JJMIE

Volume 17, Number 4, December. 2023
ISSN 1995-6665
Pages 541– 548
https://doi.org/10.59038/jjmie/170409
Editor-in-Chief: Prof. Ali M. Jawarneh

Jordan Journal of Mechanical and Industrial Engineering

Parametric Study of Inspecting Surface Defects in Investment Casting

Nabhan Yousef [*,1], Amit Sata[2]

[1]Research Scholar, Marwadi University, Rajkot (India),
[2]Professor, Department of Mechanical Engineering, Marwadi University, Rajkot (India)

Received 7 Jun 2023 Accepted 29 Sep 2023

Abstract

Metal defects detection has always been an essential task for the majority of various industries, moreover, it is the core element in the metal inspection too. This research paper explores the effectiveness of different deep learning algorithms for surface-defect detection in investment casting using the Inspection 4.0 approach. The study compared the performance of four popular deep learning algorithms, Fast R-CNN, Faster R-CNN, ResNet, and YOLO, using the accuracy metric as a performance evaluation measure. The results show that ResNet achieved the highest accuracy rate of 95.89%, followed by Faster R-CNN with 90.23%, Fast R-CNN with 89.21%, and YOLO with 86.43%. The findings of this research demonstrate that ResNet and Faster R-CNN are effective deep-learning algorithms for automated surface-defect detection in investment casting. On the other hand, Fast R-CNN and YOLO exhibited lower accuracy rates.

The outcomes of this study provide valuable insights into the effectiveness of deep learning algorithms for surface-defect detection in investment casting. The high accuracy rate achieved by ResNet and Faster R-CNN can guide the development of automated inspection systems for investment casting in various industries such as aerospace, automotive, and medical.

Keywords: Defects detection, CNN, YOLO, Fast R-CNN, Faster R-CNN, ResNet.

1. Introduction

Investment casting is a manufacturing process widely used in various industries such as aerospace, automotive, and medical for producing complex metal components with high precision. Surface defects are common in investment casting and can lead to component failure or reduced lifespan. Therefore, surface inspection is crucial in the production process to ensure quality control. Artificial Intelligence AI has shown high impact on the accuracy of detecting defects and made measurements more efficient and reliable [1]. Other AI applications of AI in manufacturing includes development of lean manufacturing of garment [2], machining parameter optimization [3],[4], calculating the overall equipment maintenance in steel products [5], [6] and shape modification of electric vehicles [7] and strengthening metallic composite alloys [8].

With the development of Industry 4.0, the inspection process can be automated using advanced technologies such as deep learning. Deep learning is a subfield of machine learning that uses artificial neural networks to learn from large datasets. It has demonstrated promising results in various applications, including image recognition and object detection. In recent years, Convolutional Neural Networks (CNNs) have emerged as a promising solution for defect inspection in various industries, including investment casting[9]. CNNs are effective in detecting and classifying surface defects in investment castings. These networks can accurately identify defects, such as cracks, blisters, and pinholes.

Furthermore, they can also distinguish between different types of material, allowing them to recognize defects in complex geometries. By utilizing CNNs, manufacturers can quickly and accurately detect surface defects and ensure the highest possible quality of their products. They have been trained on large datasets of investment casting images, allowing them to learn the relevant features for defect detection automatically. This has led to high accuracy rates and the ability to detect even subtle defects[10], [11]. Many studies have used pre-trained CNN models such as VGG or ResNet and fine-tuned them on investment casting datasets to improve accuracy. This has proven to be a cost-effective approach, resulting in good performance compared to training from scratch. Some studies have used a combination of CNNs and traditional image processing techniques for improved results, followed by morphological operations for refinement, such image processing techniques are combined with neural networks[12], with YOLO algorithm[13], and the engagement with the convolutional autoencoders CAE to improve the learning quality and mitigate the need of larger subsets of data[14], [15].Other YOLO-based models have been developed for industrial defect detection in general. *Li J et al*.[13] have used an improved YOLO for steel strip surface detection, and

* Corresponding author e-mail: nabhan.yousef110492@marwadiuniversity.ac.in.

they have achieved a 99% detection rate; their network can predict the location and scale information of the entire production line. There were also many versions of YOLO as optimized extensions of this algorithm, such as YOLO-v2[16], YOLO-v3[17], YOLO-v4[18], and YOLO-v5[19]. Some limitations of YOLO are that it cannot detect too minor defects and makes more localization errors compared to Faster R-CNN. Also,

While CNNs have successfully detected surface defects, the need for large amounts of annotated data and computational resources can limit their practicality in industrial settings. Additionally, the ability of CNNs to generalize to new types of defects can be limited. There is a need for further research in this area to address the limitations of current methods and improve the generalizability of CNNs for investment casting defect detection. This could involve the development of more efficient and effective architectures or using unsupervised or semi-supervised learning methods.

