International Journal of Advances in Applied Sciences (IJAAS)
Vol. 14, No. 3, September 2025, pp. 838~848
ISSN: 2252-8814, DOI: 10.11591/ijaas.v14.i3.pp838-848

Deep learning for image classification of submersible pump impeller

Phan Nguyen Ky Phuc[1], Doan Huu Chanh[1], Trong Hieu Luu[2]
[1]School of Industrial Engineering and Management, International University, Viet Nam National University, Ho Chi Minh City, Vietnam
[2]Faculty of Automation Engineering, College of Engineering, Can Tho University, Can Tho City, Vietnam

Article Info

Article history:

Received Dec 16, 2024
Revised May 19, 2025
Accepted Jun 8, 2025

Keywords:

Defect detection
Ensemble model
Quality inspection
ResNet50
Visual geometry group 16

ABSTRACT

This study presented a deep learning-based model in the submersible pump impellers quality inspection process. The proposed method aimed to relieve worker workload, automate the system, as well as increase the accuracy in defect detection and classification. The proposed approach aims to be implemented on systems with low investment cost and limited resources, i.e., small single-board computers, enabling flexible deployment in industrial environments. The model consisted of three convolutional neural network (CNN) models, i.e., visual geometry group 16 (VGG16), ResNet50, and a custom model. The outputs of three networks were either synthesized later through an ensemble stage or used separately. A graphical user interface (GUI) was also developed for real-time inspection and user-friendly interaction. The approach achieved up to 99.8% accuracy in identifying defects, including surface scratches, corrosion, and geometric irregularities. The proposed method improved the quality assurance process by reducing manual inspection efforts. Future research could explore advanced techniques like anomaly detection to further enhance system performance and versatility.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Trong Hieu Luu
Faculty of Automation Engineering, College of Engineering, Can Tho University
Can Tho City, Vietnam
Email: luutronghieu@ctu.edu.vn

1. INTRODUCTION

In modern industry, manual inspections are time-consuming and error-prone. In order to provide reliable and high-quality products, inspections should be automated and apply innovative technologies. Currently, computer vision and deep learning techniques are two candidate methods that are widely adopted by industry due to their prominent advantages, such as low cost and implementation simplicity. In the computer vision field, the convolutional neural network (CNN) has proved its huge value through several applications in both manufacturing and inspection processes.

According to several former studies, deep learning systems outperformed regular machine learning systems in pattern recognition, computer vision, and image processing. Simonyan and Zisserman [1] examined CNN and numerous architectures, like as LeNet, AlexNet, and GoogleNet, on the large ImageNet datasets. They deduced from this study that the amount of data may directly impact the number of epochs and accuracy of the selected model. Moreover, He *et al.* [2] provided a methodology for product inspection and testing based on deep learning approaches. Based on the goals of existing product inspection systems, they provided an effective method for sustaining and enhancing a product inspection system. Due to the approach provided, the proposed system was seen to have good system maintenance and stability. In addition, utilizing a customized AlexNet-based CNN. Krishna and Kalluri [3] suggested a method for classifying defects in yarn-dyed fabrics. The experimental findings demonstrated significant resilience in yarn-dyed fabric fault classification and a promising average classification rate. Kim *et al*. [4] also presented a distinctive recognition strategy for steel surface faults based on upgraded artificial neural network algorithms, including feature visualizing, along with accuracy evaluation. The steel surface defect classification problem was pre-trained using the visual geometry group 16 (VGG19), and the matching DVGG19 was developed to extract the feature pictures in various layers from the defect weight model. Then, a new virtualized services directory (VSD) network was developed and utilized to classify steel surface flaws. The experiment findings pointed out that the suggested approach may significantly increase average classification accuracy, and the model can converge fast, which was beneficial for steel surface defect identification utilizing a VSD network model of feature visualization and quality evaluation. For automatic fruit grades classification, Jing *et al* [5] investigated the influence of several complex CNN structures on the reliability of a strawberry grading system (quality inspection). Then, they examined several types of current deep CNN architectures, such as AlexNet, MobileNet, GoogLeNet, VGGNet, and Xception, compared to a two-layer CNN architecture. According to the results, VGGNet had the highest accuracy, whereas GoogLeNet had the most computationally efficient design. Both the two-class classification and the four-class classification showed the same findings. Guan *et al*. [6] developed an injection moulding quality inspection process system in edge intelligence. As a result, the mentioned model's accuracy was greater than 90%, demonstrating that the system may be used in the field.

