EVOLUTION IN ELECTRICAL AND ELECTRONIC ENGINEERING
e-ISSN: 2756-8458

EEEE

Vol. 6 No. 2 (2025) 8-15
https://publisher.uthm.edu.my/periodicals/index.php/eeee

The Assessment of Deep Learning-based Defect Detection Models in Electronic Components

Muhammad Raihan Mohd Nor Azmi[1], Farhanahani Mahmud[1,2]*

[1] *Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, 86400, MALAYSIA*

[2] *Microelectronics and Nanotechnology Shamsuddin Research Centre, Institute of Integrated Engineering, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, MALAYSIA*

*Corresponding Author: farhanah@uthm.edu.my
DOI: https://doi.org/10.30880/eeee.2025.06.02.002

Article Info

Received: 23 January 2025
Accepted: 30 August 2025
Available online: 30 October 2025

Keywords

ResNet50, EfficientNetB0, InceptionV3, transistor, one-class classification, multi-class classification,

Abstract

The paper presents the implementation of deep learning-based approaches from one-class classification methodologies to identify defects in a transistor in the electronics sector. No labelled data exists, and hence the present work utilizes "good" data to identify bent leads, cut leads, damage areas, and misplacement. The paper is concerned with creating an auto defect detection system capable of increasing effectiveness and precision in quality control at a more reasonable cost. Results indicate the potential to hasten defect detection in industrial practice that employs machine learning. This paper proposes a defect detection system in electronics practice through ResNet50, EfficientNetB0, and InceptionV3 models proposed in Google Colab, TensorFlow—Keras and Python programming language. Defects in the data are of diverse types such as bent lead, cut lead, damaged case, and misplaced. Performance metrics such as accuracy, precision, recall, F1-score, and AUC-ROC are observed. Results indicate that the model of EfficientNetB0 registers the most remarkable overall performance with the highest precision and generalizability, followed by ResNet50 and then by InceptionV3. All three models indicate difficulties in uncovering minority class defects due to the unbalance of the data, and hence balanced sets of data are of vital importance in creating robust auto defect detection systems. Results indicate the capability of balanced sets of data to create robust auto defect detection systems with potential accelerations of quality control in industry practices. Future studies are to rectify the data unbalance through oversampling, undersampling, and data augmentation to make these models more efficient and the auto defect detection system more trustworthy.

1. Introduction

Quality control of electronic components, such as transistors, is of central importance to contemporary manufacturing. Conventional defect detection typically depends upon manual inspection or low-level electrical test, which is laborious, subject to human fallibility, and may not be sensitive to faint defects. Deep learning is a new prospect to overcome this situation and to detect defects in an automatic and accurate way. Anomaly detection, a domain of deep learning, is of particular value to the current application because it is capable of defining departures from normal performance without necessitating generous labeling of data.

The rising complexity of transistor-fabrication procedures has involved more advanced and efficient defect-detection approaches. Although deep learning offers a promising remedy, choosing the most adequate model for a specific application is of primary importance. This work responds to the need to determine the most adequate deep learning model to detect transistor defects in a data-constrained setting, with a specific emphasis on cases involving only a few available labeled data.

The effective application of this work has the potential to make a profound effect on the electronics manufacturing sector. Defect detection can be greatly enhanced by automating it, and overall product quality can be improved. It can decrease the cost of manufacturing substantially. Besides that, valuable experience can be gained from the work of implementing anomaly detection methods in an industrial context, especially for the situation with scarce labelled data. Methodologies generated here can potentially be tailored and transferred to other fields that share a similar problem of defect detection, e.g. semiconductor production and quality control in general industries..

2. Literature Review

This chapter comprehensively reviews the literature on defect detection in electronics manufacturing, focusing on applying deep learning techniques.

2.1 Defect Detection in Electronics Manufacturing

The techniques for sensing defects within electronics are Human Eye Inspection, Machine Vision systems, and X-rays. However, these types of methods can be time-consuming and error-prone due to the involvement of the human operator. Even though some systems, like the Automated Optical Inspection (AOI) system and X-ray inspection system, have been developed to be auto-inspection systems, they have problems like noise sensitivity, many false alarms, and the inability to classify defects. These give better precision, measurement, and reliability compared with the conventional methods of defect identification [1]. These models can therefore find anomalies within high-dimensional spaces, which can be almost impractical through standard procedures.

