International Journal of Engineering Trends and Technology Volume 72 Issue 10, 235-245, October 2024
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I10P123

Original Article

Performance Analysis of Deep Learning Models for Classification of Surface-Mount Device (SMD) Components

Pratik J Gohel[1], Brijesh R Solanki[2], Amit C Rathod[1], Devang G Jani[3]

[1]*Department of EC Engineering, Government Engineering College, Bhavnagar, Gujarat, India.*
[2]*Department Electrical Engineering, Government Engineering College, Bhavnagar, Gujarat, India.*
[3]*Department EC Engineering, Shantilal Shan Engineering College, Bhavnagar, Gujarat, India.*

[1]*Corresponding Author : pratik.gohel09@gmail.com*

Received: 30 May 2024 Revised: 07 October 2024 Accepted: 14 October 2024 Published: 25 October 2024

Abstract *- Accurate classification of Surface-Mount Device (SMD) components is important for a range of electronics manufacturing and assembly applications. Impressive results have been achieved using recent deep learning models when applied to image-classification problems. This study provides a comprehensive examination of the categorization of Surface-Mounted Device (SMD) components using advanced learning models. Four cutting-edge Deep Learning (DL) models–ResNet50, VGG16, AlexNet, and MobileNet–were utilized to categorize SMD components into eight classes: capacitors, diodes, Electrolytic Capacitors (EC), Integrated Circuits (IC), LED, resistors, supercapacitors, and Zener diodes. Our approach encompasses the training of these models on a dataset containing SMD component images and the assessment of their performance in terms of accuracy, precision, recall, and F1-score. The findings indicate that MobileNet achieved the highest classification accuracy, reaching up to 98%, surpassing the other models. Through a comprehensive comparative analysis, we discern the strengths and limitations of each model in this categorization task. Our results suggest that MobileNet is the most effective deep-learning framework for SMD component classification, underscoring its potential applications in automated electronic assembly and quality control processes. This study contributes to the progress of automated electronic component classification and guides future research in selecting suitable deep learning models for similar tasks.*

Keywords *- Object detection, SMD components, Machine learning, VGG16, ResNet50, Alexnet, MobileNet.*

1. Introduction

Electronic components known as Surface-Mount Devices (SMDs) are soldered directly onto a Printed Circuit Board (PCB) surface. These parts are frequently used in industrial equipment, medical gadgets, automobile systems, and consumer electronics. The accurate identification and categorization of SMD components are crucial for automated assembly operations, inventory management, and quality control. Surface-Mount Device (SMD) components are essential for the production and assembly of contemporary electronics. These tiny electrical components include integrated circuits, diodes, capacitors, and resistors [1] [2]. In recent years, deep learning algorithms have transformed picture-categorization problems. Convolutional Neural Networks (CNNs) have shown impressive results in several fields, including computer vision [3]. The distinctive design ideas and capabilities of VGG16, ResNet50, AlexNet, and MobileNet distinguish them from other well-known CNN systems. In the context of SMD component categorization, this study attempted to compare and assess the efficacy of these four CNNs. We aimed to obtain insights into choosing the best model for practical applications by examining their tradeoffs, computational efficiency, and performance indicators [4]. The use of VGG16 exemplifies the precision and robustness of feature extraction in SMD component identification [5]. Although it requires significant computational resources, its consistent performance renders it an excellent option for handling tasks that require meticulous inspection. ResNet50 is expected to utilize its residual learning technique to retain its effectiveness even with many layers, resulting in improved accuracy in detecting SMD components. The model is perfect for AOI systems because of its robustness and consistent performance across various SMD components and situations [6]. However, the more profound architecture of ResNet50 may require substantial computational resources [7]. The eight-layer AlexNet architecture uses data augmentation techniques, dropout for regularization, ReLU activations, and three completely interconnected layers after five convolutional layers [8]. AlexNet is an excellent option for image classification and detection applications because of its popularity and efficiency. Depth-wise, separable convolutions are used by MobileNet, which significantly reduces the number of parameters and the processing burden. Consequently, MobileNet is particularly well-suited for real-time applications that operate on devices with limited processing capabilities [9]. The MobileNet architecture contains depth-wise convolution supported by pointwise convolution [10, 11].

2. Deep Learning Models
2.1. VGG16
The Visual Geometry Group (VGG) developed a deep Convolutional Neural Network (CNN) model known as VGG16. It is renowned for being easy to use and efficient and performs well on various computer vision tasks such as object detection and image classification. Outline of the VGG16 architecture at a high level.

1. Input: An RGB-channeled picture with a fixed size of 224 × 224 pixels is the input of the VGG16 model.
2. Convolutional Layers: There are thirteen convolutional layers in the model. These layers employ tiny (3 × 3) convolution filters, which enable the model to capture more intricate details.
3. Max-Pooling Layers: These layers come after each set of convolutional layers and reduce the dimensionality of the feature maps while keeping the most crucial information.
4. Fully Connected Layers: These layers come after the convolutional and max-pooling layers and consist of three fully connected layers. The first two have 4096 channels, but the third has 1000 channels (one for each class) because it uses a 1000-way ILSVRC classification.
5. Softmax Layer: A softmax layer is the last layer.