In recent years, computer performance has improved dramatically, leading to significant advancements in deep learning technology. Deep learning is capable of automatically learning complex features, giving it strong generalization abilities and making it highly effective in various object detection tasks [7], [8]. As a result, defect detection methods based on deep learning are gaining more attention. This field can be divided into object detection and object segmentation. Du *et al*. [9] enhanced the Faster R-CNN network by incorporating feature pyramid networks (FPN) and region of interest (RoI Align), enabling the detection of defects in X-ray images of automotive cast aluminum parts. Similarly, Xue *et al*. [10] used techniques like Overlap and Mosaic to expand the training dataset, achieving accurate detection of various casting defects with the you only look once, version 3 (YOLOv3) model. Duan *et al*. [11] added an spatial pyramid pooling (SPP) layer to YOLOv3 before the final convolutional layer. Experimental results demonstrated a significant improvement in the mean average precision (mAP) for recognizing casting digital radiography (DR) image defects, reaching 88.02% compared to the original network [12]. Cha *et al*. [13] and Cui *et al*. [14] introduced a CNN that combines faster R-CNN with a region proposal network (RPN), enabling the detection of multiple types of damage simultaneously at a remarkable speed of just 0.03 seconds per image. On the other hand, the SDDNet network incorporates a feature refinement module (FRB) and a skip-layer connection module to handle various texture defects [15], [16]. However, this model has limitations, such as missed detections when dealing with targets that have strong background noise or unclear texture details. It also suffers from low segmentation accuracy and limited generalization ability when identifying both workpieces and defective conditions.

This study looked into the effects of CNN architecture on industrial product inspection and classification. While previous studies investigated the impact of deep learning techniques on these tasks, they did not explicitly address their influence on system deployment using single-board computers. Through these studies, CNN and deep learning techniques have proven effective for product inspection and classification. However, these approaches require systems with fast computational capabilities and high RAM, often leading to high investment costs and power consumption, which hinder their industrial applications. To address this limitation, this study focused on CNN architectures with appropriate sizes, enabling easy deployment on single-board computers such as BeagleBone, Raspberry Pi, or Orange Pi. These optimizations enhance the practicality of CNN and deep learning for industrial inspection. The proposed mechanism also ensures system sustainability, which is a critical factor in determining the adoption of the method. Since the proposed system operates independently of the main system and each module functions autonomously, the inspection system remains operational even if a module fails or needs testing before integration. Figure 1 illustrates the system design applied in this study.

We focused on automating inspections of submersible pump impellers, which are error-prone due to various casting defects. By implementing different CNN architectures such as VGG16 [7], ResNet50 [8], and one custom model, the study investigated the effectiveness of using a non-pretrained CNN under the circumstance of limited data, which is very common in industry. Moreover, during research, the transfer learning and ensemble methods were also considered. Under transfer learning, only the output layer is trained while other layers are frozen. This strategy reduces overfitting, the training time, and utilizes the former well pre-trained model. The returned results of all models were fused later in the ensemble stage to enhance the accuracy and efficiency of defect detection and classification. From the industrial view, each model played a role as a module that contributed directly to the success of the classifier. During the operation, ineffective and inaccurate modules can be checked and replaced without creating any serious effect on the whole system.

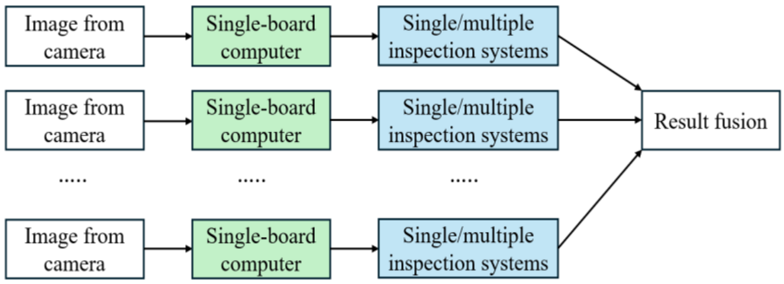


Figure 1. The system models

Additionally, a graphical user interface (GUI) was developed to improve usability and real-time decision-making capabilities. This research addressed the limitations of manual inspection, offered a deep learning-based approach for quality inspection. The integration of CNN models and GUI technology contributed to more efficient and accurate defect identification, paving the way for smarter production and inspection systems in the era of Industry 4.0 and manufacturing digitization. In the rest of the paper, the second section mentioned proper literature reviews, while the next section described the used data set and proposed methodology. The conclusion was provided in the final section.

2. RESEARCH METHOD

2.1. Dataset information

The dataset used in this study comprises 7,348 images with dimensions of 300×300 and was obtained from the "Real-life industrial dataset of casting product" on Kaggle [17]. All of the 6,633 images were used for training and validation, with 3,758 representing defective and 2,875 representing non-defective pump impellers. An 80/20 split was used to divide the training and validation data. The test folder contained 453 images of faulty pump impellers and 262 images of non-faulty pump impellers. Figure 2 provides a short dataset summary.

This data set was already used in some works, such as Ekambaran and Ponnusamy [18], Alfarizi *et al*. [19], Wang and Jing [20], Sundaram and Zeid [21], or Hu *et al*. [22], to name a few. Ekambaran and Ponnusamy [18] used multipath DenseNet and ResNet34 for product classification. Alfarizi *et al*. [19] compared the accuracy of k-nearest neighbors (KNN) and naive Bayes algorithms in detecting defects in impeller products. Using t-distributed stochastic neighbor embedding (t-SNE) visualization, the study concluded that KNN was more reliable for defect detection in industrial applications. Wang and Jing [20] introduced the coordinate attention mechanism into the backbone network to allocate more attention to the defect target. The research also used the bidirectional weighted feature pyramid network in the feature fusion network to replace the original path aggregation network, improving the model's ability to fuse features of different sizes. Sundaram and Zeid [21] introduced the quality control system using one custom CNN model for inspection and a computer application that can be deployed on the shop floor. Hu *et al*. [22] adopted the Xception model to create a robust classification system. The study also applied data augmentation techniques to enhance the dataset in Figure 3, allowing the model to generalize more effectively and improve its defect recognition capabilities. Augmentation techniques have been applied to all the images in the dataset to enhance the diversity and variability of the data. The images were labelled with tags indicating whether they are classified as "ok" (normal, as shown in Figure 3(a)) or "def" (defect/anomaly, as shown in Figure 3(b)).