Electronics manufacturing requires an assessment of products to have a measure through which a manufacturing line can be deemed free from defects. In the past, like in many other similar applications, this task used manual inspection techniques, which might be convenient but slow and prone to errors. With such requirements in the electrical systems, manual inspection is gradually deteriorating as a solution, especially with the considerably increased complexity of modern electronics production lines. Implementations of automated defect detection systems using high-resolution cameras as well as laser scanners are central to today's production lines. The best of these systems is used to scrutinize surface features to note imperfections such as solder joint failures, delamination, and other minute cracks. Nevertheless, the existence of these systems still presents some difficulties owing to the diversity of the defects and the constant changes to the manufacturing processes [2]. A combination of sensor-based data acquisition techniques enables the construction of real-time monitoring and detection of defects.

2.2 Artificial Intelligence (AI) Approaches to Anomaly Detection

AI has proven to be an efficient tool in automating defect detection for electronics. Anomaly detection is the critical component of AI that identifies data points significantly deviating from the norm, as shown in Fig. 1. Such anomalies often indicate a defective or malfunctioning condition of the system under consideration [3]. In AI, two fundamental approaches are used in detecting defects: supervised learning and unsupervised learning. Supervised methods are those that make use of labelled data during training. It classifies instances as either defective or non-defective. Unsupervised methods have normal data available for training. However, no data on the learned set is available for the model. It is a cluster and one-class classification. In unsupervised learning, anomaly detection is a way of finding outliers or abnormal patterns that indicate defects in the system [4].

AI has transformed the detection technique of defects from dealing with vast data, in which no rules are programmed into the system. For instance, methods based on fixed or threshold values cannot provide a suitable solution when confronted with problems in high-dimensional and noisy domains. On the other hand, supervised learning in AI employs categorized data to build models that can accurately classify defective and nondetective goods. Generally, clustering algorithms (for instance, k-Means and DBSCAN) are applicable in anomaly detection since they do not require labelled data. These techniques look for groups within the data or recognize outliers in the data set [5]. For instance, Principal Component Analysis (PCA) has been used in this area by performing dimensionality reduction on the sensor data while identifying defective patterns in real-time processes.

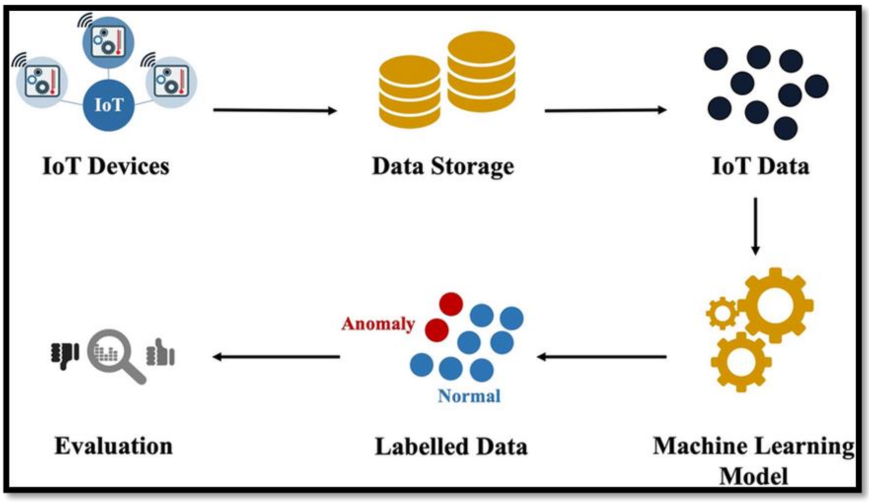


Fig. 1 *AI approaches to anomaly detection [6]*

2.3 ResNet50, EfficientNetB0 & InceptionV3 architectures

A central challenge in training exceedingly deep neural networks is the "vanishing gradient problem," wherein gradients diminish significantly as they propagate through numerous layers. This phenomenon hinders effective weight updates during the training process. ResNet50 addresses this by incorporating "residual blocks." These blocks introduce a unique "shortcut connection" that allows information to bypass specific layers, enabling gradients to flow more directly and efficiently as depicted in Fig. 2. Instead of learning the desired output directly, a residual block learns the residual function – the difference between the input and the desired output. This innovative approach facilitates the learning of complex mappings even with substantially increased network depth. ResNet50 employs a "bottleneck" design within each residual block to enhance computational efficiency. This design incorporates 1x1, 3x3, and 1x1 convolutional layers. The 1x1 convolutional layers act as dimensionality reduction and expansion steps, effectively reducing the number of parameters and computational cost while maintaining the representational capacity of the network.

