The International Journal of Advanced Manufacturing Technology (2024) 134:97–111
https://doi.org/10.1007/s00170-024-14069-7

ORIGINAL ARTICLE

Exploring deep fully convolutional neural networks for surface defect detection in complex geometries

Daniel García Peña[1,2] · Diego García Pérez[1] · Ignacio Díaz Blanco[1] · Jorge Marina Juárez[2]

Received: 7 February 2024 / Accepted: 22 June 2024 / Published online: 16 July 2024

Abstract

In this paper, we propose a machine learning approach for detecting superficial defects in metal surfaces using point cloud data. We compare the performance of two popular deep learning architectures, multilayer perceptron networks (MLPs) and fully convolutional networks (FCNs), with varying feature sets. Our results show that FCNs (F1=0.94) outperformed MLPs (F1=0.52) in terms of precision, recall, and F1-score. We found that transfer learning with pre-trained models can improve performance when the amount of available data is limited. Our study highlights the importance of considering the amount and quality of training data in developing machine learning models for defect detection in industrial settings with 3D images.

Keywords Image segmentation · Defect detection · Convolutional neural networks · Deep learning · Machine learning · 3D images · Fully convolutional networks · Transfer learning

1 Introduction

Nowadays, quality controls have become a key aspect in the manufacturing industry towards the improvement of the production processes and reduce manufacturing costs. In the specific case of metal parts manufacturing, the absence of functional and aesthetic defects must be ensured in 100% of the production prior to delivery. This has led to a great interest in the development and implementation of accurate and computationally efficient quality control systems in production lines [1, 2].

Traditionally, quality controls in production have been conducted through the manual measurement and inspection of random samples. These procedures typically depend on human operators, involve extended inspection times, and render the examination of the entire production impractical.

These procedures are still present when the inspection requires contact between the measuring equipment and the steel component being studied. Contact measurements have to be performed at a relatively slow pace to avoid collisions that could deteriorate the equipment or the production [3, 4].

The participation of human inspectors is also undesired (due to fatigue, cost, subjectivity, etc.). With the upcoming of non-contact measurement methods, such as *ultrasound* [5], *machine vision* [6], or some *interferometric techniques* [7], the measurement speed limitation and the human intervention has been significantly reduced. As a result, it is now possible to perform a comprehensive inspection of production processes.

However, data from non-contact sensors must be accurately and efficiently processed to inspect the entire production. Computer vision techniques have been widely used to process measurements in automated inspection systems [8, 9]. Typically, the workflow of computer vision inspection systems [10] consists on (1) image acquisition, (2) image processing, (3) feature extraction, and (4) decision-making stage. Traditionally, handcrafted descriptors such as size, position, and edge detection are defined as features, and decision-making algorithms classify each sample based

✉ Diego García Pérez
garciaperdiego@uniovi.es

Daniel García Peña
danigarciape@gmail.com

Ignacio Díaz Blanco
idiaz@uniovi.es

Jorge Marina Juárez
jmarina@cinsystems.es

[1] Department of Electrical Engineering, University of Oviedo, Gijón 33204, Asturias, Spain

[2] CIN Advanced Systems, Leonardo da Vinci 20, Gijón 33211, Asturias, Spain

on the extracted features. Statistical models [9, 11], fuzzy systems [12, 13], and classic *machine learning* (ML) models such as shallow *neural networks* (NN) [14] or *decision trees* (DT) [15] have been proposed as decision-making methods.

While these methods offer simplicity and effectiveness under certain conditions, they significantly depend on the feature extraction. Designing appropriate and reliable features for decision-making algorithms is a difficult and time-consuming process highly conditioned by domain knowledge of the problem, uniform backgrounds, and invariant positions of the objects through images.

The specific industrial context of this research focuses on the manufacturing casting processes for the automotive industry, which is known for its restrictive quality tolerances. Although this study is centered on casting, it is important to acknowledge that other fields, such as body-in-white (BIW) [16] inspections, play a significant role in automotive manufacturing. While BIW inspections pertain to the assembly and inspection of vehicle bodies, which involves different procedures and defect types compared to casting, the methodologies and technologies developed in this research could potentially be adapted to BIW and other manufacturing processes in the future.

In our specific industrial context, castings exhibit a diverse range of positions and structures, while defects manifest as subtle and localized irregularities, as exemplified in Fig. 1. Traditionally handcrafted feature extraction algorithms encounter significant challenges in generating an effective feature representation [17], particularly when dealing with intricate scenarios characterized by varying casting structures, flexible positioning, and small, isolated defects. When conventional 2D imaging is insufficient, the adoption of 3D sensors becomes essential to capture a comprehensive profile of the surface and identify non-compliant material [18]. The transition from 2D cameras to 3D cameras finds strong justification in the domain of quality control for casting components [19]. Particularly, for surface defects where evaluating the surface roughness is crucial in the decision-making process, conventional 2D cameras cannot adequately capture the roughness information. However, 3D imaging provides depth information that enables the calculation of roughness and sphericity.

The outcome of 3D imaging is commonly addressed by 3D point cloud comparison [20]. 3D point cloud comparison involves the process of aligning and analyzing two sets of point cloud data—collections of points in a three-dimensional coordinate system—to assess their geometric similarities and differences. When defects are smaller than the acquisition resolution, 3D point cloud comparison proves inadequate. For instance, in welding of cast pieces [21], the dimensional tolerances exceed several tenths of a millimeter, while faults like bumps may only be only fractions of millimeter. This can cause cloud comparison methods to fail in detecting these subtle defects.

Given the challenges associated with manual feature extraction, data-driven models, particularly *deep learning* (DL) techniques [22], offer a convenient alternative. In the last decade, DL models have demonstrated their ability to learn meaningful features from raw data in various domains, such as *image segmentation* [23], *object detection* [24], and *medical image classification* [25]. However, DL models based on supervised learning [26] require large labeled datasets and incur high computational costs during training and inference, making their deployment in industrial processes, such as surface defect detection, challenging.

In addition, researchers are actively working on developing more efficient DL algorithms that can reduce the computational costs and increase the reliability of these models [27].

In this proposed study, we aim to investigate the effectiveness of well-known *convolutional neural networks* (CNNs) for feature extraction in the context of 3D image analysis. Specifically, we will explore the capabilities of prominent CNN architectures, including *VGG-16* [28], *ResNet* [29], and *U-Net* [30], in extracting relevant features from 3D images.

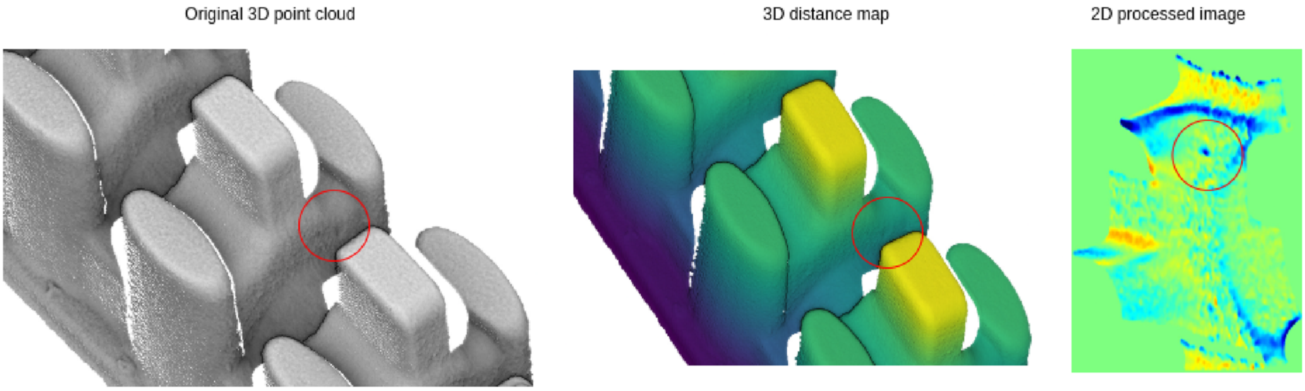


Fig. 1 Defect view example from original cloud, distance map and 2D processed image

These CNNs will be integrated with custom convolutional decoders to create *fully convolutional networks* (FCNs) [31].