2.2. Dataset training model

In this study, VGG16, ResNet50, and a custom model based on a CNN structure were adopted for training. The returned outputs from each network were sent later to the ensemble stage for making final decisions or used separately. In the ensemble stage, the final label was decided by majority rule [23]. The main use of the ensemble technique was to improve the overall performance of the entire system if it were required. By combining several independent weak classifiers, the ensemble method can create a strong classifier with higher sensitivity. The proposed structure of the classifier is shown in Figure 4.

In addition, PyQt5 was employed to create a GUI. The GUI functioned as an interactive image classification tool, allowing users to input images of pump impellers and receive real-time outcomes for classification. In this research, the dataset was trained using various CNN architectures, including VGG16, ResNet50, and a custom model. The custom model consisted of many convolutional layers activated by ReLU, followed by max pooling for dimension reduction. The first layer employed 32 3×3 filters with ReLU activation. In 2×2 windows, outputs were routed through max pooling. This method was repeated two more times. The final maximum pooling outputs were flattened and sent to a dense layer with sigmoid activation to reduce dimensionality. For probability distribution over classes, the last dense layer employed SoftMax activation.

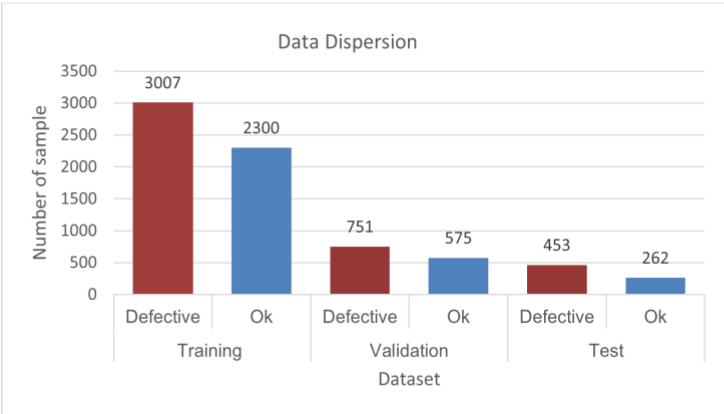


Figure 2. Data dispersion

(a) (b)

Figure 3. Random images of pump impellers from the dataset of (a) defective and (b) normal

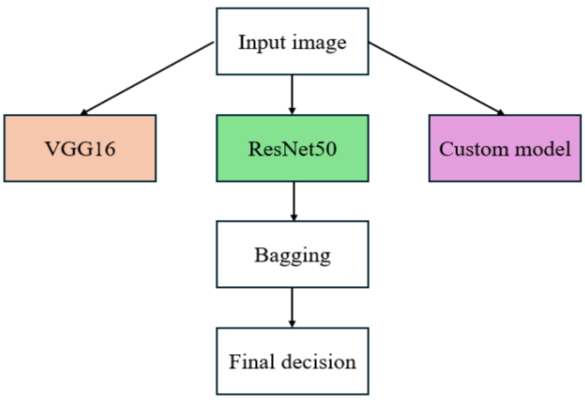


Figure 4. The proposed structure of the classifier

In general, all three considered models contained several two-dimensional convolution layers. They are the key operations in CNN that are responsible for generating a feature map that captures important patterns or features in the input image. The operation can be represented mathematically as (1).

$$(f * g)[x, y] = \sum_{i,j=-\infty}^{\infty} f[i, j]. g[x - i, y - i] \quad (1)$$

Where f is the input image and g is the kernel, x and y are the indices or coordinates of the output function resulting from the convolution. They determine the specific location in the output where the convolution operation is applied. Activation functions for different networks in this study include rectified linear unit (ReLU) [24] and sigmoid [25] functions, which are shown in (2)-(3), respectively. Furthermore, at the output layer, the SoftMax [26] activation function was adopted as in (4).

$$f(x) = max(0, x) \quad (2)$$

$$\sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} \quad (3)$$

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (4)$$

Where $\sigma(z_i)$ represents the output probability of the i^{th} class, e^{z_i} is the exponential function applied to the i^{th} class output, and $\sum_{j=1}^{K} e^{z_j}$ represents the sum of exponential functions applied to all class outputs. Two distinct functions were used to compute the loss function. The first is sparse categorical cross-entropy, which is appropriate for multi-class classification jobs where the target variable is integer encoded. This was used in the custom model stated in (5).

$$L = -\sum_{i=1}^{N} t_i * \log(p_i) \quad (5)$$

Where L represents the loss value, N is the number of classes, t_i is the true label and p_i is the predicted probability distribution over i^{th} the classes. The second loss function is binary cross-entropy, which is often used for binary classification tasks and is employed in VGG16 and ResNet50. The binary cross-entropy loss function is mathematically represented by (6).