This research paper investigates the effectiveness of different deep learning algorithms in surface-defect detection for investment casting using the Inspection 4.0 approach. We evaluate the performance of four popular deep learning algorithms, namely Fast R-CNN, Faster R-CNN, ResNet, and YOLO, in terms of accuracy.

The findings of this study can guide the development of automated inspection systems for investment casting, which can significantly reduce inspection time and improve production efficiency. The successful implementation of an automated inspection system can also improve product quality and safety, reduce costs, and increase customer satisfaction.

2. Review of literature

In 2015, Joseph Redmon et al.[20] proposed the You Only Look Once (YOLO) algorithm, which divides an input image into a grid of S×S cells. Each cell predicts the detection of an object's bounding box and its corresponding score. Moreover, detection should include class classification. Therefore, YOLO also predicts the conditional class probability of the detected object within a cell. The convolutional layers at this network are responsible for defects feature extraction. For each cell in the grid, YOLO determines if it contains a defect or not, considering the defect class too according to the defect's features. Fast R-CNN was introduced right after R-CNN[21] for object detection. Moreover, it is considered an improved R-CNN architecture with some modifications. Fast R-CNN has added an ROI Pooling layer to extract feature vectors from all proposals (ROI).

Due to using multiple stages, R-CNN was slower because training each stage takes its place and requires more time, but Fast R-CNN has only a single stage, which means faster training. On the other hand, metal defect detection tasks have primarily utilized the Faster R-CNN model, which has undergone various optimizations through modifications. As an example, *Zhao et al*.[22], in order to enhance the modelling capabilities of the network for steel defect detection, deformable convolutions were introduced by *Zhao et al. Ren et al*. [23]Ren et al. have suggested a somewhat smaller architecture of Faster R-CNN for real-time detection of steel strip defects, which has demonstrated superior performance compared to traditional image processing techniques and other deep learning-based methods.[15]. The architecture proposed by Ren et al. consists of three main components: a feature extraction network, a regional proposal network (RPN), and a regional CNN. The feature extraction network is responsible for extracting the features from the defective images, which are then utilized by both the RPN and R-CNN to identify the regions of interest (ROI). The ROIs are further processed through pooling and fully connected layers. A non-maximum suppression algorithm is subsequently employed to eliminate the frames with lower scores and retain those with higher scores, similar to the approach used in the YOLO algorithm. Please refer to Figure 5 for a visual representation of this architecture. Goyal A et al [24] have used AI for measuring temperature caused by metal cutting.

One of the commonly sued model for defects detection is ResNetwhich is a sophisticated neural network architecture with exceptional performance in various computer vision applications, such as image classification, object detection, and semantic segmentation. ResNet was introduced in 2016 by *Kaiming He et al*.[25]. One of the significant innovations of ResNet is the utilization of residual connections, which enables the network to learn residual mappings instead of direct mappings. This technique helps to overcome the vanishing gradient problem that often arises in intense neural networks and allows for training much deeper networks. The ResNet architecture consists of multiple blocks that comprise convolutional layers, batch normalization, and ReLU activation functions. The residual connections are added between the blocks, enabling the input to be directly transmitted to the output of the block, bypassing the convolutional layers.[25], [26].

3. Materials and Methods

The methodology adopted for the implementation of deep learning models and Evaluation in the detection of casting defects is shown in Figure .1

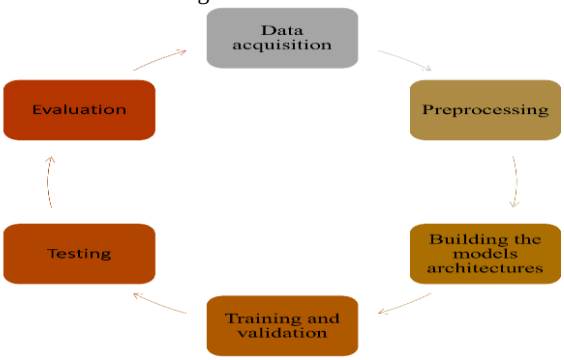

Figure 1. Methodology adopted for evaluation of different CNN architectures for inspection of surface defects in investment casting

3.1. Data Acquisition

Gathering data is a crucial step that impacts the accuracy of a trained model. In the case of deep learning, having a vast collection of images for training will result in better identification of objects, but there are other factors to consider. To achieve this, high-resolution images of investment castings were obtained from a foundry in Rajkot, India. The dataset consists of 12,000 standard casting images and 5,000 defective casting images, each with a 3472 x 3472 pixels resolution. A dataset comprising 3,600 defective casting images and 3,600 non-defective casting images was created to ensure balanced results and strike balance between dataset size, class distribution, and adherence to establish deep learning practices. Though this approach ensures the fairness and robustness for the performance of casting defects. The dataset was then split into the train, validation, and test sets, consisting of 2,880, 540, and 180 images. Figure 2 displays some examples of the dataset.

