Modeling Product Quality with Deep Learning: A Comparative Exploration

Chirumamilla Sneha[1], Rayavarapu Niharika[2], Galla Karthik[3], Dega Balakotaiah[4] and Venkata Rajulu P[5]
{snehachirumammilla99@gmail.com[1], niharikarayavarapu1@gmail.com[2], karthikgalla121@gmail.com[3], dsbalu57@gmail.com[4], venkatarajulup@gmail.com[5]}

Department of CSE, VFSTR Deemed to be University, Guntur, Andhra Pradesh, India[1, 2, 3, 4, 5]

Abstract. This work outlines a deep learning driven approach towards automatically detecting manufacturing casting defects during processes, hoping to improve industry efficiency and quality control accuracy. Submersible pump impellers classified as non-defective or defective were trained on and tested from a grayscale image dataset to benchmark different models. Preprocessing operations such as resizing, normalization, and augmentation were used to maintain consistency and stability in the input data. A tailored Convolutional Neural Network (CNN) was initially used as a baseline model, and then transfer learning was applied using three state-of-the-art architectures: EfficientNetB0, MobileNetV2, and MLP-Mixer. EfficientNetB0 obtained the highest classification accuracy of 98.75%, followed by MobileNetV2 at 98.36%. Although MLP-Mixer achieved a relatively lower accuracy of 77 08%, it provided an alternative architecture that added to the comparative study. The models were tested using standard performance measures such as accuracy, precision, recall, and F1-score. The experimental findings indicate the superiority of transfer learning in enhancing detection performance compared to traditional models. EfficientNetB0 and MobileNetV2 came out to be particularly great options for identifying casting defects with minimum misclassification. In general, the proposed system has great potential as a low cost, scalable and intelligent quality control tool with the ability to minimize human intervention and enhance defect detection in real-time production systems.

Keywords: Casting Defect Detection, Deep Learning, Transfer Learning, Convolutional Neural Network (CNN), Quality Control Automation.

1 Introduction

Maintaining high product quality is essential in modern manufacturing, as defects lead to increased costs, production downtime, and customer dissatisfaction. Manual quality inspection is often slow, inconsistent, and prone to human error. With the growing complexity of manufacturing processes, the demand for automated technologies to enhance defect detection and optimize quality control is increasing.

Deep learning has emerged as a transformative solution, offering advanced image recognition and pattern analysis capabilities for autonomous inspection systems. This work, *"*Modelling Product Quality with Deep Learning,*"* presents a comparative evaluation of several deep learning architectures for automated defect inspection in manufacturing. A labeled dataset of casting images categorized as "defective" or "non-defective" is used for model training and validation. The study employs a custom-designed Convolutional Neural Network (CNN) as

ICITSM-Part II 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2358051

the baseline model and incorporates transfer learning with three state-of-the-art pre-trained architectures: EfficientNetB0, MobileNetV2, and MLP-Mixer. Comprehensive training of these models is ensured through rigorous pre-processing, including resizing, normalization, and data augmentation.

Automated defect inspection plays a critical role in modern industry by improving accuracy and efficiency in quality control processes. Techniques such as ensemble learning and data augmentation have been shown to enhance classification performance while reducing computational costs, making them suitable for real-time applications. In electronics manufacturing, Automated Optical Inspection (AOI) has proven to be an effective alternative to manual inspections, which are both time-consuming and error-prone. Moreover, deep learning-based visual inspection methods have achieved high accuracy in defect detection, enabling real-time monitoring and reducing reliance on human intervention.

2 Related Work

Rahaman et al. [1] AOI investigations investigate a range of defect detection algorithms, from the rule-based to the higher-end feature-based and adaptive ones. The conventional approaches suffer from alignment, illumination, and classification accuracy. Innovation such as contour-based registration and adaptive templates enhances robustness and performance speed. Yet, most techniques fail at proper defect classification.

Aghapanah et al. [2]A comprehensive discussion on AI-based smart quality inspection systems that have been implemented in manufacturing reveals high accuracy of defect detection along with automation in most aspects of the inspection process, thus contributing to increased productivity and reduced error rates. But implementation difficulties in the shop floor, need for expert knowledge, and ongoing dependency on human operators for critical decisions were some challenges.

Lagrosen et al. [3] emphasized the resilience of in-line quality inspection systems, illustrating how they can inspect for faults when they occur and hence enable real-time intervention by corrective action at production. In-line quality checking systems enable higher automation, reducing the fault-ridden human factor and embracing ongoing process improvement. The study also highlighted some of the issues such as minimal collaboration in research between institutions, minimal theoretical contribution, and coordinating humans and robots is difficult. Zero Defect Manufacturing (ZDM) and multi-layer quality inspection models were also considered for evaluation as potential solutions.

Adeniran et al. [4] discussed the application of deep learning techniques to detect defects, noting their capacity to surpass traditional methods of inspection in terms of precision as well as operational efficiency. Surface defect detection using convolutional neural networks (CNN) was observed to be quite specific as they also possess flexibility to detect various defect types. Other hurdles discussed include complexity of the models and demanding computational recourses, as well as data limitations.

Miragaia et al. [5] the paper was on the utilization of AI for smart quality inspection, especially custom CNN models for high accuracy defect detection. Although it discussed automation and real-time monitoring through IoT, it revealed that there is a challenge to its implementation on the shop floor environment and calls for expert knowledge. The optimizer, adaptive moment estimation (Adam), played an important role in optimizing model performance.

Alves Gomes et al. [6] had examined the use of hybrid models combining synthetic and real data for defect detection in fields where labeled data is minimally available, resulting in improved generalization and model performance with good accuracy. Concerns about data scarcity, variability in model performance, and the proper ratio of synthetic to real data have also surfaced.

Jang et al. [7] PCB defect inspection is generally classified into two phases: Bare PCB (BPCB) and PCB Assembly (PCBA). BPCB inspection is crucial since initial defects might weaken the end product. Classic inspection processes primarily use manual or rule-based systems, whereas contemporary methods increasingly embrace AI and deep learning for efficiency and accuracy. Auto- mated Optical Inspection (AOI) has become a critical tool but still faces limitations, including high costs and computational demands. Recent advancements are set to address these issues, with AI-based approaches offering the promise of significant real-time defect detection improvements.

Jia et al. [8] covered deep learning application in robotic quality control where robot efficiency, precision, and cooperation are enhanced with human operator. The problems are the lack of real-world validation, management of data, and scalability. It has also presented models, including position control systems, digital twins, and machine learning to optimize path planning to im- prove robotic performance.

Allen et al. [9] the paper elaborated on the merits of using robots in manufacturing, especially in tasks such as quality control. The improved efficiency and accuracy endowed by robots further enhanced productivity when coupled with operators in a collaborative manner. Issues in validation, managing data, and scaling solutions across different robot models are challenges, whereas the use of position control models, digital twins, and machine learning techniques is discussed.