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i * log\,(\hat{y_i}) + (1 - y_i) * log(1 - \hat{y}_i) \quad (6)$$

Where y_i is the target label for the i^{th} sample and $\hat{y}_i$ is the predicted probability by the model for the i^{th} sample belonging to the positive class.

To utilize the power of these successful networks as well as to reduce the training time of these models, transfer learning and ensemble techniques were applied in later steps. For VGG16 and ResNet50, only the last output layer was retrained while their other layers were frozen. Stochastic gradient descent (SGD) and Adam optimization methods were used to update network parameters during training. SGD changes parameters based on gradients determined on a subset of training data, whereas Adam combines adaptive learning rates and momentum. Training used a batch size of 32 to maximize computational efficiency. Each architecture was trained for nine epochs, allowing the network to learn and modify weights and biases over time. An epoch was one trip through the complete training dataset. This method optimized the loss function, changed parameters, and increased convergence speed.

Moreover, the GUI created with PyQt5 was also developed for constructing interactive and user-friendly programs. The GUI allows the user to upload photos and select the desired architecture for inspection. When the application is launched, the user is presented with an intuitive interface in which they can browse and upload images from their local system. The selected photos were then inspected using the favored architecture of choice (e.g., VGG16, ResNet50, the custom model, or ensemble models). By utilizing the trained models within the GUI, users can easily evaluate the condition of pump impellers by simply uploading an image. The chosen model will process the image and provide the predicted classification, indicating whether the impeller is defective or non-defective. Figure 5 shows the developed GUI, and the classification log is shown in Table 1. After finishing the inspection, the GUI gives the user the option to export the results to an Excel file. The produced Excel file comprises the recognized conditions of the tested products, making it easy for further analysis and record-keeping. The PyQt5-based GUI provided a smooth and interactive experience, allowing users to quickly submit photos, choose the architecture, and export inspection results. It improved usability and allowed for more effective data handling and analysis for quality control.

Table 1. The classification log

Prod.	ID	Time	Date	VGG	ResNet	Custom	Ensemble
Impeller	973	16:45	31-05-2023	Ok	Ok	Ok	Ok
Impeller	27	16:45	31-05-2023	Def	Def	Def	Def
Impeller	50	16:45	31-05-2023	Def	Def	Def	Def
Impeller	95	16:45	31-05-2023	Def	Def	Def	Def
Impeller	71	16:46	31-05-2023	Ok	Ok	Ok	Ok
Impeller	118	16:46	31-05-2023	Ok	Ok	Ok	Ok
Impeller	26	16:46	31-05-2023	Def	Def	Def	Def
...	...	...	...	...	...	...	...

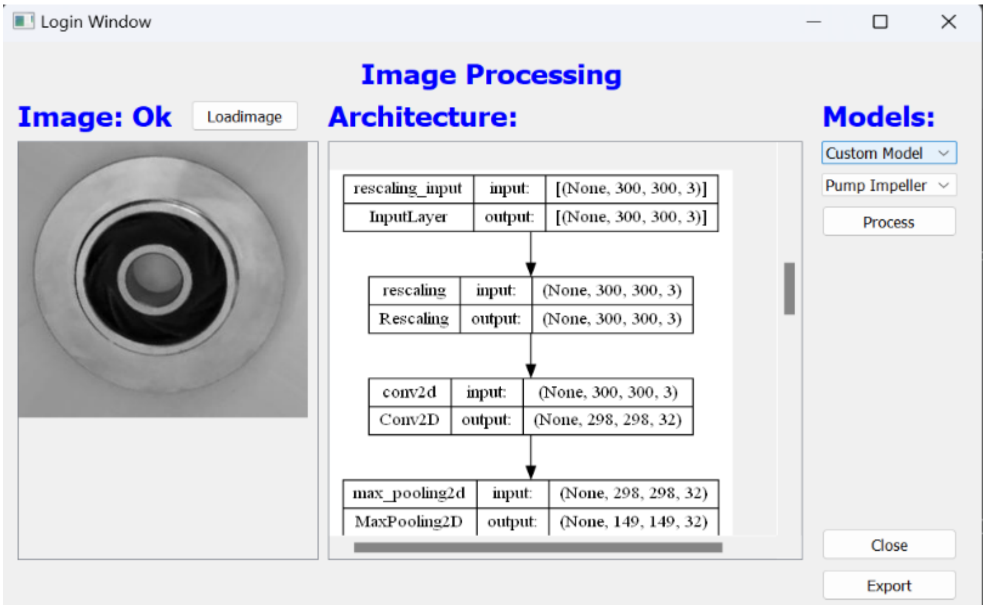


Figure 5. The developed GUI

3. RESULTS

3.1. Model performances

The evaluation of the proposed architectures was conducted using a confusion matrix, accuracy, precision, recall, and F1-score. These evaluation metrics provided a comprehensive performance of the models, valuable insights about their capabilities, as well as their strengths and weaknesses. The accuracy, precision, recall, and F1-score are computed by (7)-(10), respectively.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{7}$$