To aid in the feature extraction task, we have manually selected 15 features based on covariance matrices. These features allow us to assess the impact of the amount of input information on the accuracy of detection.

Our study is structured into two stages to establish a baseline performance and gain insights into the advantages of *transfer learning* [32]. In the initial stage, we trained the FCN model from scratch, without utilizing any pre-trained weights. This approach provided us with a foundation for evaluating the model's performance from the ground up.

Subsequently, in the second stage, we leveraged transfer learning on the FCN model that exhibited the best performance. This involves using a smaller dataset and initializing the model with pre-trained weights. By comparing the outcomes of both approaches, we illustrated the benefits of incorporating transfer learning within the context of FCN models for image segmentation [32, 33].

In addition to introducing our custom FCN approach, we conducted a comparative analysis with a traditional *multilayer perceptron* (MLP). This MLP was trained using both pixel information and manually extracted features.

The remainder of the article is organized as follows: Sect. 2 describes related work; Sect. 3 describes the data acquisition systems. The different approaches are detailed in Sect. 4, and the results are compared in Sect. 5. Finally, Sect. 6 presents the conclusions.

2 Related work

ML techniques have been demonstrated to be effective in anomaly detection in various application domains, such as anomalous consumption detection in large buildings [34], fault detection in rotating machines [35], and structural health monitoring of large infrastructures [36].

In the field of quality control by image processing, most studies focus on defect detection using 2D images [37]. In [38], the authors proposed a multilevel methodology for binary classification of defective casting parts from X-ray images. Although, initially, the use of manually extracted features from these images was the most common approach due to the simplicity and speed of the algorithms [39], the selection of these features is a complex process. To address this challenge, some studies have investigated the application of DL methods for object recognition and defect detection without manual feature extraction, obtaining good results [40–42].

The development of deep CNNs has significantly advanced various image processing tasks. Jiang et al. [43] proposed a novel approach that combines convolutional and attention layers for the detection of casting defects in X-ray images. This work harnesses the power of CNNs to effectively detect and segment casting defects. Moreover, the capability of CNNs to excel in defect detection and segmentation tasks has been previously demonstrated in the study by Ferguson et al. [44].

Most studies in this field use deep architectures such as VGG [28] or ResNet [29], which are known for being able to extract optimal features, to classify samples as defective or not. On the other hand, fields such as image segmentation have gained momentum thanks to *fully convolutional networks* (FCNs) [31], which allow the simultaneous inference of several pixels of the image and take advantage of attributes such as parameter sharing [45] to optimize the computations during the inference stage. In particular, architectures such as *U-Net* [30] have proven to be able to perform pixel-level classification with high accuracy, and pre-trained networks like *VGG/ResNet* [28, 29] are commonly used for this task.

Regarding the utilization of 3D data, there have been several studies focusing on employing ML techniques for the detection of defects in industrial components. In their work [46], the authors proposed a deep learning-based approach for identifying defects in 3D-printed objects. Similarly, in [47], a framework based on ML was introduced to detect flaws in 3D objects. However, the utilization of 3D data for industrial defect detection in machine learning research is still lacking.

In this article, we introduce a new method for real-time defect detection using FCNs, enhancing defect resolution by leveraging 3D point clouds as the initial input. This work tackles defect detection in complex geometries within the casting processes for automotive industry, which has not been previously addressed. We merge renowned CNN architectures with point cloud data, adding a novel step to process 3D point clouds into 2D features to work with FCNs and achieving defect detection on tolerances much higher than prior works. Our solution operates in a fully convolutional format, accompanied by a tailored decoder. This configuration enables CNNs to act as encoders, delivering accurate real-time image segmentation for industrial casting components.

3 Data preprocessing

3.1 Optical system

As explained in the previous section, most of the research in this field has been carried out on RGB images. Despite the convenience for a computer to display such images, RGB color spaces have a series of drawbacks. The images captured in naturally occurring conditions or environments are prone to be affected by natural lighting intensity.

On the other hand, projects based on 3D laser triangulation are invariant to changes in light intensity and other environmental effects. The technique can resolve millimeter-size bumps and changes in depth from hundreds of meters away. In addition, laser triangulation excels at measuring at shorter distances, making it perfectly suited to fields such as metrology or surface inspection [48].

3D profiling suffers with surfaces that are particularly reflective or absorb light, so determining the wavelength and laser power according to the material is key.

Each collision is added to a point cloud, which refers to a set of data points in a coordinate system. In the standard cartesian coordinate system, points are defined in terms of X, Y, and Z coordinates. Point cloud data is then projected into a 2D space where pixel intensity defines the distance between the object and the camera.

3.2 Feature extraction

Our research aims to improve the accuracy of identifying structural patterns and defects in a piece using 3D features extracted from a distance image $\mathbf{X} \in \mathbb{R}^{l \times n}$ [49], being l and n the height and width of the image. To this end, we extract the covariance matrix for each pixel using a fixed window around it, as shown in Fig. 2, and exclude pixels with values far from the mean of the window to reduce noise in the resulting matrix. It is essential to address the border pixels to ensure consistent neighborhood calculations. Therefore, we introduce zero padding along the borders of the image.

According to Eq. 1, N is the total pixels in the neighborhood, and $\boldsymbol{\mu}_{P_i}$ is the mean value of the given pixels. Each element in the neighborhood, denoted as v, represents a vector that encompasses the pixel values within the given neighborhood. Within this vector, each entry corresponds to the values recorded in the three-dimensional coordinate system, where the x and y values are directly extracted from the row and column values in the image, respectively. Higher neighborhood values reduce the noise present in the features, at the cost of losing accuracy in the detection of small defects. Conversely, a small neighborhood value allows the detection of smaller defects, but introduces more noise to the features. The optimal neighborhood value should be calibrated for each application manually.

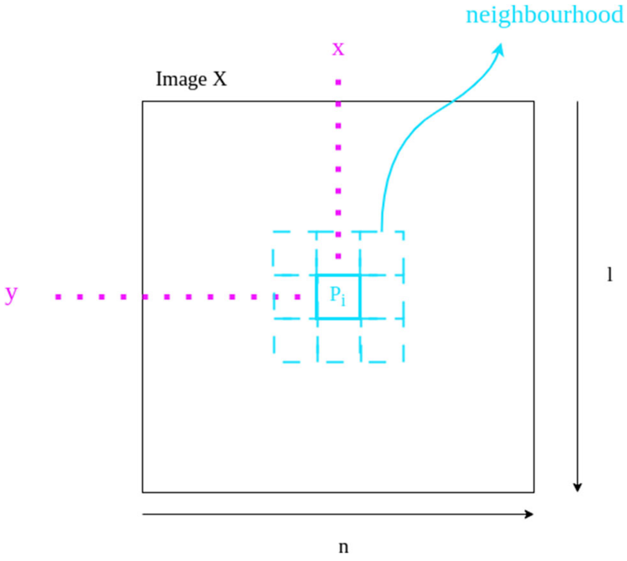


Fig. 2 Neighborhood extraction from distance image

$$\mathbf{C}_{P_i} = \frac{1}{N} \sum_{j=1}^{N} (v_j - \boldsymbol{\mu}_{P_i})(v_j - \boldsymbol{\mu}_{P_i})^T \tag{1}$$

From the covariance matrix, a total of m geometric *features* are extracted.

$$\mathcal{F}(\mathbf{X}^{(i)}) = \{\mathbf{F}_1^{(i)}, \mathbf{F}_2^{(i)}, \ldots, \mathbf{F}_m^{(i)}\} \tag{2}$$

where $\mathbf{F}_m^{(i)} \in \mathbb{R}^{l \times n}$ is the result of the *feature extraction operation* $\mathcal{F}(.)$ on the i-th image of the training set (see Fig. 3).