Li et al. [10] Machine vision and deep learning increasingly are being employed in factory inspection, precisely identifying defects and ensuring quality. Real-time, robust solutions for surface and pattern analysis are offered through new methods like FR-Net, Adaptive Adversarial Transformers, and DeepLSD. While progress is observed across various industries steel to windmills gaps in research exist, particularly in feature line detection and deep learning for feature tasks like gap spacing. This suggests major potential for future growth in smart inspection systems.

liang et al. [11] obtained reviewed deep models on defects based on CNNs, having a good accuracy and speed. Models showed the ability to adapt to varying textures and lighting conditions. However, the issues here included deep learning models being complex; resource-hungry models; and issues with data limitations, mainly related to unbalanced samples and limited dataset size.

Achmad Pratama et al. [12] these models were stressed as fairly beneficial regarding quality inspection in terms of accuracy and explainability. The defect-guided recognition neural network (DGRNN) was high-accurate and robust against visual disruption. Model overfitting and parameter efficiency were discussed as challenges, especially with regard to the CIM mechanism for im- proved performance.

Eoin et al. [13] emphasized the benefits of deep learning on surface defect inspection, including high precision and continuous monitoring of production lines. Yet, some challenges included data labeling, computational cost, and model complexity, especially with GANs. Lighter neural networks and semi- supervised learning were suggested as solutions to some of these problems.

Makris et al. [14] this approach focused on in-line quality inspection systems, with real-time detection of defects for immediate corrective measures. Automation had improved efficiency and consistency in the quality control system. Some challenges were limited cross-research-institute collaboration, the need to advance theoretical bases, and incorporating human factors in robotic systems; some models applied were zero-defect manufacturing and multilayer quality inspection.

Kong et al. [15] it discussed deep learning in surface defect inspection, particularly its high precision and continuous detection capability coupled with automation that overpowers traditional techniques of manual inspection. How-ever, it names various challenges such as data labeling problems; high computational costs; and model complexity, especially when implemented with GANs. Proposing lightweight neural networks and semi-supervised techniques could enhance the efficiency and performance of the model.

3 Proposed Work

The defect detection system is organized around a comprehensive deep learning framework that commences with image preprocessing, followed by the selection of models, their training, and subsequent evaluation. During the preprocessing phase, images are resized to dimensions of 512×512 pixels, converted to grayscale, normalized for pixel intensity, and subjected to data augmentation techniques to improve the robustness of the model. For the classification task, four distinct deep learning models were employed: CNN, MobileNetV2, EfficientNet, and MLP-Mixer. The CNN model is responsible for extracting fundamental features, while MobileNetV2 offers an efficient architecture tailored for edge computing environments. EfficientNet is designed to maximize accuracy while minimizing computational demands, and MLP-Mixer, which is based on transformer architecture, utilizes token-mixing for enhanced feature representation.

To facilitate effective learning and robust validation, the dataset was divided into training and testing subsets. The models were trained using the Adam optimizer in combination with the cross-entropy loss function, which enhanced their capability to differentiate between defective and non-defective instances. Model performance was assessed using standard classification metrics such as accuracy, precision, and recall. Among the various models evaluated, EfficientNetB0 demonstrated superior performance, striking a strong balance between accuracy and computational efficiency.

3.1 Flowchart of the Proposed Work

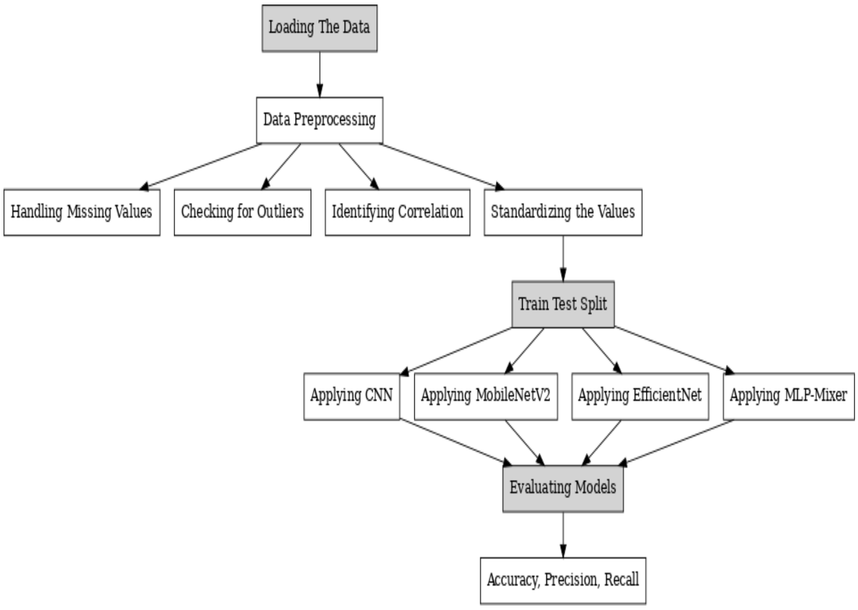

Fig. 1. Proposed Methodology for Automated Defect Detection.

The automated manufacturing defect detection system proposed employs a well-organized pipeline. The initial stage is data acquisition and preprocessing, and this involves normalizing the results, determining correlations, searching for outliers, and dealing with missing data. There are subgroups for training and testing within the sample. The four deep learning models to train the model are CNN (standard), MobileNetV2 (computational efficiency), EfficientNetB0 (high accuracy), and MLP-Mixer (novel approach). For the purpose of determining a quality inspection model minimizing human intervention and enhancing fault detection efficiency, evaluation was carried out using critical performance metrics like accuracy, precision, and recall. Fig. 1 shows the proposed methodology for automated defect detection.

3.2 Architecture of MobileNetV2

The fig. 2 shows a deep learning classification process with the use of the MobileNetV2 architecture, which is optimized for light and efficient visual classification. This architecture works particularly well for detecting defects in visual inspection applications.

Image Preprocessing

- The original RGB image of size 512×512 is resized to 224×224 pixels to align with the input specifications of the deep learning model.
- Pre-processing ensures efficient feature extraction and enhances computational feasibility.

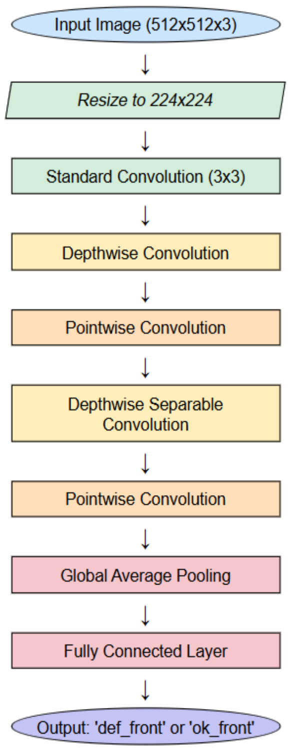

Fig. 2. Casting defect detection using MobileNetV2.

Deep Learning Model

- The pipeline uses MobileNetV2, a deep learning architecture that is optimized for efficient and real-time image classification.
- MobileNetV2 is particularly suitable for edge devices and mobile devices due to its light and efficient architecture.