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

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

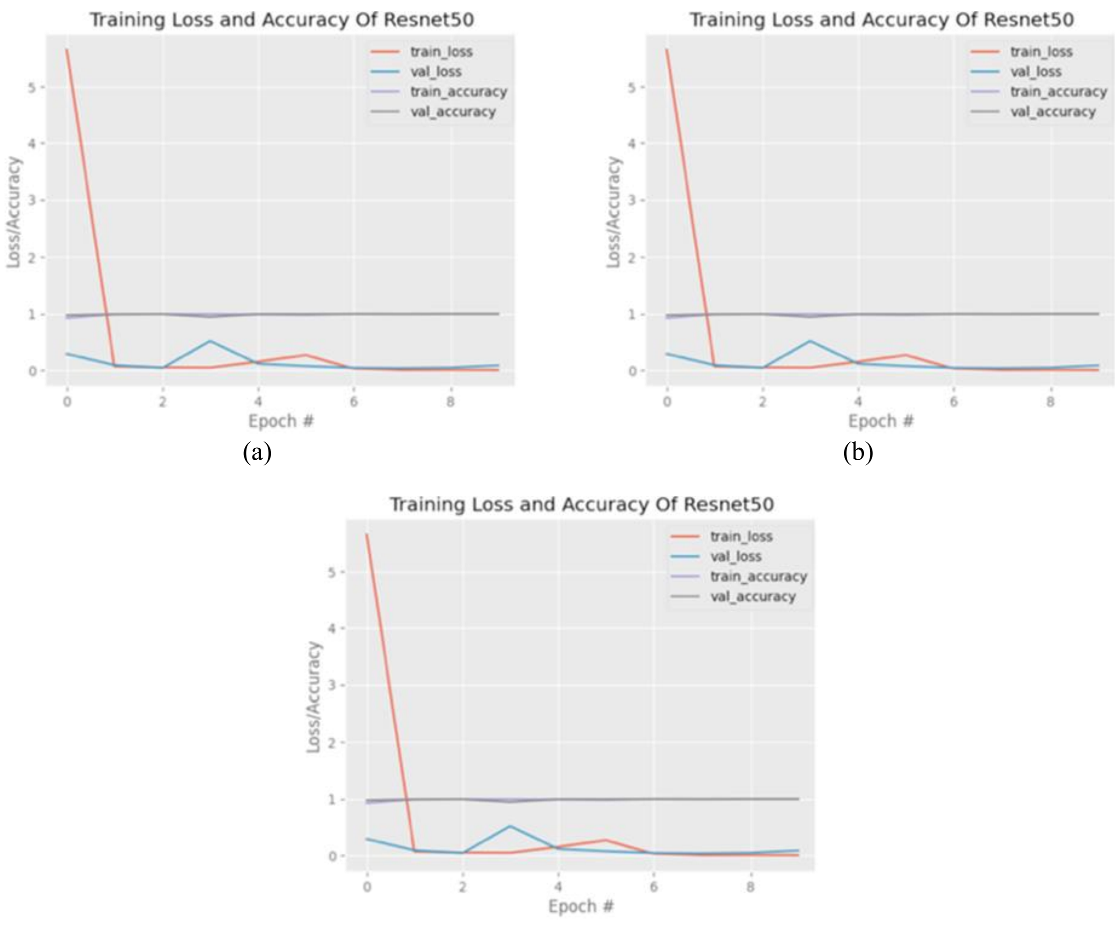
$$F1-score = \frac{2*Recall*Precision}{Recall+Precision} \tag{10}$$

Figure 6 provided details about the training process of three models. Table 2 summarizes the performance of three models after 9 epochs. We found that training accuracy correlates with the use of pre-trained models, as transfer learning enables faster convergence. The proposed method in this study tended to have an inordinately higher proportion of training efficiency, as VGG16 achieved 100% training accuracy and 99.24% validation accuracy at the 9th epoch, while ResNet50 reached 99.92% training accuracy and 99.47% validation accuracy at the same epoch. At this point, the custom model obtained 98.79% training accuracy and 99.10% validation accuracy. VGG16 and ResNet50 converged more quickly due to transfer learning, whereas the custom model required a longer training period since all parameters had to be trained from scratch, and the dataset used was not as comprehensive as the one applied for VGG16 and ResNet50.

Table 2. Summarizes the performance of three models after 9 epochs

Model	Training loss	Validation loss	Training accuracy	Validation accuracy
VGG16	0.002	0.030	1.000	0.992
ResNet50	0.003	0.008	0.999	0.995
Custom model	0.007	0.010	0.987	0.971

From the Figure 6, all VGG16 (Figure 6(a)), ResNet50 (Figure 6(b)), and the custom model (Figure 6(c)) all had high performance in picture classification. Consequently, all models can serve as a backbone which are responsible for feature extraction. Their output layer can be substituted with any proper classification method. In addition, the feature vectors from backbones can also be stacked to form a unique feature vector, which can be used as input for any classifier. Depending on the applied method of the classifier, independent models can interact with others in several ways and enhance the accuracy of the synthesis model. This study only examined the most common way to synthesize these results by using the bagging method, i.e., the majority voting rule. This has the advantage of ease when the modifications are required. The inclusion and exclusion of a network from the ensemble can be done very quickly and conveniently without any retrained activity. This cannot be achieved effectively if other methods are adopted.