Figure 2. Some samples of the dataset

3.2. Preprocessing

The enhancement of the investment casting images involved an intricate preprocessing phase that significantly elevated their overall quality. In pursuit of this objective, a comprehensive array of techniques was meticulously applied, rendering each image uniform in terms of background, color, size, orientation, and resolution. Moreover, Gaussian filtering was employed to reduce the noise and effectively enhance the visual clarity. Simultaneously, any images which were highly distorted are removed from the dataset. Moreover, to ensure a consistent appearance, a meticulous cropping process was done, to maintain a uniform background across all images. Furthermore, the images were resized, an astute decision aimed at reducing the computational demands while improving resolution standards, the uniform size of 520x520 pixels for each image was considered.

Eventually, augmentation techniques were also implemented to enhance the model's generalizability and overall performance. These augmentation techniques comprise rotation with a 0.3 rate, flipping with a 0.2 rate, and scaling with a 0.2 rate. This judicious use of augmentation not only bolstered generalizability but also ensured a consistent orientation across the entire dataset, thus augmenting the overall quality and effectiveness of the investment casting image dataset.

3.3. The proposed models

3.3.1. YOLO

The bounding box method is used in this algorithm; in other words, the YOLO algorithm detects objects within an image and surrounds the defect with a bounding box B, and a confidence value corresponds to each bounding box. The confidence value of the bounding box is given by:

$$confidence = P_r(Defect) \times IoU_{B-box}^{GT} \tag{1}$$

Where P_r(defect) is the probability of a defect being found in the grid, IoU: is the intersection over union metric; the algorithm calculates the intersection over union (IoU) between the ground truth (GT) and the bounding box (B-box) to determine their overlapping rate. To eliminate redundant boxes with lower confidence values for a specific object, the Non-Maximum Suppression (NMS) method is employed. For each bounding box, the algorithm generates a final output that includes the centre coordinates of the defect (x, y) as well as the height (h) and width (w) of the bounding box. Additionally, the confidence value of the bounding box, as described earlier, is also included in the output. Finally, the probability of a defect belonging to a particular class is calculated to provide a class-specific confidence score for each box. Class confidence corresponding to a bounding box is encoded into a tensor of size [S×S×B × (x, y,h, w, c)], where B is the bounding boxes number, S×S is the grid cells dimensions, c is the confidence value of a bounding box correlated to the detected defect. Moreover, for the class confidence score, it is given by:

$$P_r(Class_i|Defect) \times P_r(Defect) \times IoU_{B-box}^{GT} = P_r(Class_i) \times IoU_{B-box}^{GT} \tag{2}$$

3.3.1.1. YOLO Loss Function:

The loss function for YOLO is a composite of two components: the bounding box localization loss and the classification loss for multi-class detection. Here, both losses use the sum of squares errors. In order to increase the loss of bounding box coordinates prediction (λ_{coord}) and decrease the confidence score loss of bounding boxes without an object being detected ($\lambda_{background}$). In the original paper, down-weighting the loss is contributed by bounding boxes with objects and bounding boxes of background, where $\lambda_{coord}$was set to 5, and $\lambda_{background}$was set to 0.5. The loss function is given by:

$$\mathcal{L}_{loc} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{defect} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \tag{3}$$

$$\mathcal{L}_{cls} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} (\mathbb{I}_{ij}^{defect} + \lambda_{background}(1 - \mathbb{I}_{ij}^{defect}))(C_{ij} - \hat{C}_{ij})^2 + \sum_{i=0}^{S^2} \sum_{c \in C} \mathbb{I}_i^{defect} (p_i(c) - \hat{p}_i(c))^2 \tag{4}$$

$$\mathcal{L} = \mathcal{L}_{loc} + \mathcal{L}_{cls} \tag{5}$$

Where $\mathbb{I}_i^{defect}$refers to the presence of the object at cell (i) and $\mathbb{I}_{ij}^{defect}$ refers to the prediction of the jth bounding box at cell (i). The general format of yolo is shown in Fig.3.

3.3.2. Fast R-CNN

Usually, Fast RCNN consists of CNN networks trained on a larger dataset (e.g., Image Net), but with replacing the last pooling layer with an ROI Pooling layer. Moreover, replacing the Fully Connected layer with two branches, one is for classification with a SoftMax output per the number of defects classes, and the second is a bounding-box regressor, as shown in Fig.4.

In Fast RCNN, the SVM classifier is replaced by a SoftMax layer, which extends the neural network predictions and classification rather than creating another model for this task. This gives the network more consistency.