Depthwise Separable Convolution

- To lower computational complexity without compromising accuracy, MobileNetV2 utilizes depthwise separable convolutions in place of traditional convolution operations.
- The operation occurs in two steps:

Depthwise Convolution: Depthwise convolution involves applying one filter to each separate input channels such that the model can effectively learn spatial information.

Pointwise Convolution (1×1 Conv): Combines the depthwise convolution output across channels for feature fusion.

Model Layers

The model follows a structured pipeline involving:

- **Standard Convolution (3×3)**: Initial feature extraction.
- **Depthwise Separable Convolutions**: Efficient representation learning.
- **Global Average Pooling (GAP)**: Reduces feature dimensions while retaining important information.
- **Fully Connected Layer**: Maps the learned features to classification labels.

Defect Classification

The final layer outputs a classification decision between:

- **'def_front'** – Defective front side.
- **'ok_front'** – Acceptable front side.

3.3 Architecture of EfficientNetB0

This fig. 3 shows the classification pipeline using EfficientNetB0, an efficient and lightweight convolutional neural network.

Input Image: Model input is an RGB image of size $512 \times 512 \times 3$.

Preprocessing: The input image is resized to $224 \times 224 \times 3$ in order to fulfill the input size requirement of the EfficientNetB0 model.

Feature Extraction: EfficientNetB0 trained on a large dataset is employed to extract deep, high-level features from the input image.

Global Average Pooling (GAP): GAP is substituted with regular fully connected layers to reduce spatial sizes of feature maps and reduce overfitting and computational cost.

Flatten Layer: Pooling layer output is flattened into a one- dimensional vector to be ready to feed into dense layers.

Dropout Layers: Dropout layers are included to reduce the likelihood of overfitting by randomly dropping out neurons during training.

Fully Connected (Dense) Layers: Two dense layers are included in the architecture: the first with 1024 neurons and with ReLU activation, the second of 512 neurons and using ReLU activation.

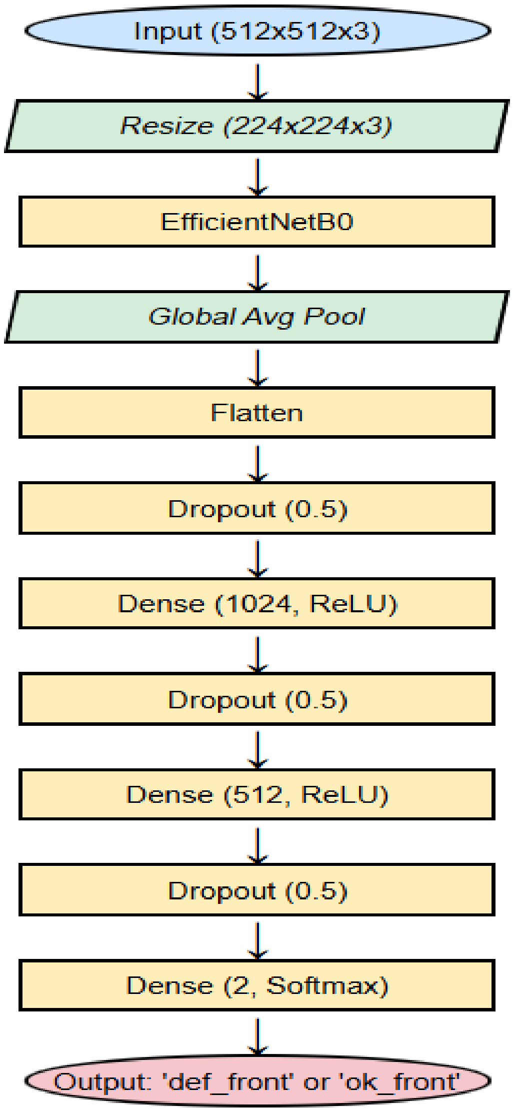

Fig. 3. Casting defect detection using EfficientNetB0.

Output Layer:

- The last dense layer is 2 neurons (corresponding to classification categories).
- Softmax activation is used to get class probabilities.

Working Pipeline

- Extract Features using the EfficientNetB0 model.
- Flatten and Pass Features through fully connected layers.
- Dropout helps improve generalization by randomly turning off neurons during training.
- Classify the Image into one of the predefined categories.

Applications

- Medical image classification.
- Object recognition tasks.
- Industrial quality inspection using deep learning.

This architecture leverages the power of EfficientNetB0, a lightweight and efficient CNN, while adding fully connected layers for classification, making it suitable for various image recognition tasks.

3.4 Architecture of CNN

This fig. 4 categorization pipeline is built upon a Convolutional Neural Network (CNN), which is a hierarchical model to train and learn semantic features richly from input images to be categorized.

Input Image: The input is an RGB image of $512 \times 512 \times 3$ size.

Preprocessing: The input image is resized to $224 \times 224 \times 3$ to satisfy the required input shape of the CNN.

Feature Extraction:

- Convolutional layers employ a number of filters to detect meaningful visual patterns and features.
- ReLU activation functions are employed to provide non-linearity so that the model can learn sophisticated feature representation

Pooling Layers: MaxPooling is employed to reduce the spatial dimensions of the feature maps without sacrificing the most significant information.

Flatten Layer: The output from the pooling layer is flattened to a one-dimensional vector in order to feed input to the fully connected layers.

Fully Connected Layers:

- The first dense layer is made up of 128 neurons that are activated by ReLU.
- There is a Dropout layer with a rate of 0.5 to minimize overfitting.
- Then a second dense layer of 64 neurons and ReLU activation comprising the last layer prior to classification.

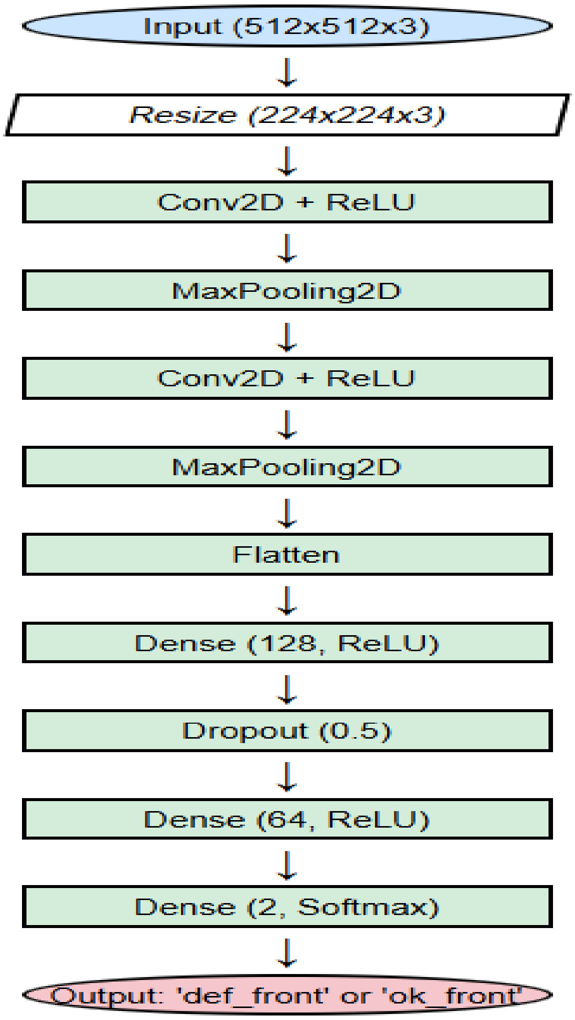

Fig. 4. Casting defect detection using CNN

Output Layer:

- The final dense layer consists of 2 neurons (corresponding to binary classification).
- Softmax activation is applied to obtain class probabilities.