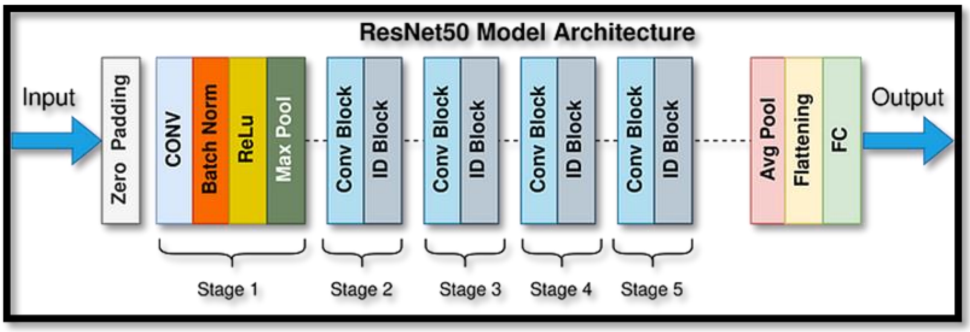


Fig. 2 *ResNet50 model architecture [7]*

EfficientNetB0 offers a new way of scaling neural network architecture. Contrary to conventional approaches that frequently emphasize increasing depth or width individually, EfficientNetB0 utilizes a "compound scaling" procedure that equally scales all network dimensions – width, depth, and resolution – with a single scaling factor as depicted in Fig. 3. This ensures a more balanced and efficient scaling of the network, without a significant increase in computational resources, leading to a remarkable enhancement in performance. Besides, the EfficientNetB0 model applies efficient convolutional operations like "mobile inverted bottleneck convolutions (MBConv)." These convolutional operations make use of depth-wise separable convolution, whose number of parameters and computations is notably low when contrasted with conventional convolutional operations. The efficient convolutional operations and compound scaling make the efficientNetB0 model extremely efficient and efficient, attaining state-of-the-art performance in numerous computer vision recognition problems.

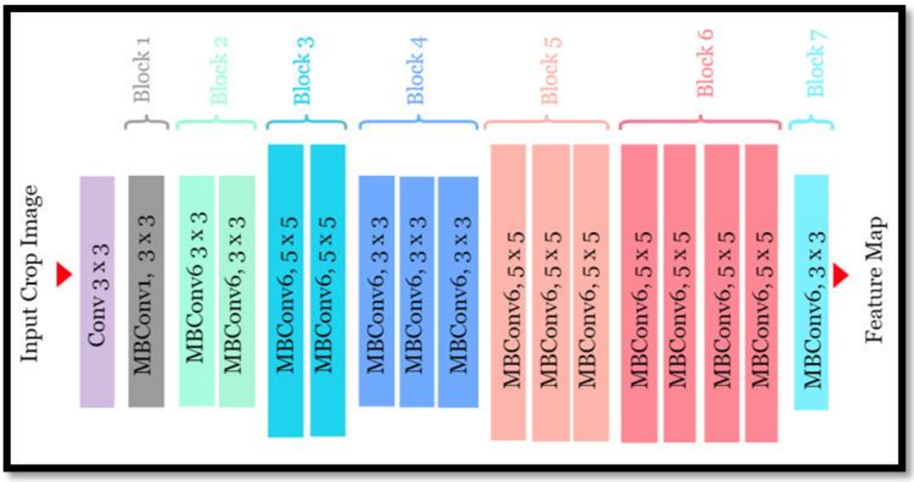


Fig. 3 *EfficientNetB0 architecture [8]*

InceptionV3 introduces the ground-breaking "inception module," a key innovation significantly enhancing feature extraction capabilities. This module combines convolutional filters of varying sizes (1x1, 3x3, 5x5), enabling the network to extract features simultaneously at multiple scales as illustrated in Fig. 4. This multi-scale approach allows the network to capture richer features, significantly improving its ability to recognize complex patterns and objects within images. InceptionV3 strategically utilizes 1x1 convolutional layers to optimize computational efficiency before applying larger convolutions. These 1x1 convolutions act as dimensionality reduction steps, significantly reducing the number of parameters and computational cost. This technique, called "bottleneck layers," enhances computational efficiency while preserving the network's representational power.