The features will be grouped according to the time involved in their computation. The final clusters are shown in Table 1.

- Level 0: Raw data. No additional processing needed.
- Level 1: Features extracted directly from the covariance matrix: *surface normals* (N_x,N_y,N_z) and *eigenvalues* (e_1,e_2,e_3).
- Level 2: Features derived from level 1 features: *anisotropy, sum of eigenvalues, entropy, sphericity, linearity, planarity, omnivariance, surface variation* [49].

4 Methodology

Once we have these features, we can train a classification function $C(.)$ (Eq. 3) that takes these features as input and

Table 1 Features clustering

Level 0	Level 1	Level 2
Distance	N_x N_y N_z e_1 e_2 e_3	Anisotropy Sum of eigenvalues Entropy Sphericity Linearity Planarity Omnivariance Surface variation

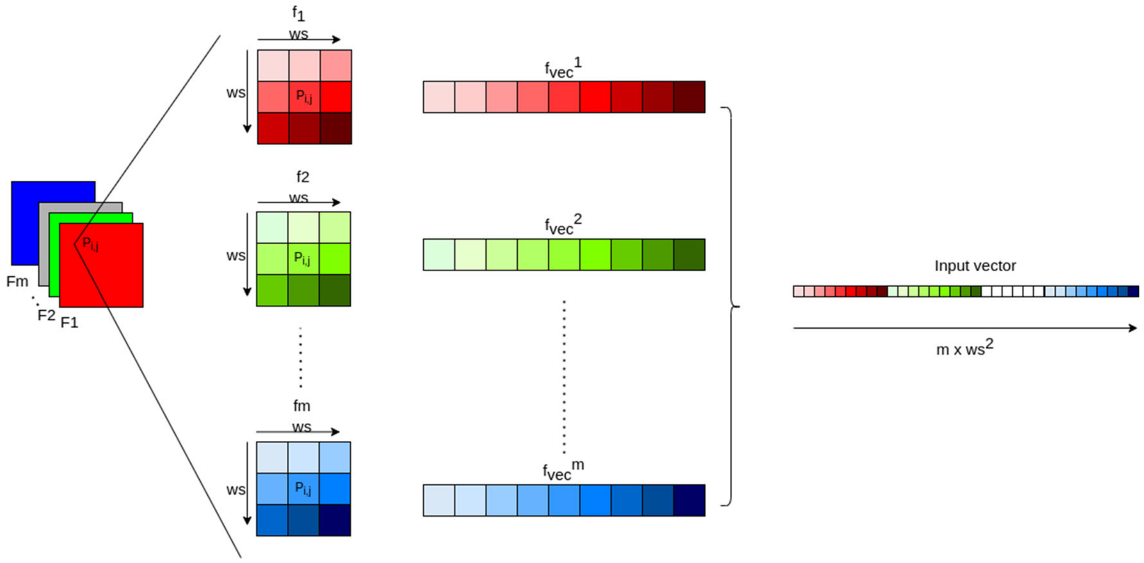


Fig. 3 2D to 1D conversion for m channels

outputs the probability of the image belonging to the faulty class.

$$C(\mathcal{F}(\mathbf{X}^{(i)})) = p(y = 1|\mathbf{X}^{(i)}) \tag{3}$$

In previous works [50, 51], the authors used vanilla fully connected neural networks to perform this task. However, fully connected networks based on pixel-wise classification may not be the best choice for image segmentation tasks, especially when dealing with large images. The pixel-by-pixel inference and feature computation can become a bottleneck in terms of time and resources. Therefore, we will explore the use of fully convolutional networks (FCNs) for this task.

Specifically, we will compare the performance of three popular FCN architectures, namely U-Net [30], VGG [28], and ResNet [29], in segmenting faulty areas in electronic component images. These architectures have been extensively used in various image segmentation tasks and have shown promising results.

4.1 Semantic image segmentation with a vanilla fully connected neural network

For a baseline comparison, a fully connected network as shown in Fig. 4 was trained to produce pixel-wise class prediction, so that the original 2D images must be converted to 1D vectors, lossing spatial information of the image. To solve this problem, information is extracted not only from the pixel, but also from a neighborhood, as shown in Fig. 3.

For each pixel P_i inside an image $\mathbf{F}_m^{(i)}$, a feature vector f_{vec} with length $ws \times ws$ corresponding to the values of $\mathbf{F}_m^{(i)}$ in $W_{P_i}^{f}$ is extracted.

$$f_{vec} = f_{l,n} = W_{f_{l,n}}{}^{ws,ws} \rightarrow W_{f_{l,n}}^{ws^2,1} \tag{4}$$

being ws the size of the neighborhood. In this way, the input vector is increased by a factor of ws. Different values of ws should be tested in order to find the optimal value that maximizes the amount of input information without failing into overtraining problems [52].

Table 2 shows the different architectures based on dense neural networks evaluated during this research. With an input layer of size $m \times ws^2$ and an output layer of 1, each neural network is designed by modifying the depth and width of the network to evaluate the differences in both accuracy and

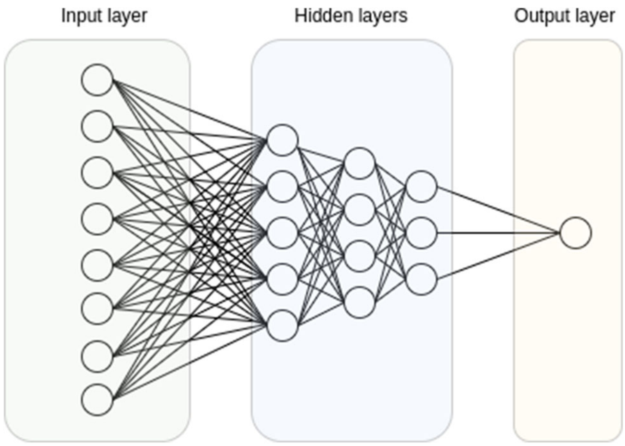
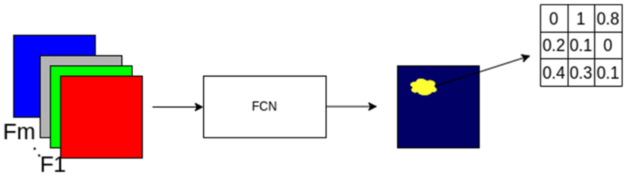


Fig. 4 "Fully connected" neural network architecture

Table 2 Dense model architectures. $m = 15$, $ws = 9$

Name	Input layer	Neurons per hidden layer	Output layer
DenseS	1215	256–64	1
DenseM	1215	512–128	1
DenseL	1215	1024–256-64	1



0	1	0.8
0.2	0.1	0
0.4	0.3	0.1

Fig. 5 Output layer format in FCN models

inference speed. The label for any given sample was set to be the true label of the central pixel of the patch.

4.2 Semantic image segmentation with fully convolutional networks

In the case of FCN models, the input images and corresponding ground truth were split into 256 × 256-pixel tiles to keep the memory consumption low during the training and validation. These tiles were adjacent and overlapped with a factor of 0.5.

As CNNs have the ability to learn features from input images, it may not be necessary to use all the manually extracted features, reducing complexity in the first layer and allowing the network to learn the optimal features to solve the given task. Moreover, the manual calculation features are computationally expensive and time-consuming. Thus, reducing the preprocessing steps will directly benefit the overall system performance.

To achieve maximum optimization of system calculations, grouping is performed based on the complexity of each calculation. As discussed in Sect. 3.2, level 1 features are more complex than level 0 features. Likewise, level 2 features are more complex than level 1 and level 0 features. For this reason, for each level, tests will be performed using all the features of its own level and below. Table 3 shows the final groups.

The final layer of the proposed FCN architecture is composed of a single channel of size 256×256, which outputs probability maps representing the likelihood of defects in each pixel. As shown in Fig. 5, the general FCN architecture includes this final output layer, which is essential for the pixel-wise defect detection task.