Working Pipeline

- Extract Features using convolutional layers.
- Down sample using pooling layers.
- Flatten and pass features through fully connected layers.
- Apply Dropout regularization to prevent overfitting.

- Classify the image into one of the predefined categories.

Applications

- Medical image classification.
- Object detection and recognition.
- Industrial quality inspection using deep learning.

This CNN-based architecture enables automated image classification by extracting hierarchical features, making it highly effective for computer vision tasks.

3.5 Architecture of MLP-Mixer

The fig. 5 depicts the structure of MLP-Mixer, a deep learning model that replaces traditional convolutional and attention mechanisms with Multi-Layer Perceptrons (MLPs) for feature extraction. The process is outlined as follows:

Input Image: The model takes a $512 \times 512 \times 3$ RGB image as input.

Patch Embedding:

- The image is divided into fixed-size smaller patches, and each is processed as a separate token.
- These patches are then converted into lower-dimensional embeddings in order to facilitate efficient and scalable processing by the model.

Layer Normalization: Normalizes inputs across features to stabilize training and improve generalization.

Token Mixing MLP: Processes information across spatial dimensions by mixing tokens within an image.

Skip Connection: Residual connections help preserve important features and enable better gradient flow.

Channel Mixing MLP: Operates independently on each token, mixing information across feature channels.

Skip Connection: Ensures efficient gradient propagation and mitigates vanishing gradient issues.

Global Average Pooling: Reduces the size of the feature map to a single vector by computing the mean over all spatial locations.

Fully Connected Layer: Transforms extracted features into a classification decision.

Output Layer:

- The last layer consists of 2 neurons, signifying the two possible output classes.
- Softmax activation is employed to obtain the probability of each class

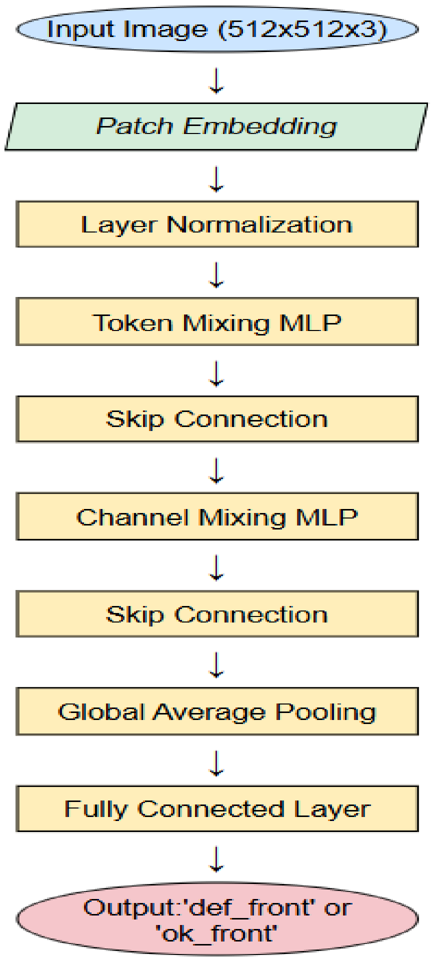

Fig. 5. Casting defect detection using MLP-Mixer.

4 Result Analysis

4.1 About Dataset

Casting is a production method where molten metal is poured into a mold with a hollow cavity in the shape of the final desired product and solidified. Although commonly used, casting is prone to numerous types of defects that can ruin the quality of the final product. The rationale for creating this dataset is to assist with the detection of such casting defects, which are discontinuities or irregularities that occur during the casting process. Typical types of defects include blow holes, pinholes, burrs, shrinkage defects, mold material flaws, improper pouring, and metallurgical irregularities. Due to the nature of these defects, much of the quality control check is left to human inspections by manufacturers. But it is labor-intensive and susceptible to error, which could result in wrong decisions. This error results in rejection of large batches, hence huge economic losses. To overcome these issues, this research proposes to automate the process of defect detection through a deep learning-based classification method.

The dataset used is 8,964 grayscale images, each 512×512 pixels, taken in a top-down perspective of submersible pump impellers. Data augmentation has been pre-applied, and the same lighting was used for every image to have constant quality. The dataset is divided into two classes: "Defective" (those containing casting defects) and "Ok" (images of defect-free impellers). The data set is divided into training and test sets for model building. The training set comprises 4,204 defective and 3,237 non-defective images, while the test set includes 899 defective and 624 non-defective samples. This dataset serves as the foundation for creating a deep learning model to improve the accuracy and efficiency of automated casting defect inspections. Fig. 6 shows the defective and non-defective manufacturing dataset.

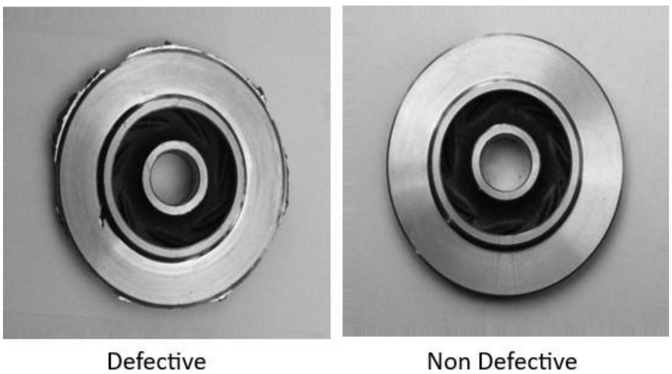

Fig. 6. Defective and Non-Defective Manufacturing Dataset.

4.2 Evaluation Metrics

To assess the effectiveness of the defect detection models, several evaluation metrics were employed to provide a comprehensive understanding of classification performance:

4.2.1 Accuracy: Represents the ratio of correct predictions to the total number of samples.

$$\text{Accuracy} = \frac{C_d + C_n}{C_d + C_n + I_d + I_n} \tag{1}$$

Where

- C_d: Correctly predicted defective samples
- C_n: Correctly predicted non-defective samples
- I_d: Non-defective samples incorrectly predicted as defective

- I_n: Defective samples incorrectly predicted as non-defective

4.2.2 Precision: Indicates the proportion of samples predicted as defective that are actually defective.

$$\text{Precision} = \frac{C_d}{C_d+I_d} \tag{2}$$

Where

- C_d: True defective predictions
- I_d: False positive predictions (non-defective labeled as defective)

4.2.3 Recall (Sensitivity): Measures the model's ability to identify actual defective samples.

$$\text{Recall} = \frac{C_d}{C_d+I_n} \tag{3}$$

Where

- C_d: True defective predictions
- I_n: False negative predictions (missed defective samples)

4.2.4 F1 Score: A harmonic mean of precision and recall, especially useful for imbalanced datasets.

$$F1\ Score = 2 \times \frac{\left(\frac{c_d}{c_d+I_d}\right)\times\left(\frac{c_d}{c_d+In}\right)}{\left(\frac{c_d}{c_d+I_d}\right)+\left(\frac{c_d}{c_d+I_n}\right)} \tag{4}$$

4.2.5 Specificity: Evaluates how well the model identifies non-defective items correctly.

$$\text{Specificity} = \frac{C_n}{C_n+I_d} \tag{5}$$

Where

- C_n: True non-defective predictions
- I_d: False positives

4.2.6 ROC Curve and AUC: The Receiver Operating Characteristic (ROC) curve plots the trade-off between true positive and false positive rates. The Area Under the Curve (AUC) provides a scalar representation of the model's classification performance.