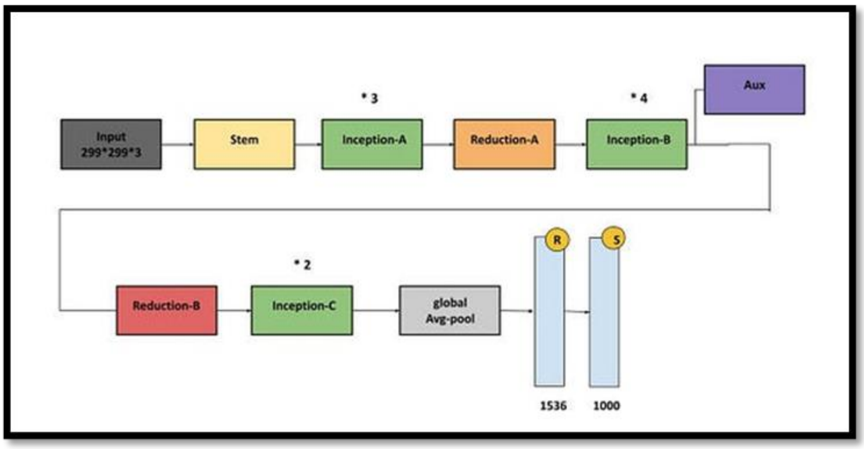


Fig. 4 *InceptionV3 architecture [9]*

3. Methodology

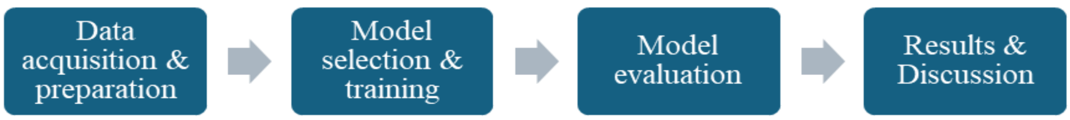


Fig. 5 *Flowchart of project methodology*

Fig. 5 shows the flow chart of the project methodology. This study commenced with the meticulous organization and loading of the image dataset. The dataset, comprising images of transistors with various defects (damaged_case, bent_lead, cut_lead, misplaced), was accessed from MVTec AD [10] and loaded to Google Drive. The train folder consists of 213 good dataset and 40 defect data while the test folder consists of 60 good data and 40 defect data. Subsequently, a comprehensive exploratory data analysis (EDA) phase was conducted. This involved visualizing the dataset through functions that displayed sample images from each defect category, providing a visual understanding of the data distribution. Furthermore, the class distribution within each directory was visualized using bar plots, enabling the identification of potential class imbalances. To prepare the dataset for subsequent model training, a function was implemented to create a Pandas DataFrame that associates each image filename with its corresponding defect label. This labelled dataset was the foundation for the subsequent model development and evaluation stages.

After the exploratory analysis and data preparation, the focus turned to model evaluation. For this investigation, three pre-trained deep learning models were chosen: ResNet50, EfficientNetB0, and InceptionV3. The defect classification system was built on top of these models, which are well known for their outstanding performance on image classification tasks. The image dataset was pre-processed to guarantee compatibility with these models. In order to achieve the best performance with the selected architecture, all images had to be resized to a standard resolution of 224x224 pixels. A training set and a validation set were the two subsets created from the pre-processed dataset. The models were trained using the training set, which made up 80% of the data, with the remaining 20% set aside for validation. In order to keep an eye on the model's performance during training and avoid overfitting—a situation in which the model performs well on training data but poorly on unseen data—the validation set acted as an independent dataset. The training dataset was then used to train each of the three pre-trained models. The pre-trained weights of these models were adjusted for the particular task of transistor defect classification during the training process. By utilizing the knowledge that the models have already gained from their initial training on the ImageNet dataset, this method—known as transfer learning—significantly speeds up the training process and enhances performance on the intended task.

A wide range of classification performance metrics, including accuracy, precision, recall, F1-score, and AUC-ROC, were used to thoroughly assess each trained model's performance. The overall correctness of the model's predictions is represented by accuracy. It is determined by dividing the number of correctly predicted cases by the total number of cases. Although it may seem simple, accuracy can be deceptive in datasets that are unbalanced, meaning that one class is substantially more prevalent than others.