Although there are a large number of architectures available to choose from, during the research, we will focus on U-Net, VGG, and ResNet. The details of the implementation of this architecture are explained in the subsequent sections.

Table 3 Feature groups for model input layers

Name	Features
Group 1	Level 0
Group 2	Level 0 + Level 1
Group 3	Level 0 + Level 1 + Level 2

4.2.1 U-Net architectures

Unlike traditional architectures, U-Net [30] does not employ any fully connected layers. Instead, it only uses convolutional layers, with a ReLU activation function starting each normal convolution process. This design allows U-Net to effectively capture both fine-grained and high-level features in the input data, making it well suited for image segmentation tasks.

As shown in Fig. 6, U-Net addresses the bottleneck issue of the traditional autoencoder architecture by using skip connections between the encoder and decoder components. This allows U-Net to adapt to segmentation problems and segment objects of different sizes by preserving the fine-grained features of the original image.

4.2.2 VGG encoder

VGG19 [28] is a convolutional neural network (CNN) trained on the ImageNet dataset, known for its good performance and simple architecture with 19 layers. It has the potential for transfer learning and can reduce the risk of overfitting. The decoder is constructed using deconvolution blocks concatenated, and a pre-layer is added to convert the input tensor from n to 3 channels for the intended 3-channel input tensor. Figure 7 shows the final architecture.

4.2.3 ResNet encoder

The ResNet50 [29] architecture has several advantages for image segmentation using an FCN model, including its deep architecture, residual connections, and good performance on image classification tasks. The use of residual connections facilitates the flow of gradients through the network and improves the training of deep networks, enhancing the model's performance. Additionally, pre-trained weights and resources related to ResNet50 are widely available, making it a popular choice for researchers and practitioners. The decoder is constructed using several deconvolution blocks, and a *pre-layer must be added to convert the input tensor from n to 3 channels as the network is intended to have a 3-channel input tensor*. Figure 8 shows the final architecture.

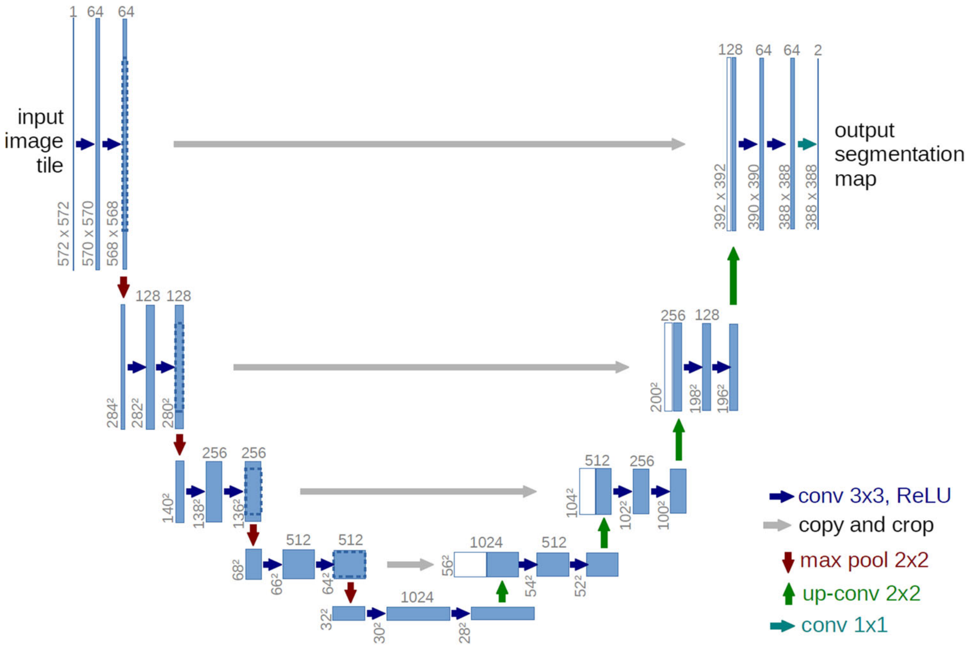


Fig. 6 U-Net architecture

5 Experimental setup

The training process takes place on an Ubuntu system with a NVIDIA GeForce GTX 1060 GPU. In this study, every model is trained 500 epochs with early stopping after epoch 70. The model batch size for the MLP models is set to 2048. On the other hand, the batch size for U-Net models is set to 8. Adam method [53] is used as the optimizer during the training stage with learning rate of 1×10^{-3}, $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

In the case of the MLP models, as the pixel prediction is evaluated independently, we use binary cross-entropy to compute the error between predictions and ground truth. On the other hand, a similarity metric is used for computing the error between the predicted image and ground truth image in FCN models. We found out that the best loss function for our dataset is a combination of Focal [54] ($\gamma = 1$) and Tversky loss ($\alpha = 0.3$,$\beta = 0.7$) [55]. Tversky allows us to set different weights for FP and TN, unlike Dice loss. Adding Focal loss helps to focus on hard cases with low probabilities. These hyperparameters were extracted empirically to achieve the best performance on the segmentation task.

$$\mathcal{L} = (1 - \mathcal{L}_{tversky})^{\gamma} \tag{5}$$

$$\mathcal{L}_{tversky} = \frac{TP}{TP + \alpha \times FP + \beta \times FN} \tag{6}$$

5.1 Dataset

A total of 1000 samples were extracted from different casting processes using the optical system described in Sect. 3.1. Each image has a resolution of 256 × 256. Data was manually labeled as an image segmentation problem, so images are labeled pixel-wise. The dataset was extracted from real casting lines from automotive factories over a period of 1 week. In order to simplify the experiment process and avoid the impact of unbalanced classes during our experiments, all defects are labeled as a generic defect, although the dataset contains a wide variety of defect types (buns, sands, cracks, etc.). A total of 63 samples of the same production process with enough defects are used to evaluate the accuracy of the models. To increase the number of available samples, rotation and transformation techniques based on the dihedral group D4 are applied [56].

5.2 Transfer learning

Using transfer learning with a smaller dataset can be a useful approach when resources such as data or computational power are limited. By training a fully convolutional network (FCN) model on a smaller dataset while using a pre-trained model as a starting point, you can potentially improve the model's performance and reduce the amount of training time and resources needed.

In this study, we trained both ResNet50 and VGG19 pre-trained with *imagenet* dataset on four different sizes of datasets (25%, 50%, 75%, and 100%) and compared the results to understand how the model's performance is affected by the size of the training dataset. This allowed us to identify the trade-off between performance and the amount of data used for training and potentially identify the optimal balance for our specific tasks and resources.