4.2.7 Confusion Matrix: A matrix representing:

- C_d: True Positives
- C_n: True Negatives
- I_d: False Positives
- I_n: False Negatives

4.3 Mathematical Equations

4.3.1 MLP-Mixer: MLP-Mixer replaces convolutions with MLP layers for token and channel mixing.

Token Mixing: $Z' = X + W_2\sigma(W_1\text{LayerNorm}(X))$ (6)

where:

- X is the input token representation.

- W_1, W_2 are learnable weight matrices.
- σ is the activation function (GELU/ReLU).
- LayerNorm normalizes the input.

Channel Mixing: $Y = Z' + W_4\sigma(W_3\mathrm{LayerNorm}(Z'))$ (7)

Where W_3, W_4 are learnable weight matrices.

4.3.2 EfficientNetB0: EfficientNetB0 scales width, depth, and resolution via a compound scale factor φ.

Compound Scaling: $d = \alpha^{\varphi} d_0,\ w = \beta^{\varphi} w_0,\ r = \gamma^{\varphi} r_0$ (8)

where:

- d, w, r are network depth, width, and input resolution.
- d_0, w_0, r_0 are base values.
- α, β, γ are scaling coefficients.

MBConvBlock: $y = \mathrm{Conv}(x) \cdot \sigma(\mathrm{SE}(x))$ (9)

where:

- $\mathrm{Conv}(x)$ is a depthwise separable convolution.
- σ is the activation function.
- $\mathrm{SE}(x)$ represents the Squeeze-and-Excitation mechanism.

4.3.3 MobileNetV2: MobileNetV2 employs an efficient structure that uses depth- wise separable convolutions and inverted residual blocks for fast and lightweight inference.

Inverted Residual Block: This block expands the feature map dimensions using a pointwise convolution, applies depthwise convolution, and projects the result back to a lower dimension. The computation can be represented as:

$F_{out} = F_{in} + \mathrm{DWConv}(\sigma(\mathrm{Conv}_{1\times1}(F_{in})))$ (10)

Where:

- F_{in}: Input feature map
- $\mathrm{Conv}_{1\times1}$: Pointwise convolution to increase dimensions
- DWConv: Depthwise convolution applied channel-wise
- σ: Non-linear activation function (ReLU6)
- F_{out}: Output feature map after residual connection

Linear Bottleneck: In the final stage of the residual block, a linear projection is applied without activation to preserve information:

$F_{proj} = \mathrm{Conv}_{1\times1}(F_{dw}) \cdot \zeta(\mathrm{BN}(F_{dw}))$ (11)

Where:

- F_{dw}: Output of depthwise convolution
- ζ: Identity (no activation)
- BN: Batch normalization
- F_{proj}: Final output of the bottleneck block

4.3.4 Convolutional Neural Networks (CNN): CNNs use convolution and pooling operations to extract hierarchical spatial features from images, followed by dense layers for classification.

Convolution Operation: The convolution process involves a filter sliding over the image and computing dot products with overlapping regions:

$$A(i,j)=\sum\sum I(i+u,j+v)\cdot K(u,v) \tag{12}$$

Where:

- $I(i,j)$: Pixel value at position (i,j) in the input image
- $K(u,v)$: Kernel/filter value at position (u,v)
- $A(i,j)$: Activation value at output position (i,j)

Max Pooling: This operation reduces dimensionality by selecting the maximum value in each local region:

$$P(i,j)=\max(A(i+m,j+n) \tag{13}$$

Where:

- $A(i,j)$: Activation map values
- $P(i,j)$: Pooled output at position (i,j)

Dense Layer: Fully connected layers transform feature maps into final classification scores:

$$O=W\cdot X+b \tag{14}$$

Where:

- X: Flattened input vector
- W: Weight matrix
- b: Bias term
- O: Output vector for classification

4.4 Model Performance Summary

Accuracy and Loss Analysis: The performance of some of the most popular deep learning architectures in detecting defects in casting was evaluated using four architectures: a bespoke CNN, EfficientNetB0, MLP-Mixer, and Mo- bileNetV2.

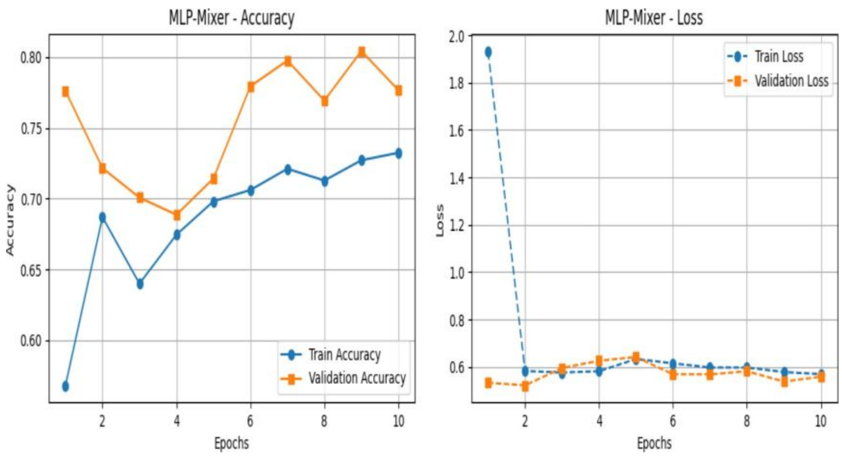

Fig. 7. Accuracy and loss curves for MLP-Mixer.

The training and validation were carried out using a dataset consisting of gray-scale images of submersible pump impellers that were either defective or non-defective. To analyze model performance and learning behavior, loss and accuracy graphs were examined, offering insight into convergence rates, generalization capacity, and suitability for real-time industrial defect classification. Several evaluation metrics were employed to understand how well each model managed class imbalance, detected flaws, and maintained a balance between recall and precision. These metrics served as indicators of the model's effectiveness and highlighted areas for possible enhancement. To further strengthen performance, optimization strategies such as tuning hyperparameters and refining preprocessing steps were considered essential for ensuring model robustness and dependable operation across various testing environments.