(a) (b)

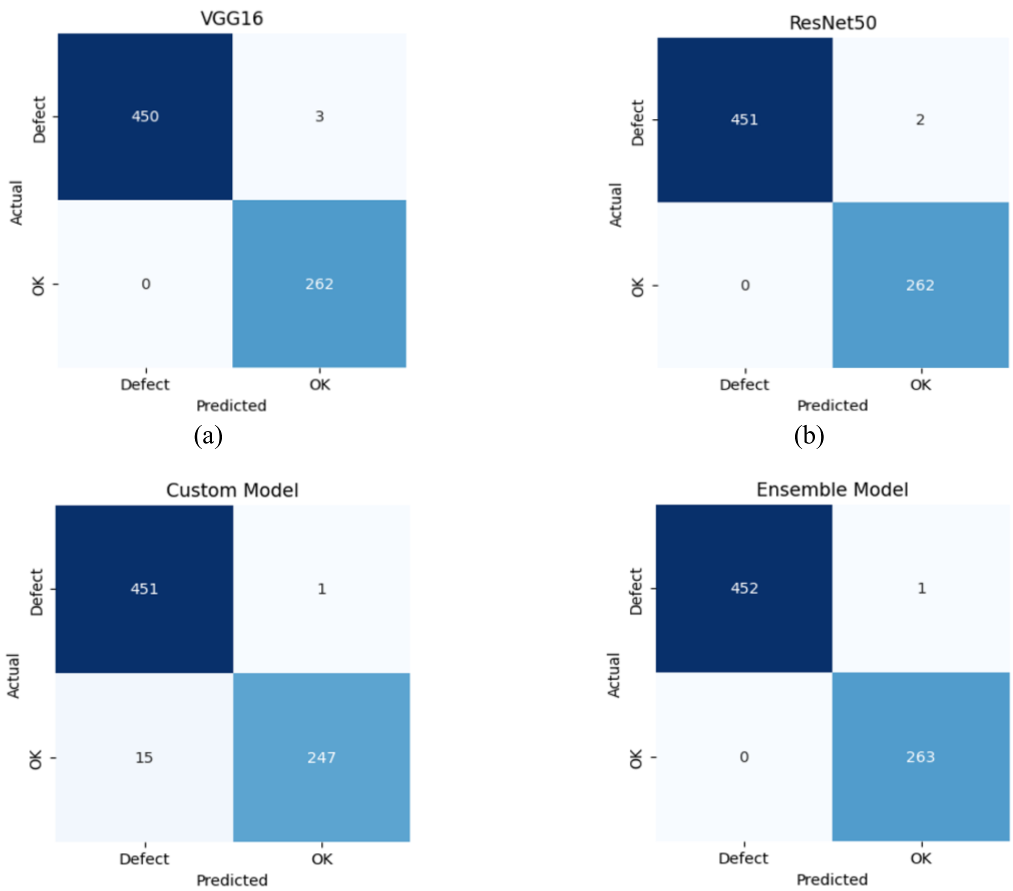
(c)

Figure 6. Training loss and accuracy of three models of (a) training loss and accuracy of the ResNet 50 model, (b) training loss and accuracy of the VGG16 model, and (c) training loss and accuracy of the custom model

3.2. Testing application

In the testing phase, the performance of the trained models was evaluated on a separate dataset containing 453 images of faulty pump impellers and 262 images of non-faulty pump impellers. Our findings indicate that higher feature sensitivity is not associated with poor performance in classifying pump impellers accurately. The proposed method may benefit from incorporating ensemble strategies without negatively affecting its generalization capability. These test images, representing real-life samples not used during training or validation, provided an unbiased benchmark for assessing the models' ability to accurately distinguish between faulty and non-faulty impellers based on the learned features and patterns. Figure 7 shows the confusion matrix of all models.

During the testing phase, the VGG16 model exhibited exceptional performance in analyzing the testing photos. It achieved impressive accuracy, correctly predicting 450 out of 453 faulty photos and accurately identifying all non-defective images (Figure 7(a)). This highlighted the model's effectiveness in distinguishing between the two classes. In Figure 7(b), the ResNet50 model demonstrated exceptional accuracy in predicting the test photos. It accurately identified 451 out of 453 faulty photos and correctly classified all non-defective images, showcasing its effectiveness in distinguishing between the two classes. These numbers for the custom model were 452 out of 453 in the defective class and 247 out of 263 in the non-defective class (Figure 7(c)), while the ensemble model had 451 out of 453 in the defective class and 263 out of 263 in the non-defective class (Figure 7(d)).