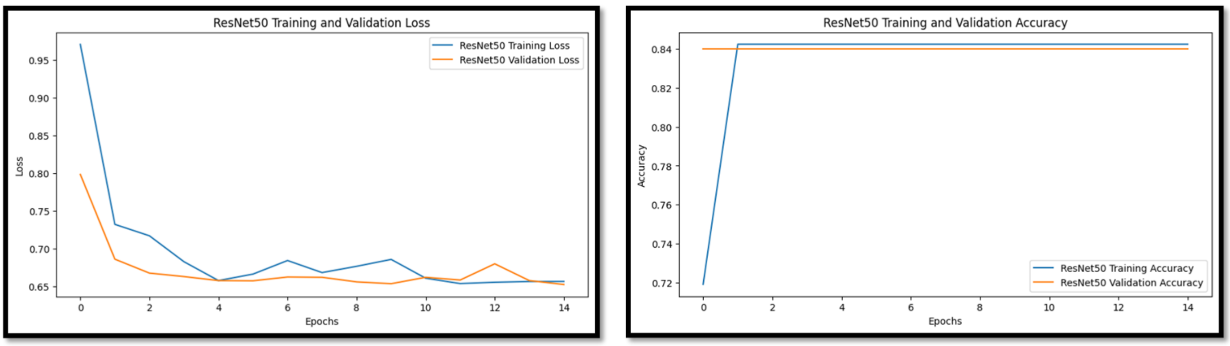
The evaluation's findings were then carefully examined and contrasted. A comparative bar chart was used to display each model's performance, including average training and validation accuracies during the training, AUC-ROC scores, and the validation accuracies from the confusion matrix. A clear and succinct comprehension of the relative performance of each model was made possible by this visual representation.

4. Results and Discussion

This chapter expands upon research conclusions using deep learning algorithms to identify transistors' defects. Selected models have been InceptionV3, ResNet50, and EfficientNetB0

4.1 Results

Fig. 6 illustrates the training and validation progress of the ResNet50 model over 15 epochs. The training accuracy steadily increases from around 49% to 84.39%, indicating successful learning by the model. While the validation accuracy also improves, it remains relatively stable around 84%, suggesting that the model is generalizing well to unseen data and is not overfitting significantly.



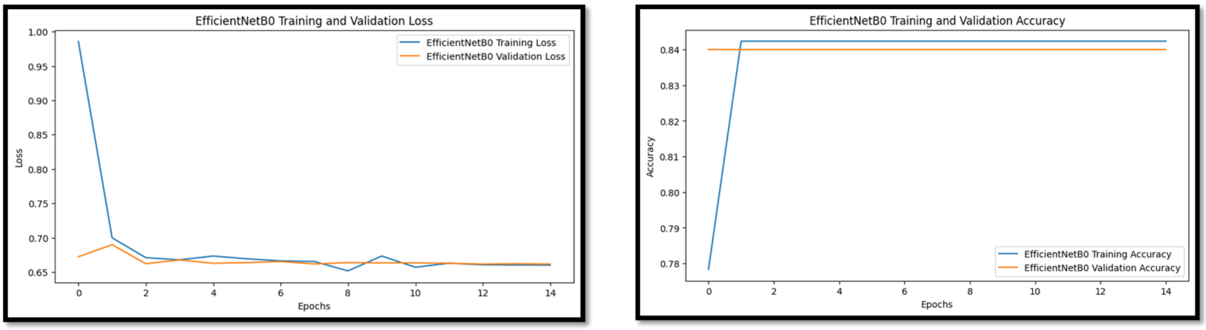
(a) (b)

Fig. 6 *(a) ResNet50 training and validation loss graph; (b) ResNet50 training and validation accuracy*

Fig. 7 displays the training and validation progress of the EfficientNetB0 model over 15 epochs. The training accuracy steadily increases from around 62% to 89.26%, indicating successful learning. The validation accuracy