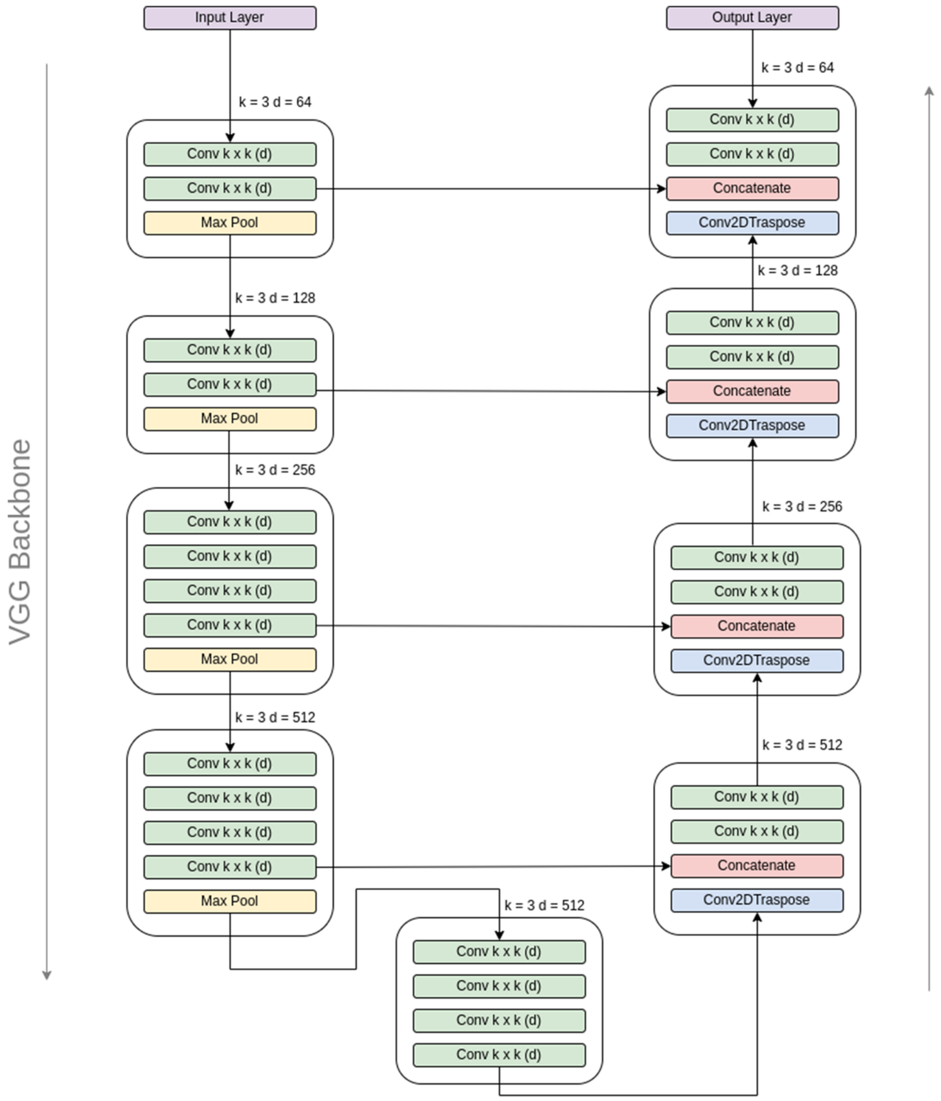


Fig. 7 VGG19-based encoder-decoder architecture

It is important to keep in mind that the performance of the model may also depend on the quality and diversity of the data, and not just the quantity. Using a well-curated and diverse dataset, even if it is small, may result in better performance compared to a larger but less diverse dataset. To obtain a diverse dataset, it is necessary to manually collect and curate the data, as automatic methods could not capture the full range of diversity needed for the model to generalize well. This can involve selecting images that cover a wide range of environment conditions, backgrounds, angles, and object sizes, among other factors.

5.3 Evaluation

One of the most common metrics to determine the accuracy of binary classification problems is the F1-score. As shown in Eq. 7, it is calculated as a combination of precision and recall and is equivalent to the Dice coefficient with two classes.

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{7}$$

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

The intersection over union (IoU), also known as the Jaccard Index, is one of the most commonly used metrics in semantic segmentation. The IoU is defined as the area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth and is calculated using Eq. 10.

$$IoU = \frac{TP}{TP + FN + FP} \tag{10}$$

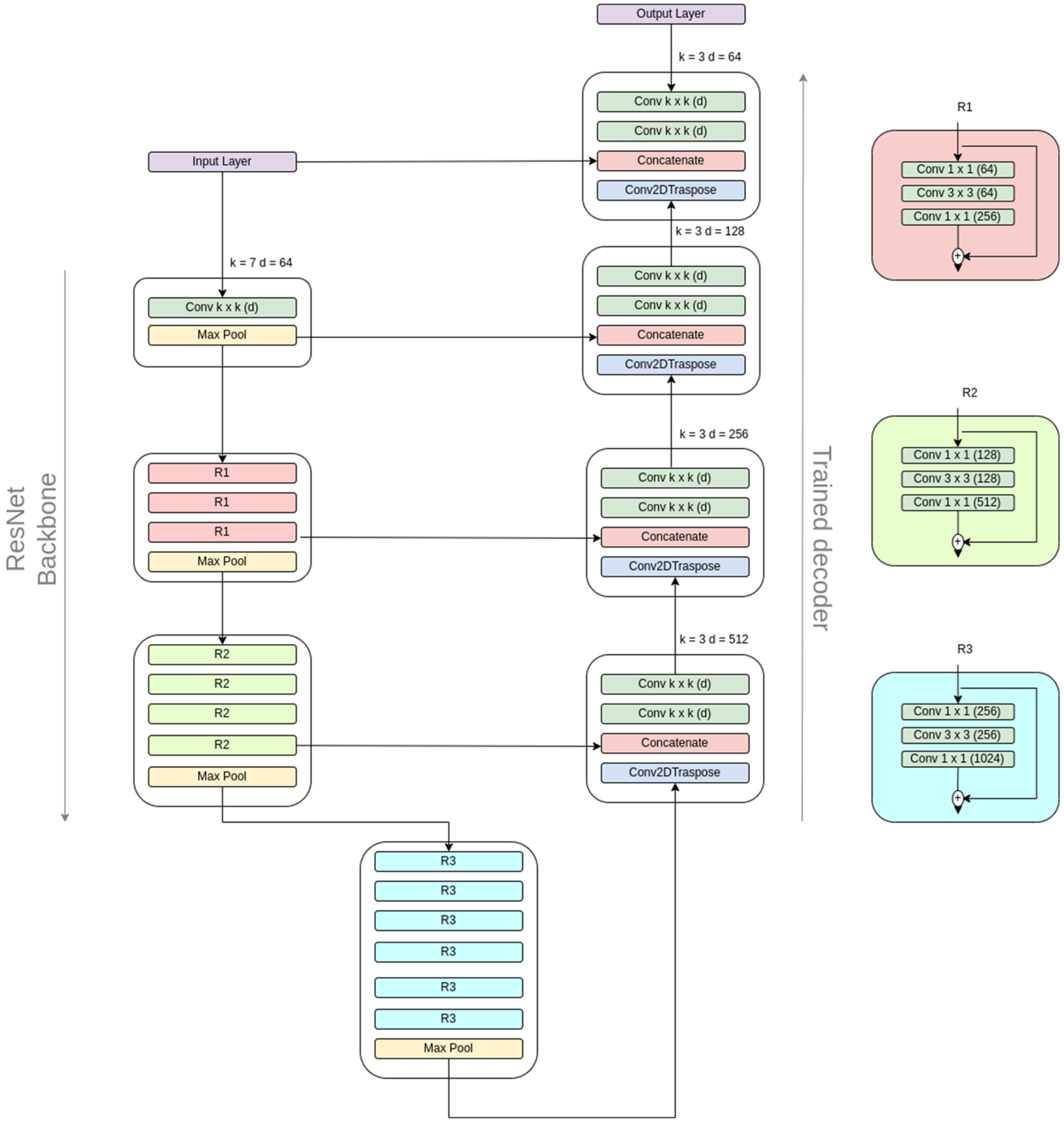


Fig. 8 ResNet-50-based encoder-decoder architecture

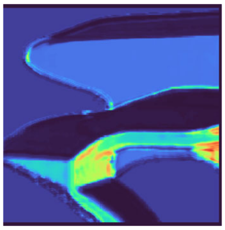
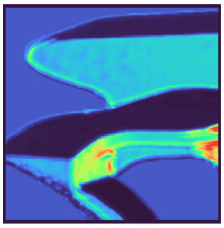
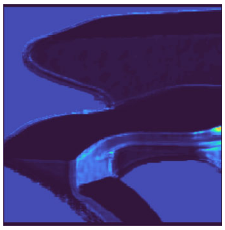
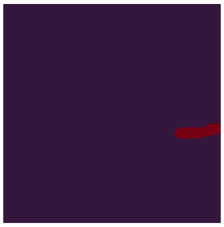
Although the model may still be trained using pixel-level labels, the performance will be evaluated using blob-level metrics. This approach can provide a more accurate evaluation of the model's performance by taking into account the connected regions of pixels, rather than just individual pixels, which can lead to a more accurate representation of the defective regions.

