019, 1237(2):022026.

[38] Lu Z, Sun H, Xu Y. Adversarial robustness enhancement of UAV-oriented automatic image recognition based on deep ensemble models. *Remote Sens.* 2023, 15(12):3007.

[39] Liu J, Du Y, Yang K, Wu J, Wang Y, *et al.* Edge-cloud collaborative computing on distributed intelligence and model optimization: a survey. *IEEE Commun. Surv. Tutor.* 2026, 28:5049–5080.

[40] Yang J, Liu B, Shu D, Li H, Yang Q, *et al.* Effect of casting pressure on porosity, microstructure, and mechanical properties of large die casting aluminum alloy parts. *Int. J. Metalcast.* 2026, 20(2):1066–1080.

[41] Bosse S, Lehmhus D, Kumar S. Automated porosity characterization for aluminum die casting materials using X-ray radiography, synthetic X-ray data augmentation by simulation, and machine learning. *Sensors* 2024, 24(9):2933.

[42] Bhat V, Balikai VG, Revankar P, Gorwar M. A review on quality assurance in aluminium die casting through deep learning-based defect detection. In *Proceedings of E3S Web of Conferences*, Tomsk, Russian Federation, October 22–24, 2024, p. 01010.

[43] Ryadnenko D. rembg: a tool to remove image backgrounds. Available: https://pypi.org/project/rembg/ (accessed on 5 December 2024).

[44] Clark A, Contributors. Pillow: the friendly PIL fork. Available: https://pypi.org/project/pillow/ (accessed on 5 December 2024).

[45] Contributors of OpenCV. opencv-python: Python bindings for OpenCV. Available: https://pypi.org/project/opencv-python/ (accessed on 5 December 2024).

[46] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc. IEEE* 1998, 86(11):2278–2324.

[47] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, June 27–30, 2016, pp. 770–778.

[48] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, *et al.* An image is worth 16×16 words: transformers for image recognition at scale. *arXiv* 2020, arXiv:2010.11929.

[49] PyTorch Team. torch: PyTorch deep learning framework. Available: https://pypi.org/project/torch/ (accessed on 5 December 2024).