Evolution in Electrical and Electronic Engineering Vol. 6 No. 2 (2025) p. 8-15

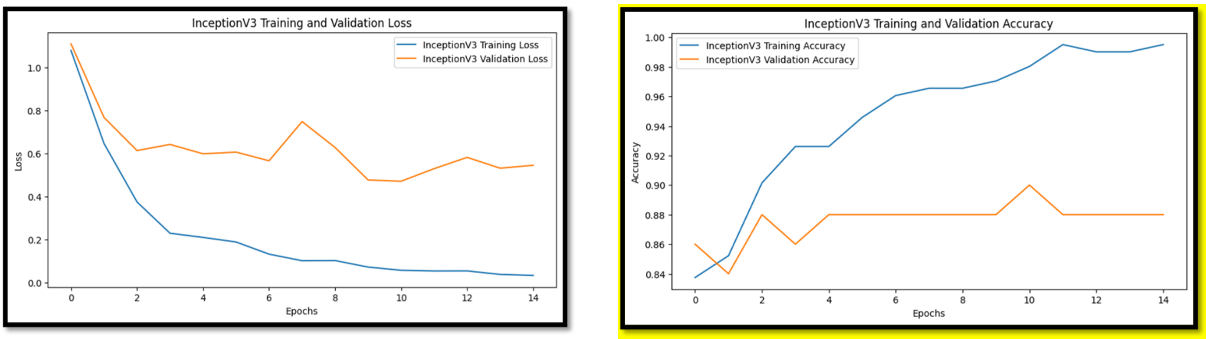
remains consistently high at 84%, suggesting good generalization to unseen data. While the model shows promising performance, a slightly larger gap between training and validation accuracy compared to ResNet50 might indicate a potential for minor overfitting, which could be addressed through further optimization.



(a) (b)

Fig. 7 *(a) EfficientNetB0 training and validation loss graph; (b) EfficientNetB0 training and validation accuracy*

Fig. 8 presents the training and validation progress of the InceptionV3 model over 15 epochs. The training accuracy increases from around 80% to 99.89%, while the validation accuracy remains consistently high at 84%. This rapid convergence of training accuracy suggests that the model may be overfitting to the training data. Further analysis and potential regularization techniques might be necessary to improve the model's generalization ability.



(a) (b)

Fig. 8 *(a) InceptionV3 training and validation loss graph; (b) InceptionV3 training and validation accuracy*

```
Evaluating ResNet50...
4/4 ━━━━━━━━━━ 12s 3s/step
ResNet50 Classification Report:
              precision    recall  f1-score   support

   bent_lead       0.00      0.00      0.00         2
    cut_lead       0.00      0.00      0.00         2
damaged_case       0.00      0.00      0.00         2
        good       0.84      1.00      0.91        42
   misplaced       0.00      0.00      0.00         2

    accuracy                           0.84        50
   macro avg       0.17      0.20      0.18        50
weighted avg       0.71      0.84      0.77        50
```

(a)

```
Evaluating EfficientNetB0...
4/4 ━━━━━━━━━━ 12s 2s/step
EfficientNetB0 Classification Report:
              precision    recall  f1-score   support

   bent_lead       0.00      0.00      0.00         2
    cut_lead       0.00      0.00      0.00         2
damaged_case       0.00      0.00      0.00         2
        good       0.84      1.00      0.91        42
   misplaced       0.00      0.00      0.00         2

    accuracy                           0.84        50
   macro avg       0.17      0.20      0.18        50
weighted avg       0.71      0.84      0.77        50
```

(b)

```
Evaluating InceptionV3...
4/4 ━━━━━━━━━━ 9s 1s/step
InceptionV3 Classification Report:
              precision    recall  f1-score   support

   bent_lead       0.00      0.00      0.00         2
    cut_lead       0.00      0.00      0.00         2
damaged_case       0.00      0.00      0.00         2
        good       0.83      0.95      0.89        42
   misplaced       0.00      0.00      0.00         2

    accuracy                           0.80        50
   macro avg       0.17      0.19      0.18        50
weighted avg       0.70      0.80      0.75        50
```

(c)

Fig. 9 *(a) ResNet50 classification report; (b) EfficientNetB0 classification report; (c) InceptionV3 classification report*

The evaluation of ResNet50, EfficientNetB0, and InceptionV3 models on a test dataset revealed significant performance disparities across different defect classes, as depicted in Fig. 9. All three models successfully identified "good" components, achieving high recall and F1-scores. However, their performance on other defect classes, such as "bent_lead," "cut_lead," and "misplaced," was abysmal, with zero precision, recall, and F1-scores. This suggests that class imbalance within the training dataset may contribute to the models' difficulty identifying these minority classes. Overall, the models exhibited moderate accuracy and low macro-averaged performance, indicating a need for further improvement to classify all defect types effectively.