3.3.3. Faster R-CNN Model:

3.3.3.1. Faster R-CNN architecture:

Modifying the RPN module to be a fully convolutional neural network which produces proposals (bounding boxes) with multi-scales and aspect ratios. RPNs could be considered as attention. It has the same concept because it tells the network where to look rather than focusing on non-important regions of the image; this can lead to better performance because it reduces the computation. In other words, Faster R-CNN is combined with Fast R-CNN mentioned in the previous section, and the RPN module consists of convolutional layers, where both parts share the same CNN layers, making the training done only once.

The RPN (Regional Proposal Network) operates on the feature map extracted from the last shared CNN layer with the Fast R-CNN model. It uses a sliding window approach with a specific size (n x n) to traverse the feature map. As a result, for each window, several region proposals are generated. These proposals are then filtered using an Objectness score based on the IOU (Intersection over Union) score, similar to R-CNN's filtering process.

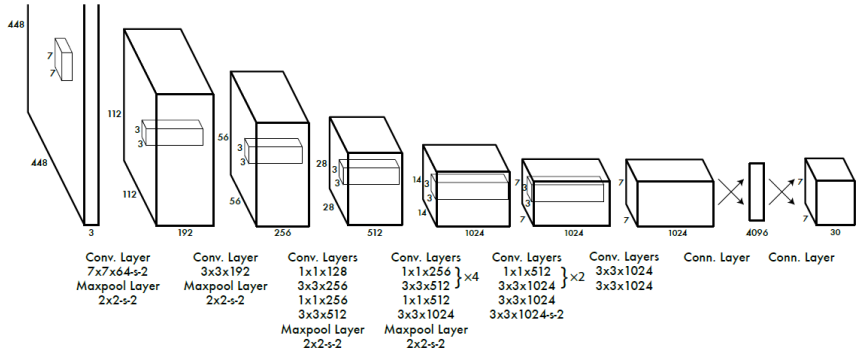

Figure 3. YOLO architecture[27]

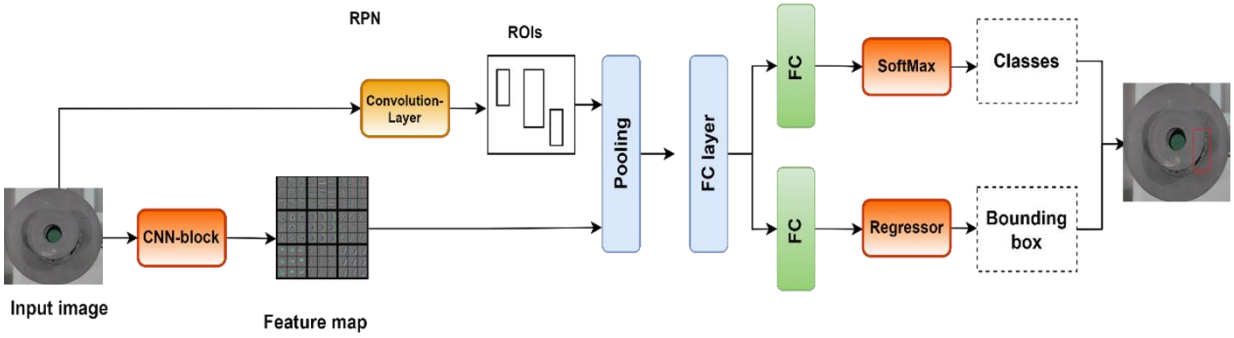

Figure 4. Fast R-CNN architecture

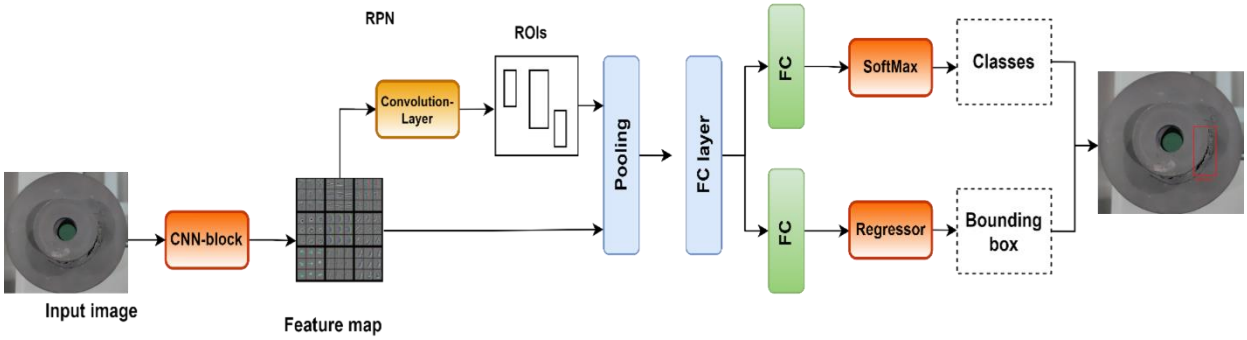

Figure 5. Faster R-CNN architecture

3.3.3.2. Deformable convolutions

The ROIs are divided into k×k blocks by pooling layer, where each proposal (bounding box) has a size of w×h, then the output of a regular RPN is given by:

$$y(i,j) = \sum_{P \in bin(i,j)} \frac{x(P_0 + P)}{n_{i,j}} \quad (6)$$

Equation (6) shows the formula for the output of the characteristic graph after pooling. It calculates the sum of the values of pixels (x) in a specific region of interest (ROI) that is defined by the upper left corner pixel (p0) and the current pixel (p) at any position in a given coordinate ($b_{in}(i,j)$). The pixel's value ($n_{i,j}$) is a normalization factor. The deformable convolution technique adjusts the pin locations to generate feature maps and applies offsets of size (2×k×k). This enhanced offset region is then pooled to produce the feature map, and the deformable ROI formula is applied:

$$y(i,j) = \sum_{P \in bin(i,j)} \frac{x(P_0 + P + \Delta p_{i,j})}{n_{i,j}} \quad (7)$$

Where $\Delta p_{i,j}$ is offset at each location (i≥0, k>j). Deformable convolution compared to the regular convolution is shown in Fig.6.

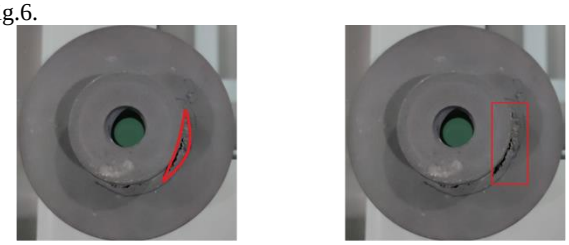

Figure 6. Normal and deformable convolutions

Soft Non-Maximum Suppression is used the same as R-CNN, which reduces the false detection rate in the final detected defects by eliminating the duplicate frames.

3.3.3.3. RPN loss function

The loss function for RPN is the sum of the classification loss and the anchor regression coefficients loss, and this formula gives it:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (8)$$

Where "i" is the anchor index. $N_{cls}$is the class number, and N_{reg} is the regression coefficients number. L_{cls} is the binary classification loss of two classes (foreground, background). "p_i" is the output score for classification of the i^{th} anchor, and p_i^* is the ground truth label (0 or 1). L_{eg} (t_i, t_i^*) is the regression loss; it is active only when the anchor contains a defect (p_i^* is 1), and it is the prediction output of the regression layer.

The regression coefficients are calculated as follows:

$$t_x^* = \frac{x^* - x_a}{w_a}$$
$$t_x^* = \frac{y^* - y_a}{h_a}$$
$$t_w^* = \log(\frac{w^*}{w_a}) \quad (9)$$
$$t_h^* = \log(\frac{h^*}{h_a})$$

Where (x,y) is the anchor centre coordinates, and (w,h) is the width and height of the anchor. K_a, k^* stands for anchor parameters and ground truth anchor parameters, respectively, where k ϵ (x, y, h, w). Regression loss of the i^{th} anchor is applied to the corresponding i^{th}regressor if it has a positive abjectness score.The concept of anchor boxes has been introduced to detect objects at different scales and aspect ratios. Rather than relying on a pyramid of images or filters with varying scales and aspect ratios, anchor boxes are fixed with predetermined scales and aspect ratios. Regions mapped to these anchor boxes can detect objects at various scales and aspect ratios. Ratios. The shared convolutional computation within the RPN and R-CNN modules helps reduce the computations.

Faster R-CNN deploys nine anchors (3 different scales at e different aspect ratios) for each location in the feature map, and it has 2×9 abjectness scores with 4×9 coordinates, Fig.7.

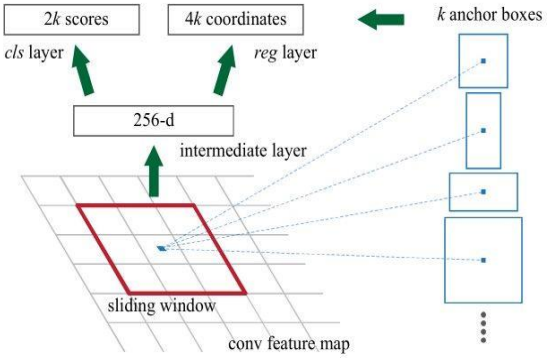

Figure 7. RPN with sliding window[28]

3.3.3.4. Objectness Score

It is a concept consisting of an IOU score, where while training the RPN, each anchor is assigned a positive or negative IOU score. For example, if An IOU score between an anchor and ground truth is more significant than 0.7, then the abjectness score is positive, and the anchor is classified as foreground (object). Moreover, for an IOU<0.3, the abjectness score is negative, and the anchor is classified as background.