It is important to note that the manual labeling of images for training data can also have an impact on pixel-level metrics. In some cases, there may be normal pixels that are incorrectly classified as defective or defective pixels that are labeled as normal. This can affect the performance of the model when evaluated using metrics such as precision, recall, and F1-score, which are highly sensitive to bad labeled data. In such cases, metrics like intersection over union (IoU) can provide a better understanding of the model's overall performance by taking into account the overlap between the predicted and ground truth regions and providing a more holistic view of the model's ability to detect defects.

In order to compute the blob-level metrics, we will convert ground truth images to lists of bounding boxes considering the top left and the bottom right pixels as limits. Similarly, we create the list of predicted bounding boxes.

We will consider that the predicted bounding box is representing the real bounding box if $IOU > 0.5$. The following

Table 4 MLP qualitative results on a validation sample

DenseS	DenseM	DenseL	Ground truth
			

blob metrics will be calculated on this premise: precision, recall, F1-score.

In order to evaluate and compare the computational efficiency of different models, the time taken by each model to run the inference on a 256 × 256 window will be measured and averaged over the hole test dataset.

6 Results

In this section, we conduct qualitative and quantitative analyses to show the performance of the different methods proposed.

Qualitative results The segmentation masks generated by the models and the corresponding ground truth were visually examined for the detection results. As demonstrated in Table 4, models based on *MLPs* were able to separate the object from the background, but were unable to properly segment the defects. Conversely, Table 5 shows that the performance of *FCNs* was heavily influenced by the features employed. Models utilizing features from group 1 (G1) exhibited poor performance, while those incorporating features from groups 2 (G2) and 3 (G3) displayed higher accuracy in defect segmentation. However, it is worth noting that the G2 models generated more false positives compared to the G3 models, as shown in Table 9.

Quantitative results Complementing the qualitative results explained above, we provide a comparison of the proposed methods based on the metrics described in Sect. 5.3. As previously discussed, pixel-level results may be subject to

Table 5 FCN empirical results on a validation sample

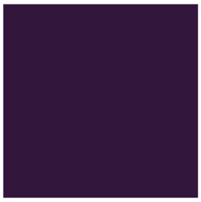
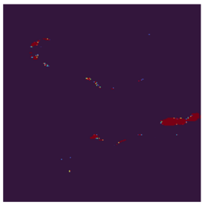
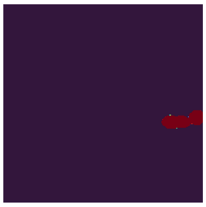
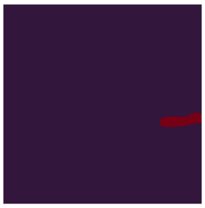
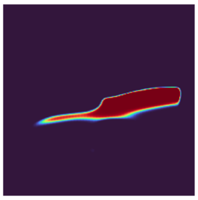
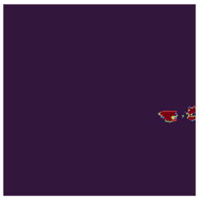
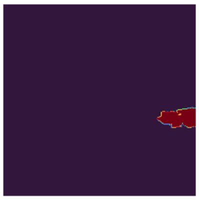
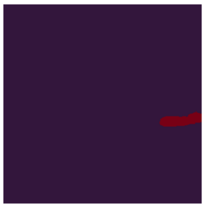
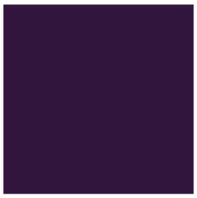
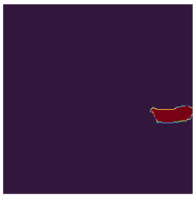
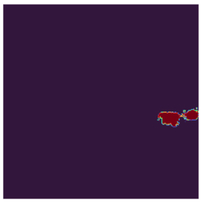
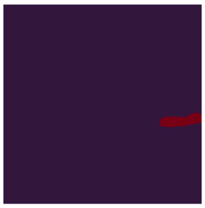
Architecture	G1	G2	G3	Ground truth
U-Net				
VGG19				
ResNet50				

Table 6 Pixel-wise dense model results. Values in bold indicate best model performaces

Name	IoU	Inference time (ms)
DenseS	0.035	**47.0**
DenseM	0 0.072	61.9
DenseL	**0.080**	90.6

misinterpretation due to labeling inaccuracies. As seen in Tables 6 and 7, the performance of *MLP* models was consistently poor, regardless of model size. Although all models displayed high recall values, the low precision scores suggest a high number of false positives. The metric results for the *FCN* models are presented in Tables 8 and 9, where the number of false positives was significantly reduced. The performance of these models was influenced by the features utilized, with better results achieved with the introduction of more features. There was no significant difference between the performance of models using feature groups 2 (G2) and 3 (G3). In terms of accuracy, the different model architectures showed similar results, though the U-Net model was noted for its computational efficiency.

Based on the findings presented above, we proposed evaluating transfer learning and investigating the impact of the number of training samples on both the *ResNet-50* and *VGG-19* architectures using the G2 features. In our study, we established the baseline by using the results obtained with 100% of the available data and no transfer learning.

Despite achieving slightly better results with the G3 features, we opted to use the G2 features in this industrial application due to their computational efficiency. The G2 features consist of a set of 7 features extracted from raw images, whereas the G3 features comprise 15 features, increasing the computational resources in the preprocessing stage. Given the real-time nature of the industrial setting, optimizing the computational cost is of utmost importance. In addition to utilizing G2 features, we also chose to freeze the pre-trained layers in both the *ResNet-50* and *VGG-19* architectures.

As illustrated in Fig. 9, our focus on utilizing the G2 features aligns with the need to strike a balance between accuracy and computational efficiency in this specific application.

The results show that the amount of data used for training has a significant impact on the performance of the model. As expected, the F1-score increases as more data is used for training. These results suggest that transfer learning can be an effective technique to train the models when the amount of available data is limited. Overall, the findings highlight the importance of carefully considering the amount of data available for training when developing machine learning models.

Table 7 Blob-wise dense model results. Values in bold indicates best model performaces

Name	Prec	Rec	F1
DenseS	**0.351**	0.997	**0.520**
DenseM	0.269	**1**	0.424
DenseL	0.345	0.917	0.503

Table 8 Pixel-wise FCN's results. Trained with group 1 (G1), group 2 (G2), and group 3 (G3) of features. Values in bold indicates best model performaces

Name	IoU	Inference time (ms)
ResNet50-G3	**0.989**	64.3
Unet-G3	0.987	**44.3**
VGG16-G3	**0.989**	70.4
ResNet50-G2	0.986	63.5
Unet-G2	0.977	**43.6**
VGG16-G2	**0.988**	69.6
ResNet50-G1	**0.959**	62.3
Unet-G1	0.75	**43.3**
VGG16-G1	0.665	66.5

The results show that the amount of data used for training has a significant impact on the performance of the model. As expected, the F1-score increases as more data is used for training. However, it is important to note that despite this quantitative improvement, qualitative analysis shows that the model's capability to detect defects is present even with fewer data (see Table 10). These results suggest that transfer learning can be an effective technique to train the models when the amount of available data is limited. Overall, the findings highlight the importance of carefully considering the amount of data available for training when developing machine learning models, pointing out that strategic data usage can yield significant benefits even before large datasets are accessible.