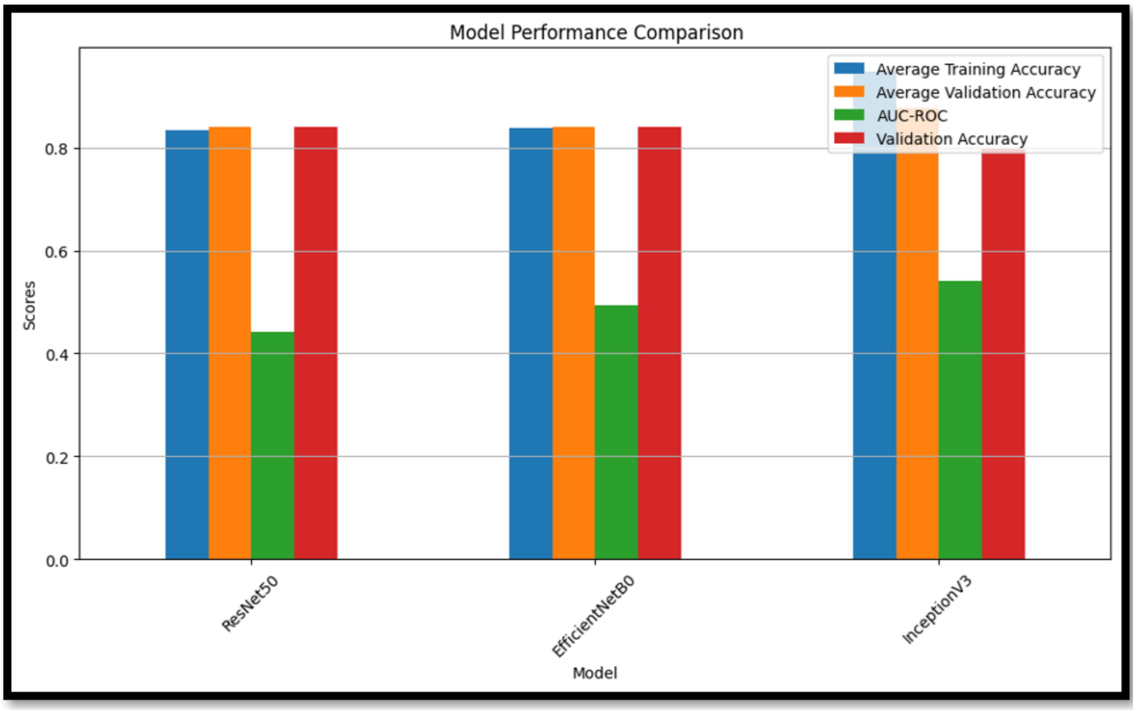


Fig. 10 *Model performance comparison*

Fig. 10 shows the bar chart of the average training accuracy and validation accuracy, the AUC-ROC, and the validation accuracy from the test dataset of ResNet50, EfficientNetB0, and InceptionV3. The comparison of model performances outlines different advantages and disadvantages. Namely, improving generalization is crucial since, although InceptionV3 provides training accuracy equal to 94.68%, the validation accuracy and other results for minority defect classes are still far from optimal due to overfitting. ResNet50 is balanced between the training accuracy and validation accuracy and shows reasonable indications of generalization and also EfficientNetB0 shows the same results with slightly better training accuracy of 83.81%.

4.2 Discussion

Overall, the ResNet50 and EfficientNetB0 models demonstrated promising performance with good generalization to unseen data. ResNet50 exhibited a steady increase in training and validation accuracy, suggesting a well-balanced performance. EfficientNetB0 also showed strong performance, although a slight gap between training and validation accuracy might indicate a minor risk of overfitting. Conversely, InceptionV3's validation accuracy stalled while training accuracy remained high, suggesting overfitting. Moreover, the dataset is imbalanced—abnormal/defect samples are underrepresented, which can inflate aggregate accuracy while masking degraded sensitivity on rare failure cases.