The MLP-Mixer model achieved a performance of 77.08%, which is significantly lower than the other models (fig. 7). The accuracy curve has sharp fluctuations, and the loss curve shows that the learning process is unstable. MLP-Mixer's architecture is token-mixing dependent, as compared to convolutional operations, which may not be appropriate for casting defect detection, where spatial hierarchies are a fundamental component. The model's performance confirms that convolution-based architectures are more appropriate for this task.

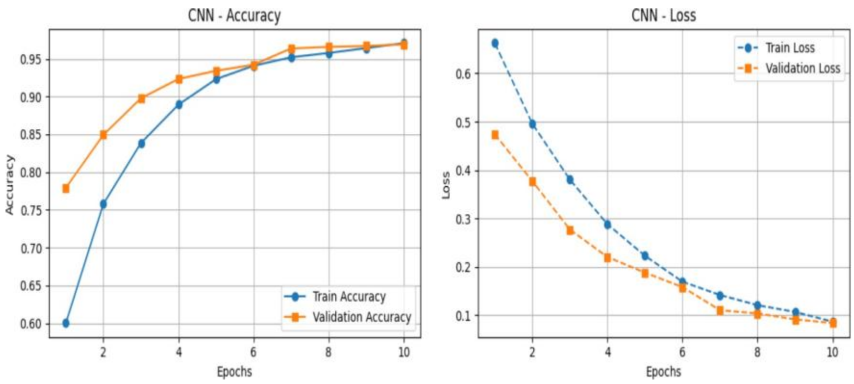

Fig. 8. Classification accuracy and loss trends for CNN.

The CNN model achieved 94.68% accuracy with the good classification performance (fig. 8). The accuracy curve and loss curve show smooth learning, and there exists no severe overfitting or underfitting. However, the CNN model performed poorly compared to more intricate architectures in extracting the in-depth defect patterns, which may impact its low accuracy. CNN remains an effective baseline model but lacks power for high-precision defect detection in industrial applications.

EfficientNetB0 gave the best performance among all the models with an accuracy of 98.75% (fig. 9). The accuracy plot converges rapidly with minor oscillations, and the loss plot has low values throughout, which indicates effective learning and great generalization. EfficientNetB0 is better because it uses a compound scaling approach that maintains depth, width, and resolution in equilibrium for improved feature extraction. High model accuracy renders it a viable industrial defect detection candidate with reliability and efficiency

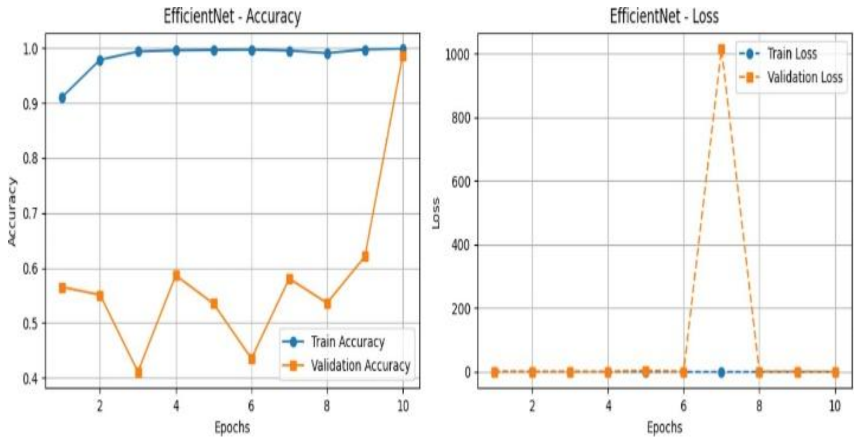

Fig. 9. Accuracy and loss curves for EfficientNetB0.

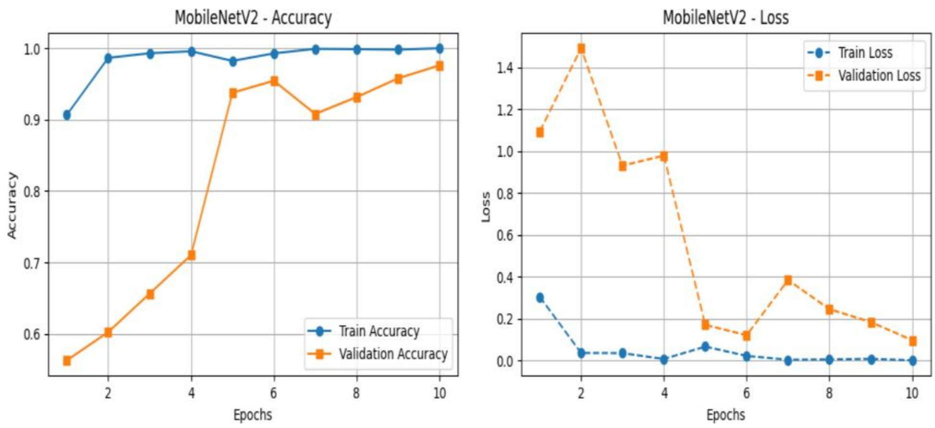

Fig. 10. Accuracy and loss curves for MobileNetV2.

MobileNetV2 model accuracy stood at 98.36%, a close second to EfficientNetB0 (fig. 10). The convergence curves for loss and accuracy were smooth, depicting the model to generalize without the propensity to overfit. Lean MobileNetV2 architecture via depthwise separable convolutions provides efficient computation at high accuracy levels. Therefore, it's an ideal model to use real-time in the manufacturing environment with minimal computational powers.

Confusion Matrix Evaluation: Confusion matrices provide a detailed breakdown of classification outcomes, helping evaluate how well each model distinguishes between defective and non-defective samples.

The MLP-Mixer model showed moderate effectiveness (fig. 11). It correctly identified 799 defective instances but failed to detect 100 others, labeling them as non- defective. Additionally, it accurately classified 375 non-defective items while misclassifying 249 good samples as defective. The model demonstrated a relatively high error rate across both categories, suggesting its token-based architecture may not capture the spatial hierarchies required for precise defect recognition.

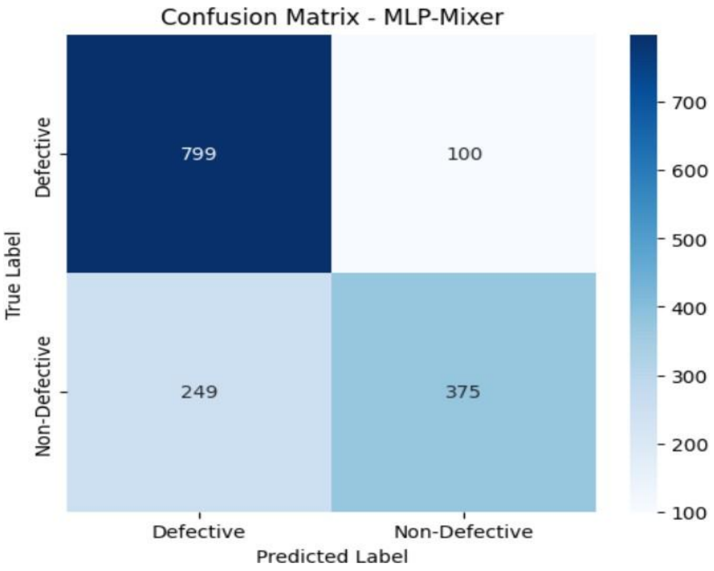

Fig. 11. Confusion Matrix for MLP-Mixer.