3.3.3.5. Faster R-CNN training

Regression coefficients with RPN loss are applied for better localization accuracy of defects. All anchors are organized accordingly with their class sores; after that, NMS is applied to eliminate the anchors that do not satisfy the thresholding criteria mentioned earlier. The boundary anchors produced by the RPN are kept within the boundaries of the image. The RPN generates proposals that are then used to train the Fast R-CNN for object detection. All generated proposals are used during the training phase, but only the top N proposals are used during the testing phase.

3.3.4. ResNet

For the ResNet model, the categorical cross-entropy loss is utilized to assess the disparity between the predicted probability distribution and the actual probability distribution of the labels. Its purpose is to gauge the degree of dissimilarity between these two distributions. It is defined as:

$$L = -1/N * sum\ (y * log(y\_hat) + (1 - y) * log\ (1 - y\_hat)) \quad (10)$$

Where N is the number of samples, y is a one-hot encoded vector representing the actual labels, y_hat is a vector of predicted probabilities, and the log is the natural logarithm. The categorical cross-entropy loss encourages the network to

assign a high probability to the correct class and a low probability to the incorrect classes. The network adjusts its weights during training to minimize this loss function, improving classification performance.

4. Result and Discussion

All the previous models, including YOLO, Fast R-CNN, Faster R-CNN, and ResNet, were trained based on the 85% preprocessed images and validated on 10 %.Where the dataset splitting for training, validation, and testing was performed using random selection to ensure data representativeness and minimize potential biases. The hyperparameters of the training process were adjusted after deploying three ways of automated methods to find the optimal hyperparameters, including (Grid Search, Random search, and Bayesian optimization) as shown in Table.1, Table. 2, Table.3 and Table. 4.

The best training architectures for each model were taken and tested based on the test dataset (5% of the dataset), the test dataset was randomly sampled 5% of the entire dataset while ensuring that the proportion of defective and non-defective casting images in the testing set reflected on the entire dataset. This approach helps ensure that the testing set is representative of the overall dataset and is not biased towards any specific class.

and the result is shown in Fig.8.

In this study, we evaluated the performance of four deep learning algorithms (ResNet, Fast R-CNN, Faster R-CNN, and YOLO) for the task of detecting surface defects in investment casting. The accuracy results of these algorithms are as follows: ResNet achieved the highest accuracy at 95.89%, followed by Faster R-CNN with 90.23%, Fast R-CNN with 89.21%, and YOLO with 86.43%.

To provide a more comprehensive discussion, we stated the details of each algorithm highlights and limitations, the comparison is shown in Table 5.

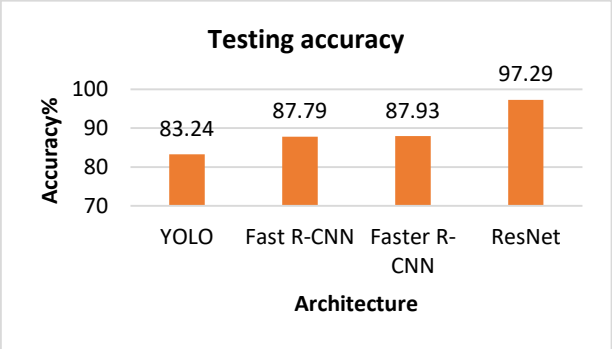

Figure 8. Accuracy of different architectures on the testing set

Table 1. Hyperparameters of the Yolo algorithm

Method	Hyperparameters					Training Accuracy	Validation Accuracy
	Learning rate	Batch size	Number of epochs	Confidence threshold	Intersection over union		
Grid search	**0.001**	**16**	**1000**	**0.7**	**0.9**	**84.45%**	**82.22%**
Random Search	**0.001**	**4**	**1200**	**0.7**	**0.92**	**85.32%**	**83.51%**
Bayesian optimization	**0.012**	**64**	**3200**	**0.5**	**0.8**	**87.21%**	**86.43%**

Table 2. Hyperparameters of Fast R-CNN

Method	Hyperparameters						Training Accuracy	Validation Accuracy
	Learning rate	Batch size	Number of epochs	Momentum	Weight decay	Non-Maximum Suppression		
Grid search	**0.0001**	**16**	**120**	**0.93**	**0.0005**	**0.5**	**87.17%**	**85.67%**
Random Search	**0.0001**	**16**	**750**	**0.91**	**0.0005**	**0.3**	**86.32%**	**86.00%**
Bayesian optimization	**0.0001**	**32**	**920**	**0.90**	**0.0005**	**0.5**	**89.43%**	**89.21%**