Table 9 Blob-wise FCN's results. Values in bold indicates best model performaces

Name	Prec	Rec	F1
ResNet50-g3	0.894	0.973	**0.932**
Unet-g3	0.878	0.963	0.918
VGG16-g3	0.910	0.954	0.932
ResNet50-g2	0.900	0.985	**0.941**
Unet-g2	0.677	931	0.784
VGG16-g2	0.874	0.945	0.908
ResNet50-g1	0.145	0.183	0.162
Unet-g1	0.367	0.237	0.288
VGG16-g1	0.476	0.471	**0.473**

Trained with group 1 (g1), group 2 (g2), and group 3 (g3) of features

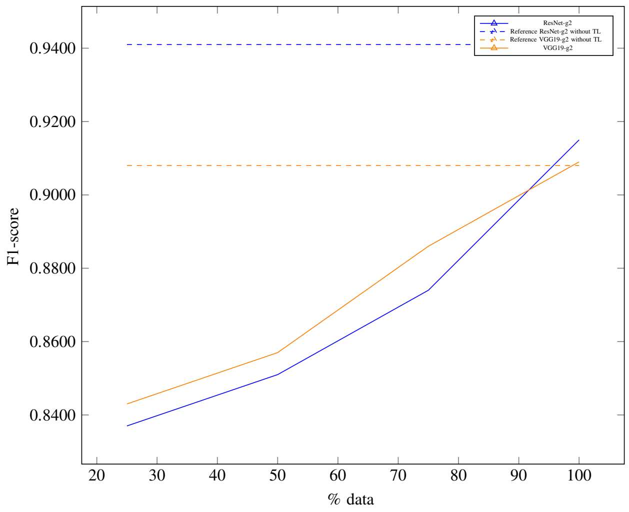


Fig. 9 FCN results applying transfer learning

7 Conclusions

Our study has made significant strides in demonstrating the applicability and effectiveness of machine learning, particularly deep learning techniques, in the realm of automated defect detection within the casting processes in the manufacturing sector. Focusing on the use of fully convolutional networks (FCNs) integrated with 3D imaging, this research represents a substantial advancement in the field of quality control, especially in the context of surface defect detection in metal parts.

Model architecture and feature extraction A key finding of our research is the pivotal role of model architecture in determining the system's performance. The use of FCNs, harnessing the power of convolutional neural networks (CNNs), has proven to be a game-changer in feature extraction from 3D images. This approach has outperformed traditional methods, which are often hampered by manual feature extraction and subjective human interpretation.

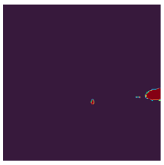
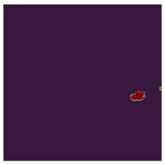
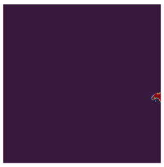
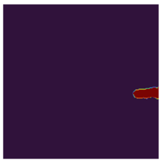
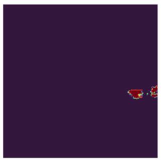
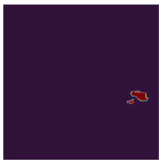
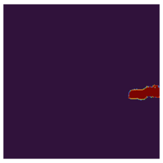
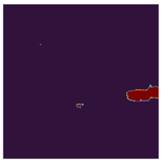
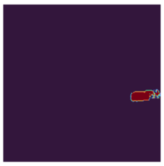
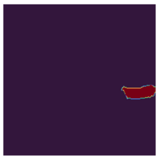
Data quality and quantity The study underscores the direct relationship between the volume and quality of training data and the accuracy of the defect detection models. The improvement in the F1-score with increased training data exemplifies the necessity of comprehensive datasets for effectively training machine learning models in precision-critical applications.

Inference time in real-time applications A cornerstone of our research has been the testing of these models in real-time applications. One of the key metrics, inference time, has been meticulously measured to ensure the practicality of these models in live manufacturing environments. Our findings indicate that the optimized FCN models not only maintain high accuracy but also achieve rapid inference times, making them viable for integration into production lines for immediate defect detection.

Impact on manufacturing processes The implementation of these deep learning techniques in real-time applications marks a significant evolution in manufacturing processes. The enhanced accuracy and efficiency in defect detection achieved by our methods can lead to substantial reductions in waste, improvements in product quality, and increased overall production efficiency. This is particularly crucial in sectors where surface defects can have serious repercussions on product functionality and safety.

Broader implications and challenges The implementation of advanced machine learning technologies in manufacturing raises several challenges, including data security, privacy, and the need for ongoing model maintenance and updates. Addressing these issues requires a collaborative effort between engineers, data scientists, and industry practitioners to ensure that these technologies are applied effectively and responsibly.

Table 10 Comparison of qualitative results of FCN-based approaches on % of data used for training

Architecture	25%	50%	75%	100%	Without TL
VGG19					
ResNet50					

Efficiency and reliability enhancements The real-time application of these models has not only validated their theoretical effectiveness but also demonstrated their potential to revolutionize industrial quality control. The balance achieved between accuracy and rapid inference times signifies a major leap forward in deploying smart, efficient, and reliable technologies in manufacturing processes.

In conclusion, our research not only validates the efficacy of machine learning techniques in enhancing defect detection but also highlights their practical applicability in industrial settings. The careful selection of features, optimization of model architecture, and consideration of training data volume have proven crucial in improving system performance. The successful integration of advanced deep learning models, particularly in the context of 3D imaging and real-time applications, represents a significant advancement in industrial manufacturing, paving the way for more intelligent, efficient, and reliable production processes.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by Daniel García Peña and Diego García Pérez. The first draft of the manuscript was written by Daniel García Peña, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Myklebust O, Eleftheriadis R, Capellan A (2014) Continuous improvement and benchmarking to achieve zero defect manufacturing. Adv Mater Res 1039:609–614. https://doi.org/10.4028/www.scientific.net/amr.1039.609
2. Caiazzo B, Di Nardo M, Murino T, Petrillo A, Piccirillo G, Santini S (2022) Towards zero defect manufacturing paradigm: a review of the state-of-the-art methods and open challenges. Comput Ind 134. https://doi.org/10.1016/j.compind.2021.103548
3. Bastas A (2020) Comparing the probing systems of coordinate measurement machine: scanning probe versus touch-trigger probe. Measurement 156. https://doi.org/10.1016/j.measurement.2020.107604
4. Meli F, Küng A (2007) AFM investigation on surface damage caused by mechanical probing with small ruby spheres. Meas Sci Technology 18(2):496. https://doi.org/10.1088/0957-0233/18/2/S24
5. Osumi A, Ito Y (2012) Basic study of non-contact measurement system for internal defect in solid materials using high-intensity aerial ultrasonic waves. In: 2012 Proceedings of SICE annual conference (SICE), pp 1120–1125
6. Ngo N-V, Hsu Q-C, Hsiao W-L, Yang C-J (2017) Development of a simple three-dimensional machine-vision measurement system for in-process mechanical parts. Adv Mech Eng 9(10):1687814017717183. https://doi.org/10.1177/1687814017717183
7. Takada K, Yokohama I, Chida K, Noda J (1987) New measurement system for fault location in optical waveguide devices based on an interferometric technique. Appl Opt 26(9):1603–1606. https://doi.org/10.1364/AO.26.001603
8. Blasco J, Aleixos N, Moltó E (2007) Computer vision detection of peel defects in citrus by means of a region oriented segmentation algorithm. J Food Eng 81(3):535–543. https://doi.org/10.1016/j.jfoodeng.2006.12.007
9. Sun T-H, Tseng C-C, Chen M-S (2010) Electric contacts inspection using machine vision. Image Vis Comput 28(6):890–901. https://doi.org/10.1016/j.imavis.2009.11.006
10. Malamas EN, Petrakis EG, Zervakis M, Petit L, Legat J-D (2003) A survey on industrial vision systems, applications and tools. Image Vis Comput 21(2):171–188
11. Cootes TF, Page G, Jackson C, Taylor CJ (1996) Statistical grey-level models for object location and identification. Image Vis Comput 14(8):533–540
12. Bezdek JC, Keller J, Krisnapuram R, Pal N (1999) Fuzzy models and algorithms for pattern recognition and image processing vol 4. Springer
13. Chang J, Han G, Valverde JM, Griswold NC, Duque-Carrillo J-F, Sanchez-Sinencio E (1997) Cork quality classification system using a unified image processing and fuzzy-neural network methodology. IEEE Trans Neural Networks 8(4):964–974
14. Tsai D-M, Chen J-J, Chen J-F (1998) A vision system for surface roughness assessment using neural networks. Int J Adv Manuf Technol 14:412–422
15. Drake P, Packianather M (1998) A decision tree of neural networks for classifying images of wood veneer. Int J Adv Manuf Technol 14:280–285
16. Kovac I (2004) Flexible inspection systems in the body-in-white manufacturing. In: International workshop on robot sensing, 2004. ROSE 2004, pp 41–48. https://doi.org/10.1109/ROSE.2004.1317612
17. Czimmermann T, Ciuti G, Milazzo M, Chiurazzi M, Roccella S, Oddo CM, Dario P (2020) Visual-based defect detection and classification approaches for industrial applications—a survey. Sensors 20(5). https://doi.org/10.3390/s20051459
18. Molleda J, Usamentiaga R, García DF, Bulnes FG, Espina A, Dieye B, Smith LN (2013) An improved 3D imaging system for dimensional quality inspection of rolled products in the metal industry. Comput Ind 64(9):1186–1200. https://doi.org/10.1016/j.compind.2013.05.002 . Special Issue: 3D Imaging in Industry
19. Li R, Jin M, Paquit VC (2021) Geometrical defect detection for additive manufacturing with machine learning models. Materials & Design 206:109726. https://doi.org/10.1016/j.matdes.2021.109726
20. Gojcic Z, Zhou C, Wegner JD, Wieser A (2019) The perfect match: 3d point cloud matching with smoothed densities. In: Proceedings