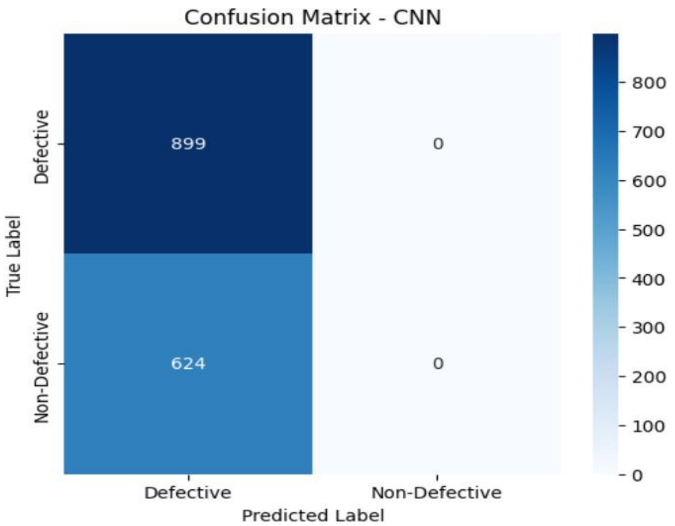

Fig. 12. Confusion Matrix for CNN.

The custom CNN model successfully detected all 899 defective samples (fig. 12). However, it incorrectly labeled all 624 non-defective images as defective, indicating a strong class bias. This resulted in a high false positive rate, which could lead to unnecessary rejections in a real-world production line. While effective in identifying flaws, the model struggled with distinguishing normal items.

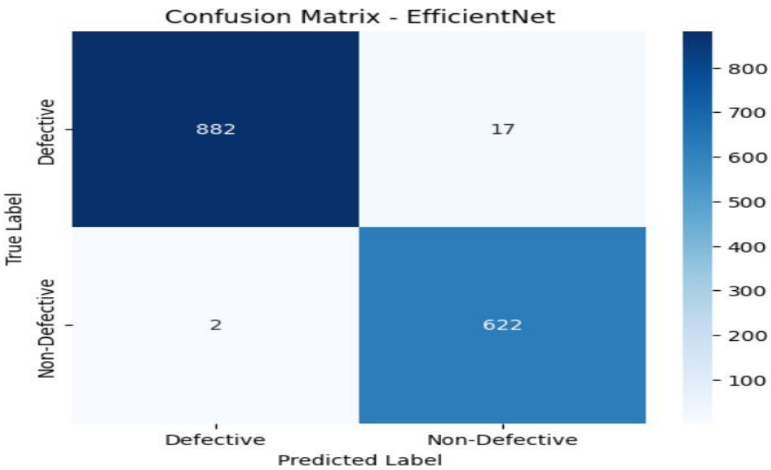

Fig. 13. Confusion Matrix for EfficientNetB0

EfficientNetB0 delivered highly reliable predictions (fig. 13). It accurately classified 882 out of 899 defective cases, with only 17 false negatives. Among the non-defective items, 622 were correctly identified, with just 2 being misclassified. This low error margin showcases the model's strong generalization and its suitability for high-precision inspection systems in manufacturing.

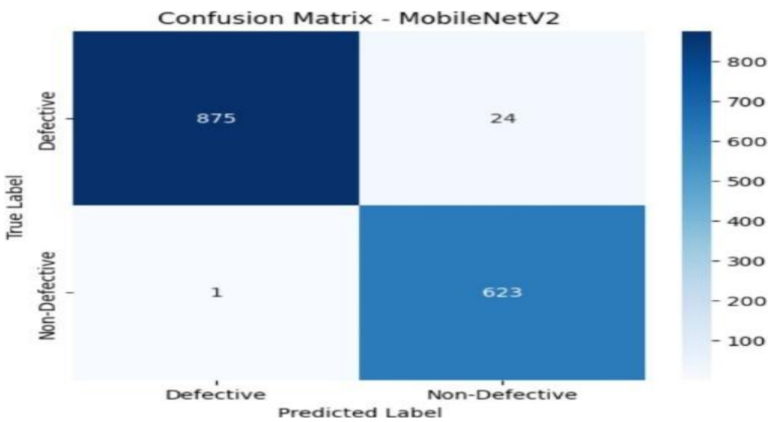

Fig. 14. Confusion Matrix for MobileNetV2.

MobileNetV2 performed nearly on par with EfficientNetB0. It correctly identified 875 defective samples and misclassified 24 as non-defective (fig. 14). For the non-defective class, it accurately labeled 623 out of 624 samples, with just a single false positive. These results indicate MobileNetV2's consistent and efficient classification performance, making it ideal for real time defect detection in environments with limited computational resources.

Model Accuracy Trends: The study reveals that model performance improves with advanced architectures, with models like EfficientNetB0 and MobileNetV2 showing higher accuracy and stability. Less conventional models, like MLP- Mixer, show lower accuracy and performance fluctuations, suggesting their choice is crucial for high accuracy and reliability in automated inspection systems.

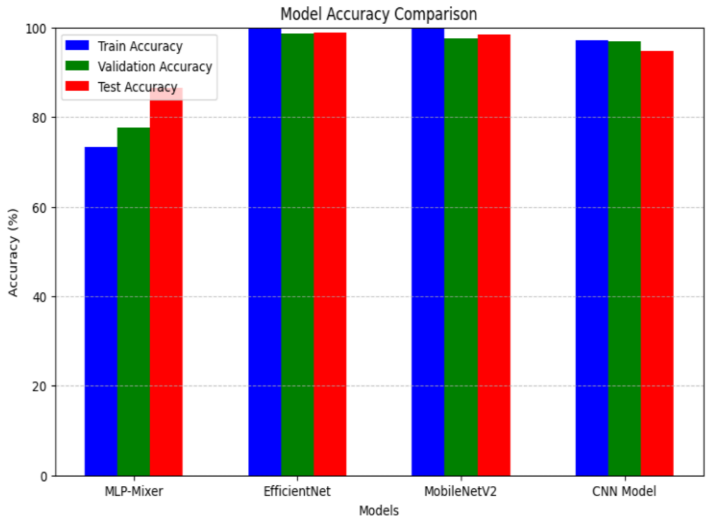

Fig. 15. Model Accuracy Comparison.

The bar chart in fig. 15 shows the comparison of accuracy of four deep learning models: MLP-Mixer, EfficientNetB0, MobileNetV2, and CNN. The accuracies are split into three phases train (blue), validation (green), and test (red). EfficientNetB0 and MobileNetV2 run best with train, validation, and test accuracies approximating 98%. CNN runs well too, with the same accuracy in all phases. The difference between training to validation accuracy of MLP-Mixer is wide, yet test accuracy is on par. The contrast is useful in monitoring model overfitting/underfitting and generalization trends.