Table 3. Hyperparameters of Faster R-CNN

Method	Hyperparameters						Training Accuracy	Validation Accuracy
	Learning rate	Batch size	Number of epochs	Momentum	Weight decay	Non-Maximum Suppression		
Grid search	**0.001**	**32**	**100**	**0.95**	**0.0005**	**0.5**	**89.41%**	**88.13%**
Random Search	**0.001**	**32**	**150**	**0.93**	**0.0005**	**0.3**	**93.76%**	**90.65%**
Bayesian optimization	**0.001**	**64**	**180**	**0.99**	**0.0005**	**0.5**	**91.43%**	**90.23%**

Table 4. Hyperparameters of ResNet

Method	Hyperparameters					Training Accuracy	Validation Accuracy
	Learning rate	Batch size	Number of epochs	Weight decay	Dropout rate		
Grid search	**0.001**	**32**	**30**	**0.0005**	**0.35**	**98.7%**	**98.21%**
Random Search	**0.001**	**32**	**50**	**0.0005**	**0.3**	**97.78%**	**95.32%**
Bayesian optimization	**0.001**	**64**	**35**	**0.0005**	**0.40**	**97.43%**	**95.89%**

Table 5. performance comparison of the discussed models

Algorithm	Accuracy (%)	Strengths and Examples of Successful Detection	Limitations
ResNet	95.89	- Precise detection of intricate and subtle defects (e.g., micro-cracks, surface irregularities, porosity)	- May have longer processing times
Faster R-CNN	90.23	- Good balance between speed and accuracy	- May struggle with smaller, less prominent defects
Fast R-CNN	89.21	- Decent speed-accuracy trade-off	- Challenges with complex textured defects, partial obstructions
YOLO	86.43	- Real-time capabilities	- Difficulty in localizing smaller defects, reliance on object size

The above table provides a clear comparison of the algorithms' performance, highlighting their respective strengths and limitations in detecting surface defects in investment casting.

In summary, the choice of algorithm should be tailored to the specific requirements of the application. ResNet's accuracy makes it a strong candidate for intricate defect detection, while Faster R-CNN and Fast R-CNN strike a balance between speed and accuracy. YOLO is ideal for real-time scenarios but may require additional support for precise localization of smaller defects.

These examples illustrate how each algorithm performs in detecting surface defects in investment casting, shedding light on their strengths and limitations. The results showcase the potential of deep learning in industrial applications, emphasizing the need for algorithm selection based on the nature of the defects and the operational context.

Eventually, while the algorithms were not specifically evaluated for identifying the precise defect type, our study aimed to establish their effectiveness in the broader context of defect detection, which is a fundamental aspect of automated quality control in various industries.

5. CONCLUSION:

In conclusion, the results of this study demonstrate that deep learning algorithms can effectively detect surface defects in investment casting. Among the algorithms tested, ResNet achieved the highest accuracy rate of 95.89%, followed by Faster R-CNN at 90.23%, Fast R-CNN at 89.21%, and YOLO at 86.43%. The high accuracy rate achieved by ResNet can be attributed to its ability to learn features at different scales and to capture more complex relationships between features. However, it is essential to note that each algorithm has advantages and limitations in detecting defects in investment casting. Faster R-CNN and Fast R-CNN have a faster detection speed than ResNet but are less accurate in detecting more minor defects. YOLO has a faster detection speed but could be more accurate in detecting more significant defects. Overall, the results of this study provide valuable insights into the use of deep learning algorithms for surface defect detection in investment casting and can be used as a basis for further research in this area.

References:

[1] M. Wieczorowski, D. Kucharski, P. Sniatala, P. Pawlus, G. Krolczyk, and B. Gapinski, “A novel approach to using artificial intelligence in coordinate metrology including nano scale,” *Measurement*, vol. 217, p. 113051, 2023, doi: https://doi.org/10.1016/j.measurement.2023.113051.

[2] R. Marudhamuthu, M. krishnaswamy, and D. M. Pillai, “The Development and Implementation of Lean Manufacturing Techniques in Indian garment Industry.,” *Jordan Journal of Mechanical and Industrial Engineering*, vol. 5, no. 6, pp. 527–532, Dec. 2011, [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&db=asr&AN=73944047&site=ehost-live

[3] M. Soori and M. Asmael, “A review of the recent development in machining parameter optimization,” *Jordan Journal of Mechanical and Industrial Engineering*, vol. 16, no. 2, pp. 205–223, 2022.

[4] M. T. Hayajneh, M. S. Tahat, and J. Bluhm, “A study of the effects of machining parameters on the surface roughness in the end-milling process,” *Jordan Journal of Mechanical and Industrial Engineering*, vol. 1, no. 1, 2007.

[5] O. T. R. Almeanazel, “Total productive maintenance review and overall equipment effectiveness measurement,” *Jordan Journal of Mechanical and Industrial Engineering*, vol. 4, no. 4, 2010.