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

Figure 7. Training loss and accuracy of all models of (a) confusion matrix of the VGG model, (b) confusion matrix of the ResNet50 model, (c) confusion matrix of the custom model, and (d) confusion matrix of the ensemble model

This study investigated a comprehensive evaluation and analysis of the proposed system. However, additional and in-depth research may be required to confirm its long-term effectiveness, particularly regarding its scalability in diverse industrial environments. Although the custom models did not perform so well compared to VGG16 and ResNet50 due to the lack of pre-trained data, the proposed system has a very high sensitivity for the remaining cases. This can be attributed to the robust sensitivity of the two pre-trained networks. Given a data point belonging to class C, if all networks are independent of other networks and the sensitivity of network i regarding to class C is denoted as p_i^C, i.e., $p_i^C = P(X\ is\ classfied\ as\ C|X \in C)$, the sensitivity of the system when using the ensemble model regarding to class C is computed as in (11).

$$P^C = p_1^C p_2^C\, p_3^C + (1-p_1^C)p_2^C\, p_3^C + p_1^C(1-p_2^C)\, p_{3+}^C\, p_1^C\, p_2^C(1-p_{3)}^C \tag{11}$$

In discussion, Ekambaran and Ponnusamy [18] utilized VGG19 and ResNet34, achieving an F1-score of 99.54% with a classification time of 454 ms. Similarly, Alfarizi *et al.* [19] conducted a comparative analysis of KNN and naïve Bayesian, demonstrating that KNN achieved an accuracy of 98.11%, whereas naïve Bayesian only reached 85.38%. This finding suggests that KNN is significantly more effective for fault detection systems. Furthermore, Hu *et al.* [22] employed the Xception Aug model to enhance system stability. The study compared CNN, Inception V3, and Xception Aug, reporting accuracy rates of 98.02%, 98.48%, and 99.16%, respectively. Based on these results, the author concluded that the Xception model is the most suitable for ensuring system stability. In our contribution, we compare VGG16, ResNet50, and a custom model. The findings indicate that both VGG16 and ResNet50 achieved a training accuracy of 99%. Additionally, we developed an application capable of running on embedded computing devices, providing a practical and cost-effective classification solution for industrial manufacturing environments.

4. CONCLUSION

The study used VGG16, ResNet50, and the custom model to accurately categorize photos of casting products, specifically pump impellers. Our findings offer definitive proof that this phenomenon is linked to subtle alterations in product quality, rather than being caused by increased quantities of image data. The study employed VGG16, ResNet50, and a custom model to accurately categorize photos of casting products-specifically, pump impellers. Each model performed admirably, achieving high accuracies on both training and testing datasets. Additionally, a user-friendly graphical interface was developed using PyQt5, enabling users to input photos, select different architectures, and export the categorization results to an Excel file. The proposed design is well-suited for industrial inspection and ensures system sustainability, as it operates independently of the main system with each module functioning autonomously. Moreover, aside from the low investment cost, the system's overall accuracy in its initial stage-when data are limited-can be enhanced by leveraging well-known pre-trained models in parallel with custom models. The ensemble method also played an important role in increasing the accuracy of the model in the initial stage. So, future research may look into various datasets be used to train the models. This will aid in evaluating the models' performance on a variety of manufacturing products and determine their generalization capabilities. Furthermore, increasing the GUI functions, such as adding real-time image capturing and offering visualization tools for improved model interpretation, would improve the user experience even further. This study highlights the successful implementation of CNN architectures and ensemble methods in industrial quality control and exhibits the possibilities of using GUI technology for efficient and user-friendly picture categorization systems.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Phan Nguyen Ky Phuc	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓
Doan Huu Chanh		✓				✓	✓	✓		✓		✓		✓
Trong Hieu Luu	✓		✓	✓	✓		✓		✓	✓	✓		✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in [Kaggle] at https://www.kaggle.com/datasets/ravirajsinh45/real-life-industrial-dataset-of-casting-product/data, reference number [17].

REFERENCES

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015, pp. 1–14. [Online]. Available: https://shortlinkurl.id/8z2o8

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[3] S. T. Krishna and H. K. Kalluri, "Deep learning and transfer learning approaches for image classification," *International Journal of Recent Technology and Engineering*, vol. 7, no. 5, pp. 427–432, 2019, [Online]. Available: https://www.ijrte.org/wp-content/uploads/papers/v7i5s4/E10900275S419.pdf

[4] T. H. Kim, H. R. Kim, and Y. J. Cho, "Product inspection methodology via deep learning: an overview," *Sensors*, vol. 21, no. 15, p. 5039, Jul. 2021, doi: 10.3390/s21155039.

[5] J. Jing, A. Dong, P. Li, and K. Zhang, "Yarn-dyed fabric defect classification based on convolutional neural network," *Optical Engineering*, vol. 56, no. 09, p. 1, Sep. 2017, doi: 10.1117/1.oe.56.9.093104.

[6] S. Guan, M. Lei, and H. Lu, "A steel surface defect recognition algorithm based on improved deep learning network model using feature visualization and quality evaluation," *IEEE Access*, vol. 8, pp. 49885–49895, 2020, doi: 10.1109/ACCESS.2020.2979755.

[7] T. Wang, Y. Chen, M. Qiao, and H. Snoussi, "A fast and robust convolutional neural network-based defect detection model in product quality control," *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9–12, pp. 3465–3471, Feb. 2018, doi: 10.1007/s00170-017-0882-0.

[8] C. Dewi, R. C. Chen, and H. Yu, "Weight analysis for various prohibitory sign detection and recognition using deep learning," *Multimedia Tools and Applications*, vol. 79, no. 43–44, pp. 32897–32915, Nov. 2020, doi: 10.1007/s11042-020-09509-x.

[9] W. Du, H. Shen, J. Fu, G. Zhang, and Q. He, "Approaches for improvement of the X-ray image defect detection of automobile casting aluminum parts based on deep learning," *NDT and E International*, vol. 107, p. 102144, Oct. 2019, doi: 10.1016/j.ndteint.2019.102144.