of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)

21. Hegedűs-Kuti J, Szőlősi J, Varga D, Abonyi J, Andó M, Ruppert T (2023) 3D scanner-based identification of welding defects-clustering the results of point cloud alignment. Sensors (Basel, Switzerland) 23(5):2503. https://doi.org/10.3390/s23052503
22. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. nature 521(7553):436–444
23. Wang X-Y, Bu J (2010) A fast and robust image segmentation using FCM with spatial information. Digital Signal Process 20(4):1173–1182. https://doi.org/10.1016/j.dsp.2009.11.007
24. Zhao Z-Q, Zheng P, Xu S-T, Wu X (2019) Object detection with deep learning: a review. IEEE Trans Neural Netw Learn Syst 30(11):3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865
25. Wang W, Liang D, Chen Q, Iwamoto Y, Han X-H, Zhang Q, Hu H, Lin L, Chen Y-W (2020) Medical image classification using deep learning. In: Deep learning in healthcare, pp 33–51. Springer
26. Learned-Miller EG (2014) Introduction to supervised learning. I: Department of Computer Science, University of Massachusetts 3
27. Menghani G (2023) Efficient deep learning: a survey on making deep learning models smaller, faster, and better. ACM Comput Surv 55(12):1–37
28. Very deep convolutional networks for large-scale image recognition (2014)
29. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
30. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18, pp 234–241. Springer
31. Long J, Shelhamer E, Darrell T (2014) Fully convolutional networks for semantic segmentation. arXiv. https://doi.org/10.48550/ARXIV.1411.4038
32. Weiss K, Khoshgoftaar TM, Wang D (2016) A survey of transfer learning. Journal of Big data 3(1):1–40
33. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
34. Chou J-S, Telaga AS (2014) Real-time detection of anomalous power consumption. Renew Sustain Energy Rev 33:400–411. https://doi.org/10.1016/j.rser.2014.01.088
35. Li Z, Li J, Wang Y, Wang K (2019) A deep learning approach for anomaly detection based on SAE and LSTM in mechanical equipment. Int J Adv Manuf Technol 103. https://doi.org/10.1007/s00170-019-03557-w
36. Azimi M, Pekcan G (2019) Structural health monitoring using extremely-compressed data through deep learning. Computer-Aided Civil and Infrastructure Engineering. https://doi.org/10.1111/mice.12517
37. Baygin M, Karakose M, Sarimaden A, Akin E (2017) Machine vision based defect detection approach using image processing. In: 2017 International artificial intelligence and data processing symposium (IDAP), pp 1–5. https://doi.org/10.1109/IDAP.2017.8090292
38. Pastor-López I, Sanz B, Tellaeche A, Psaila G, de la Puerta JG, Bringas PG (2021) Quality assessment methodology based on machine learning with small datasets: industrial castings defects. Neurocomputing 456:622–628. https://doi.org/10.1016/j.neucom.2020.08.094
39. Jia H, Murphey YL, Shi J, Chang T-S (2004) An intelligent real-time vision system for surface defect detection. In: Proceedings of the 17th international conference on pattern recognition, 2004. ICPR 2004, vol3, pp 239–2423. https://doi.org/10.1109/ICPR.2004.1334512
40. Chen X, Chen J, Han X, Zhao C, Zhang D, Zhu K, Su Y (2020) A light-weighted CNN model for wafer structural defect detection. IEEE Access 8:24006–24018. https://doi.org/10.1109/ACCESS.2020.2970461
41. Xu Y, Li D, Xie Q, Wu Q, Wang J (2021) Automatic defect detection and segmentation of tunnel surface using modified mask R-CNN. Measurement 178. https://doi.org/10.1016/j.measurement.2021.109316
42. Zhang M, Wu J, Lin H, Yuan P, Song Y (2017) The application of one-class classifier based on CNN in image defect detection. Procedia Computer Science 114:341–348. https://doi.org/10.1016/j.procs.2017.09.040. Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30—November 1, 2017, Chicago, Illinois, USA
43. Jiang L, Wang Y, Tang Z, Miao Y, Chen S (2021) Casting defect detection in X-ray images using convolutional neural networks and attention-guided data augmentation. Measurement 170. https://doi.org/10.1016/j.measurement.2020.108736
44. Ferguson MK al (2018) Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning. Smart Sustain Manuf Syst 2. https://doi.org/10.1520/SSMS20180033
45. Ravanbakhsh S, Schneider J, Poczos B (2017) Equivariance through parameter-sharing. In: International conference on machine learning, PMLR, pp 2892–2901
46. Chen W, Zou B, Zheng Q, Huang C, Li L, Liu J (2023) Research on anti-interference detection of 3D-printed ceramics surface defects based on deep learning. Ceram Int. https://doi.org/10.1016/j.ceramint.2023.04.081
47. Dimitriou N, Leontaris L, Vafeiadis T, Ioannidis D, Wotherspoon T, Tinker G, Tzovaras D (2020) A deep learning framework for simulation and defect prediction applied in microelectronics. Simul Model Pract Theory 100:102063. https://doi.org/10.1016/j.simpat.2019.102063
48. Sousa GB, Olabi A, Palos J, Gibaru O (2017) 3D metrology using a collaborative robot with a laser triangulation sensor. Procedia Manuf 11:132–140
49. Blomley R, Weinmann M, Leitloff J, Jutzi B (2014) Shape distribution features for point cloud analysis-a geometric histogram approach on multiple scales. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 2(3):9
50. Peña DG (2021) Diseño e implementación de técnicas de machine learning para la detección de defectos superficiales en piezas sometidas a procesos de estampado o fundición. Universidad de Oviedo
51. Bianco JPM (2022) Mejora de la eficiencia computacional de técnicas machine learning para la etección temprana de defectos superficiales en piezas sometidas a procesos de fundición y estampado de chapa. Universidad de Oviedo
52. Dietterich T (1995) Overfitting and undercomputing in machine learning. ACM Comput Surv 27(3):326–327. https://doi.org/10.1145/212094.212114
53. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv. https://doi.org/10.48550/ARXIV.1412.6980
54. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. arXiv. https://doi.org/10.48550/ARXIV.1708.02002
55. Salehi SSM, Erdogmus D, Gholipour A (2017) Tversky loss function for image segmentation using 3D fully convolutional deep networks. arXiv. https://doi.org/10.48550/ARXIV.1706.05721
56. Dieleman S, Fauw JD, Kavukcuoglu K (2016) Exploiting cyclic symmetry in convolutional neural networks. In: Balcan MF, Weinberger KQ (eds) Proceedings of The 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research,

vol 48, pp 1889–1898. PMLR, New York, New York, USA. https://proceedings.mlr.press/v48/dieleman16.html

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.