[6] A. R. I. Kheder, G. S. Marahleh, and D. M. K. Al-Jamea, “Strengthening of Aluminum by SiC, Al2O3 and MgO.,” *Jordan Journal of Mechanical and Industrial Engineering*, vol. 5, no. 6, 2011.

[7] Q. Zhang, Y. Wang, W. Lin, Y. Luo, and X. Wu, “Contact Mechanics Analysis and Optimization of Shape Modification of Electric Vehicle Gearbox.,” *Jordan Journal of Mechanical and Industrial Engineering*, vol. 14, no. 1, 2020.

[8] G. Dixit and M. M. Khan, “Sliding Wear Response of an Aluminium Metal Matrix Composite: Effect of Solid Lubricant Particle Size.,” *Jordan Journal of Mechanical and Industrial Engineering*, vol. 8, no. 6, 2014.

[9] S. Perri, F. Spagnolo, F. Frustaci, and P. Corsonello, “Welding defects classification through a Convolutional Neural Network,” *Manuf. Lett.*, vol. 35, pp. 29–32, 2023, doi: https://doi.org/10.1016/j.mfglet.2022.11.006.

[10] T. P. Nguyen, S. Choi, S.-J. Park, S. H. Park, and J. Yoon, “Inspecting method for defective casting products with convolutional neural network (CNN),” *Int. J. Precis. Eng. Manuf. Technol.*, vol. 8, pp. 583–594, 2021.

[11] J.-K. Kuo, J.-J. Wu, P.-H. Huang, and C.-Y. Cheng, “Inspection of sandblasting defect in investment castings by deep convolutional neural network,” *Int. J. Adv. Manuf. Technol.*, vol. 120, no. 3–4, pp. 2457–2468, 2022.

[12] S. W. Lawson and G. A. Parker, “Intelligent segmentation of industrial radiographic images using neural networks,” in *Machine Vision Applications, Architectures, and Systems Integration III*, 1994, vol. 2347, pp. 245–255.

[13] J. Li, Z. Su, J. Geng, and Y. Yin, “Real-time Detection of Steel Strip Surface Defects Based on Improved YOLO Detection Network,” *IFAC-PapersOnLine*, vol. 51, no. 21, pp. 76–81, 2018, doi: 10.1016/j.ifacol.2018.09.412.

[14] S. Oh, J. Cha, D. Kim, and J. Jeong, “Quality inspection of casting product using CAE and CNN,” in *2020 4th International Conference on Imaging, Signal Processing and Communications (ICISPC)*, 2020, pp. 34–38.

[15] X.-G. Wang, J.-S. Wang, P. Tang, and W.-Y. Liu, “Weakly- and Semi-Supervised Fast Region-Based CNN for Object Detection,” *J. Comput. Sci. Technol.*, vol. 34, no. 6, pp. 1269–1278, 2019, doi: 10.1007/s11390-019-1975-z.

[16] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[17] X. Kou, S. Liu, K. Cheng, and Y. Qian, “Development of a YOLO-V3-based model for detecting defects on steel strip surface,” *Measurement*, vol. 182, p. 109454, 2021, doi: https://doi.org/10.1016/j.measurement.2021.109454.

[18] V. G. Raj, M. Srihari, and A. Mohan, “Casting defect detection using YOLO V4,” *Int. Res. J. Mod. Eng. Technol. Sci*, vol. 3, no. 4, pp. 1581–1585, 2021.

[19] Z. Guo, C. Wang, G. Yang, Z. Huang, and G. Li, “Msft-yolo: Improved yolov5 based on transformer for detecting defects of steel surface,” *Sensors*, vol. 22, no. 9, p. 3467, 2022.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[21] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[22] W. Zhao, F. Chen, H. Huang, D. Li, and W. Cheng, “A new steel defect detection algorithm based on deep learning,” *Comput. Intell. Neurosci.*, vol. 2021, 2021, doi: 10.1155/2021/5592878.

[23] Q. Ren, J. Geng, and J. Li, “Slighter Faster R-CNN for real-time detection of steel strip surface defects,” in *2018 Chinese Automation Congress (CAC)*, 2018, pp. 2173–2178. doi: 10.1109/CAC.2018.8623407.

[24] A. Goyal, S. Dhiman, S. Kumar, and R. Sharma, "A Study of Experimental Temperature Measuring Techniques used in Metal Cutting.," *Jordan Journal of Mechanical and Industrial Engineering*, vol. 8, no. 2, 2014.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: Generalizing residual architectures," *arXiv Prepr. arXiv1603.08029*, 2016.

[27] M. Kaushal, "Rapid-YOLO: A novel YOLO based architecture for shadow detection," *Optik (Stuttg).*, vol. 260, p. 169084, 2022.

[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.