[10] L. Xue, J. Hei, X. Chen, Y. Xu, and L. Zheng, "An efficient method of casting defects detection based on deep learning," in *Proceedings of 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications, AEECA 2020*, IEEE, Aug. 2020, pp. 480–483. doi: 10.1109/AEECA49918.2020.9213492.

[11] L. Duan, K. Yang, and L. Ruan, "Research on automatic recognition of casting defects based on deep learning," *IEEE Access*, vol. 9, pp. 12209–12216, 2021, doi: 10.1109/ACCESS.2020.3048432.

[12] Z. Guo, C. Wang, G. Yang, Huang, and G. Li, "MSFT-YOLO: improved YOLOv5 based on transformer for detecting defects of steel surface," *Sensors*, vol. 22, no. 9, p. 3467, May 2022, doi: 10.3390/s22093467.

[13] Y. J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, Sep. 2018, doi: 10.1111/mice.12334.

[14] L. Cui, X. Jiang, M. Xu, W. Li, P. Lv, and B. Zhou, "SDDNet: a fast and accurate network for surface defect detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021, doi: 10.1109/TIM.2021.3056744.

[15] R. Sustika, A. Subekti, H. F. Pardede, E. Suryawati, O. Mahendra, and S. Yuwana, "Evaluation of deep convolutional neural network architectures for strawberry quality inspection," *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 4, pp. 75–80, Dec. 2018, doi: 10.14419/ijet.v7i4.40.24080.

[16] H. Ha and J. Jeong, "CNN-based defect inspection for injection molding using edge computing and industrial IoT systems," *Applied Sciences*, vol. 11, no. 14, p. 6378, Jul. 2021, doi: 10.3390/app11146378.

[17] D. Ravirajsinh, "Casting product image data for quality inspection," Kaggle. Accessed: Jul. 01, 2025. [Online]. Available: https://www.kaggle.com/datasets/ravirajsinh45/real-life-industrial-dataset-of-casting-product?resource=download-directory

[18] D. Ekambaram and V. Ponnusamy, "Identification of defects in casting products by using a convolutional neural network," *IEIE Transactions on Smart Processing and Computing*, vol. 11, no. 3, pp. 149–155, Jun. 2022, doi: 10.5573/IEIESPC.2022.11.3.149.

[19] M. F. Alfarizi, M. Fatchan, and W. Hadikristanto, "Comparison of defective casting product classification results using the k-nearest neighbors algorithm," *International Journal of Applied Research and Sustainable Sciences*, vol. 2, no. 6, pp. 417–426, 2024, doi: 10.59890/ijarss.v2i6.1968.

[20] P. Wang and P. Jing, "Deep learning-based methods for detecting defects in cast iron parts and surfaces," *IET Image Processing*, vol. 18, no. 1, pp. 47–58, Jan. 2024, doi: 10.1049/ipr2.12932.

[21] S. Sundaram and A. Zeid, "Artificial intelligence-based smart quality inspection for manufacturing," *Micromachines*, vol. 14, no. 3, p. 570, Feb. 2023, doi: 10.3390/mi14030570.

[22] H. Hu, S. Li, J. Huang, B. Liu, and C. Che, "Casting product image data for quality inspection with Xception and data augmentation," *Journal of Theory and Practice of Engineering Science*, vol. 3, no. 10, pp. 42–46, Oct. 2023, doi: 10.53469/jtpes.2023.03(10).06.

[23] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1–2, pp. 1–39, Feb. 2010, doi: 10.1007/s10462-009-9124-7.

[24] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, pp. 807–814, 2010, [Online]. Available: https://www.cs.toronto.edu/~fritz/absps/reluICML.pdf

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[26] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 227–236. doi: 10.1007/978-3-642-76153-9_28.

BIOGRAPHIES OF AUTHORS

Phan Nguyen Ky Phuc received his B.E. degree in Mechanical in 2008 from Ho-Chi Minh City University of Technology, and his M.S. and Ph.D. degrees in Industrial Management in 2011 and 2013 from National Taiwan University of Science and Technology. He is currently a professor at the School of Industrial Engineering and Management of International University. His research interests include inventory management, ranking fuzzy numbers, dynamic programming, and meta-heuristic algorithms. He can be contacted at email: pnkphuc@hcmiu.edu.vn.

Doan Huu Chanh received his B.E. degree in Logistics and Supply Chain Management in 2023 from Ho Chi Minh City International University. He is currently a graduate student in the School of Industrial Engineering and Management at the National Taiwan University of Science and Technology. His research interests include dynamic programming, optimization, and Markov decision processes. He can be contacted at email: doanhuuchanh11@gmail.com.

Trong Hieu Luu received his B.E. degree in Mechatronics in 2010 and M.S. degree in Automation and Control in 2016 from Can Tho University. He received the Ph.D. degree in Course of Applied Marine and Environmental Studies from Tokyo University of Marine Science and Technology in 2020. He is currently a lecturer in mechatronics at the College of Technology, Can Tho University. His research interests include computer vision, AI with image processing, and robotic technology. He can be contacted at email: luutronghieu@ctu.edu.vn.