5. Conclusion

This study aimed to investigate the performance of deep learning models (ResNet50, EfficientNetB0, and InceptionV3) for detecting defects in transistors. This has provided an insight into the real identification of defects within the production process to help the manufacturers make necessary changes immediately and minimize waste. On edge devices, deployment may simplify the models for low-latency applications, such as in-line real-time defect detection on a production line [11]. While the models demonstrated high accuracy in identifying non-defective components, their performance significantly degraded when classifying minority defect classes such as "bent_lead" and "cut_lead." This limitation can be attributed to class imbalance within the dataset, where most data points belonged to the "good" class, leading to inherent bias towards this category. Consequently, the models struggled to learn and classify the less frequent defect classes effectively. These findings highlight the critical importance of addressing class imbalance in future research to improve the accuracy and reliability of deep learning models for transistor defect detection. The findings indicate that extensive work needs to be carried out on the quality of datasets, the choice of model structures, and the nature of the data distribution of defect detection systems suited to electronics industries. The use of a more diverse variety of defects and diversification in the data sources must strengthen the generalizability of the models.

Acknowledgement

This author would like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia and the people who supported this research directly and indirectly throughout its completion.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Muhammad Raihan Mohd Nor Azmi, Farhanahani Mahmud;* ***data collection:*** *Muhammad Raihan Mohd Nor Azmi;* ***analysis and interpretation of results:*** *Muhammad Raihan Mohd Nor Azmi, Farhanahani Mahmud;* ***draft manuscript preparation:*** *Muhammad Raihan Mohd Nor Azmi. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Fan, S.-K. S., Hsu, C.-Y., Jen, C.-H., Chen, K.-L., & Juan, L.-T. (2020). Defective wafer detection using a denoising autoencoder for semiconductor manufacturing processes. Advanced Engineering Informatics, 46, 101166. https://doi.org/10.1016/j.aei.2020.101166

[2] Zou, Y., Jeong, J., Latha Pemula, Zhang, D., & Onkar Dabeer. (2022). Spot-the-Difference Self-supervised Pre-training for Anomaly Detection and Segmentation. Lecture Notes in Computer Science, 392–408. https://doi.org/10.1007/978-3-031-20056-4_23

[3] Abu, M., Amir, A., Lean, Y. H., Zahri, N. A. H., & Azemi, S. A. (2021). The Performance Analysis of Transfer Learning for Steel Defect Detection by Using Deep Learning. Journal of Physics: Conference Series, 1755(1), 012041. https://doi.org/10.1088/1742-6596/1755/1/012041

[4] Wu, Y., Cao, H., Yang, G., Lu, T., & Wan, S. (2022). Digital Twin of Intelligent Small Surface Defect Detection with Cyber-Manufacturing Systems. ACM Transactions on Internet Technology. https://doi.org/10.1145/3571734

[5] Davtalab, O., Kazemian, A., Yuan, X., & Khoshnevis, B. (2020). Automated inspection in robotic additive manufacturing using deep learning for layer deformation detection. Journal of Intelligent Manufacturing. https://doi.org/10.1007/s10845-020-01684-w

[6] Niu, M., Song, K., Huang, L., wang, qi, Yan, Y., & Meng, Q. (2020). Unsupervised Saliency Detection of Rail Surface Defects using Stereoscopic Images. IEEE Transactions on Industrial Informatics, 1–1. https://doi.org/10.1109/tii.2020.3004397

[7] S. Mukherjee, "The Annotated ResNet-50," Medium, Aug. 18, 2022. https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758

[8] C. H. Patel, Devang Undaviya, H. Dave, Sheshang Degadwala, and D. Vyas, "EfficientNetB0 for Brain Stroke Classification on Computed Tomography Scan," May 2023, doi: https://doi.org/10.1109/icaaic56838.2023.10141195.

[9] A. BRITAL, "Inception V3 CNN Architecture Explained," Medium, Oct. 24, 2021. https://medium.com/@AnasBrital98/inception-v3-cnn-architecture-explained-691cfb7bba08

[10] Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., & Steger, C. (2021). The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. International Journal of Computer Vision. https://doi.org/10.1007/s11263-020-01400-4

[11] Psarommatis, F. (2021). A generic methodology and a digital twin for zero defect manufacturing (ZDM) performance mapping towards design for ZDM. Journal of Manufacturing Systems, 59, 507–521. https://doi.org/10.1016/j.jmsy.2021.03.021