Performance Comparison Table: To provide a well-rounded evaluation of the models, four primary classification metrics accuracy, precision, recall, and F1-score were used. These indicators give insight into each model's strengths and weaknesses in detecting casting defects, particularly in distinguishing between defective and non-defective samples. Among all the models tested, EfficientNetB0 achieved the highest accuracy and showed consistent results across all metrics. MobileNetV2 closely followed with comparable performance, making both models highly suitable for deployment in industrial inspection scenarios. On the other hand, the CNN model, although fairly accurate, showed limitations in precision and F1-score, suggesting a tendency to misclassify non-defective parts as defective. The MLP-Mixer model recorded the lowest performance, highlighting challenges in applying token mixing architectures to spatial defect detection tasks.

Table 1. Performance Metrics for Defect Classification Models.

pModel	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
MLP-Mixer	77.08	77.35	77.08	76.41
CNN	94.68	34.84	59.03	43.82
MobileNetV2	98.36	98.41	98.36	98.36
EfficientNetB0	98.75	98.78	98.75	98.75

The results in Table 1 show that advanced, pre-trained architectures such as EfficientNetB0 and MobileNetV2 significantly outperform the simpler CNN and MLP-Mixer models. EfficientNetB0 stands out as the most robust model, with nearly perfect alignment between precision, recall, and F1-score, ensuring reliable predictions in practical applications.MobileNetV2 also demonstrates strong results, especially considering its lightweight design, making it a suitable candidate for environments where computational efficiency is crucial. While the CNN model achieves decent overall accuracy, its low precision highlights a higher false positive rate, which may lead to unnecessary part rejections in a manufacturing setting. Meanwhile, MLP-Mixer's lower metrics suggest that it may require further optimization or be less suited for vision-based defect detection tasks. These findings underline the importance of using efficient, well-tuned architectures in automated quality control systems, where accuracy, speed, and consistency are vital.

5 Conclusion

This paper introduces an automated defect detection system based on deep learning technology that is intended to improve the quality assurance processes in manufacturing settings. The system uses a custom CNN model and transfer learning from EfficientNetB0, MobileNetV2, and MLP-Mixer, with high ac- curacy in classifying defects. EfficientNetB0 outperformed MobileNetV2 and CNN at 98.75% and 94.68%, respectively. The system minimizes reliance on manual labor-intensive verifications, offering an economic, scalable solution for quality control efficiency enhancement. Future improvements will focus on real-time defect detection, optimizing inference speed, and integrating edge AI for on-site analysis. Model optimization techniques like pruning and quantization will be explored for efficiency and cost-effectiveness. The system will also be integrated with IoT manufacturing environments for continuous monitoring and predictive maintenance.

References

[1] Abd Al Rahman, M., & Mousavi, A. (2020). A review and analysis of automatic optical inspection and quality monitoring methods in electronics industry. *IEEE Access*, *8*, 183192–183271. https://ieeexplore.ieee.org/abstract/document/9214824

[2] Aghapanah, H., Rad, A. S., & Rasti, R. (2024). A survey of deep learning in advancing steel industry standards. In *2024 20th CSI International Symposium on Artificial Intelligence and*

Signal Processing (AISP) (pp. 1–7). IEEE. https://ieeexplore.ieee.org/abstract/document/10475283

[3] Azamfirei, V., Psarommatis, F., & Lagrosen, Y. (2023). Application of automation for in-line quality inspection: A zero-defect manufacturing approach. *Journal of Manufacturing Systems, 67*, 1–22. https://www.sciencedirect.com/science/article/pii/S0278612522002291

[4] Chukwunweike, J., Anang, A. N., Adeniran, A. A., & Dike, J. (2024). Enhancing manufacturing efficiency and quality through automation and deep learning: Addressing redundancy, defects, vibration analysis, and material strength optimization. *World Journal of Advanced Research and Reviews, 23*. GSC Online Press. https://wjarr.co.in/sites/default/files/WJARR-2024-2800.pdf

[5] Gaspar, F., Carreira, D., Rodrigues, N., Miragaia, R., Ribeiro, J., Costa, P., & Pereira, A. (2025). Synthetic image generation for effective deep learning model training for ceramic industry applications. *Engineering Applications of Artificial Intelligence, 143*, 110019. https://www.sciencedirect.com/science/article/pii/S0952197625000193

[6] Hütten, N., Alves Gomes, M., Hölken, F., Andricevic, K., Meyes, R., & Meisen, T. (2024). Deep learning for automated visual inspection in manufacturing and maintenance: A survey of open-access papers. *Applied System Innovation, 7*(1), 11. https://www.mdpi.com/2571-5577/7/1/11

[7] Jang, J., Tang, Q., & Jung, H. (2024). PCB defect classification with data augmentation-based ensemble method for sustainable smart manufacturing. *Sustainability, 16*(23), 10417. https://www.mdpi.com/2071-1050/16/23/10417

[8] Jia, Z., Wang, M., & Zhao, S. (2024). A review of deep learning-based approaches for defect detection in smart manufacturing. *Journal of Optics, 53*(2), 1345–1351. https://link.springer.com/article/10.1007/s12596-023-01340-5

[9] Khanam, R., Hussain, M., Hill, R., & Allen, P. (2024). A comprehensive review of convolutional neural networks for defect detection in industrial applications. *IEEE Access*. https://ieeexplore.ieee.org/abstract/document/10589380

[10] Li, X., Li, F., Yang, H., & Wang, P. (2025). Deep learning-enabled visual inspection of gap spacing in high-precision equipment: A comparative study. *Machines, 13*(2), 74. https://doi.org/10.3390/machines13020074

[11] Liang, T., Liu, T., Wang, J., Zhang, J., & Zheng, P. (2024). Causal deep learning for explainable vision-based quality inspection under visual interference. *Journal of Intelligent Manufacturing*, 1–22. https://link.springer.com/article/10.1007/s10845-023-02297-9

[12] Ludwika, A. S., & Rifai, A. P. (2024). Deep learning for detection of proper utilization and adequacy of personal protective equipment in manufacturing teaching laboratories. *Safety, 10*(1), 26. https://www.mdpi.com/2313-576X/10/1/26

[13] Mohandas, R., Southern, M., O'Connell, E., & Hayes, M. (2024). A survey of incremental deep learning for defect detection in manufacturing. *Big Data and Cognitive Computing, 8*(1), 7. https://www.mdpi.com/2504-2289/8/1/7

[14] Papavasileiou, A., Michalos, G., & Makris, S. (2025). Quality control in manufacturing: Review and challenges on robotic applications. *International Journal of Computer Integrated Manufacturing, 38*(1), 79–115. https://www.tandfonline.com/doi/full/10.1080/0951192X.2024.2314789

[15] Zheng, X., Zheng, S., Kong, Y., & Chen, J. (2021). Recent advances in surface defect inspection of industrial products using deep learning techniques. *The International Journal of Advanced Manufacturing Technology, 113*, 35–58. https://link.springer.com/article/10.1007/S00170-021-06592-8