2.2. ResNet50
Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian presented ResNet, a kind of Convolutional Neural Network (CNN) 2015. ResNet50 is a variation of ResNet. The network's 48 convolutional layers, one MaxPool layer, and one average pool layer are represented by the "50" in ResNet50. Outline of ResNet50 architecture:

1. Input: A picture with RGB channels and a fixed size of 224 by 224 pixels serves as the input for the ResNet50 model.
2. Convolutional Layers: There Several convolutional layers exist in this model. These layers employ tiny (3x3) convolution filters, which enable the model to pick up more intricate details.
3. Shortcut Connections: The utilization of shortcut connections, often referred to as skip or residual connections, is a major ResNet invention. These connections avoid the vanishing gradient issue, which is prevalent in deep neural networks, and enable the network to bypass one or more layers.
4. Bottleneck Design: The building block of ResNet50 has a bottleneck design that minimizes the number of parameters and matrix multiplications and allows for considerably quicker training of each layer.
5. Fully Connected Layers: Fully connected layers come after the max-pooling and convolutional layers. The classification was performed using these layers.
6. Softmax Layer: The last layer, called the softmax layer, produces a probability distribution over the classes.

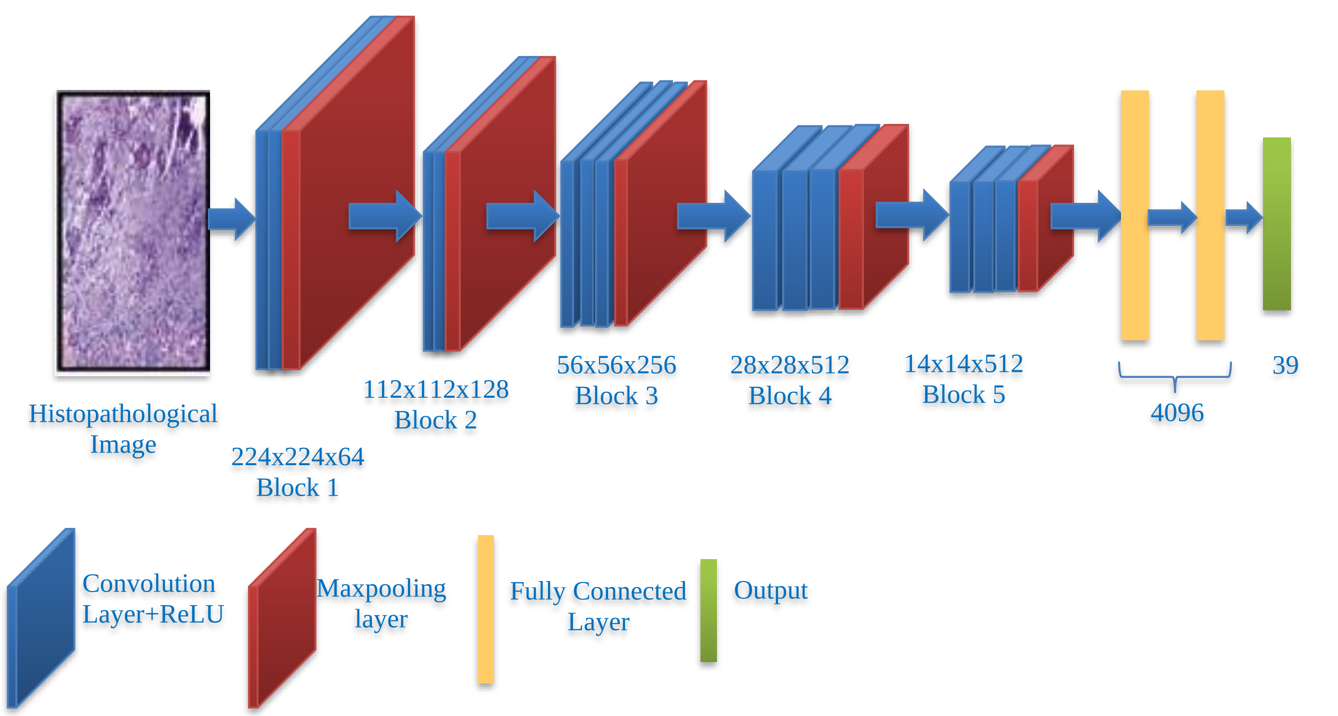


Fig. 1 VGG16 architecture [12]

The ResNet50 architecture was trained using the ImageNet dataset, comprising 14 million images categorized into 1000 classes. Although intricate, ResNet50 is widely used in numerous deep learning applications owing to its adaptability and exceptional efficiency.

2.3. AlexNet

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton created the Convolutional Neural Network (CNN) architecture called AlexNet. A high-level overview of the AlexNet architecture.

1. Input: An image with RGB channels measuring 227 × 227 pixels is fed into the AlexNet model.
2. Convolutional Layers: The model has five convolutional layers, some of which are followed by max-pooling layers.
3. Fully Connected Layers: The final three levels are connected entirely.
4. ReLU Activation: A convolution filter and a nonlinear activation function known as" ReLU" comprise each convolution layer.
5. Dropout: The architecture's initial two fully linked layers employed a dropout of 0.5 to lessen overfitting.
6. Softmax Layer: This layer produces a probability distribution across the classes as its output.

AlexNet's notable characteristic is the utilization of a GPU to enhance the training performance. This architectural design pioneered the training of 60 million parameters, rendering it susceptible to overfitting.

Nevertheless, the incorporation of Dropout and Data Augmentation plays a vital role in mitigating overfitting.

Adopting the ReLU activation function instead of the tanh or sigmoid function led to accelerated training durations.

In contrast to other activation functions that tend to saturate at higher activation values, Deep Learning Networks commonly integrate ReLU nonlinearity to expedite the training.

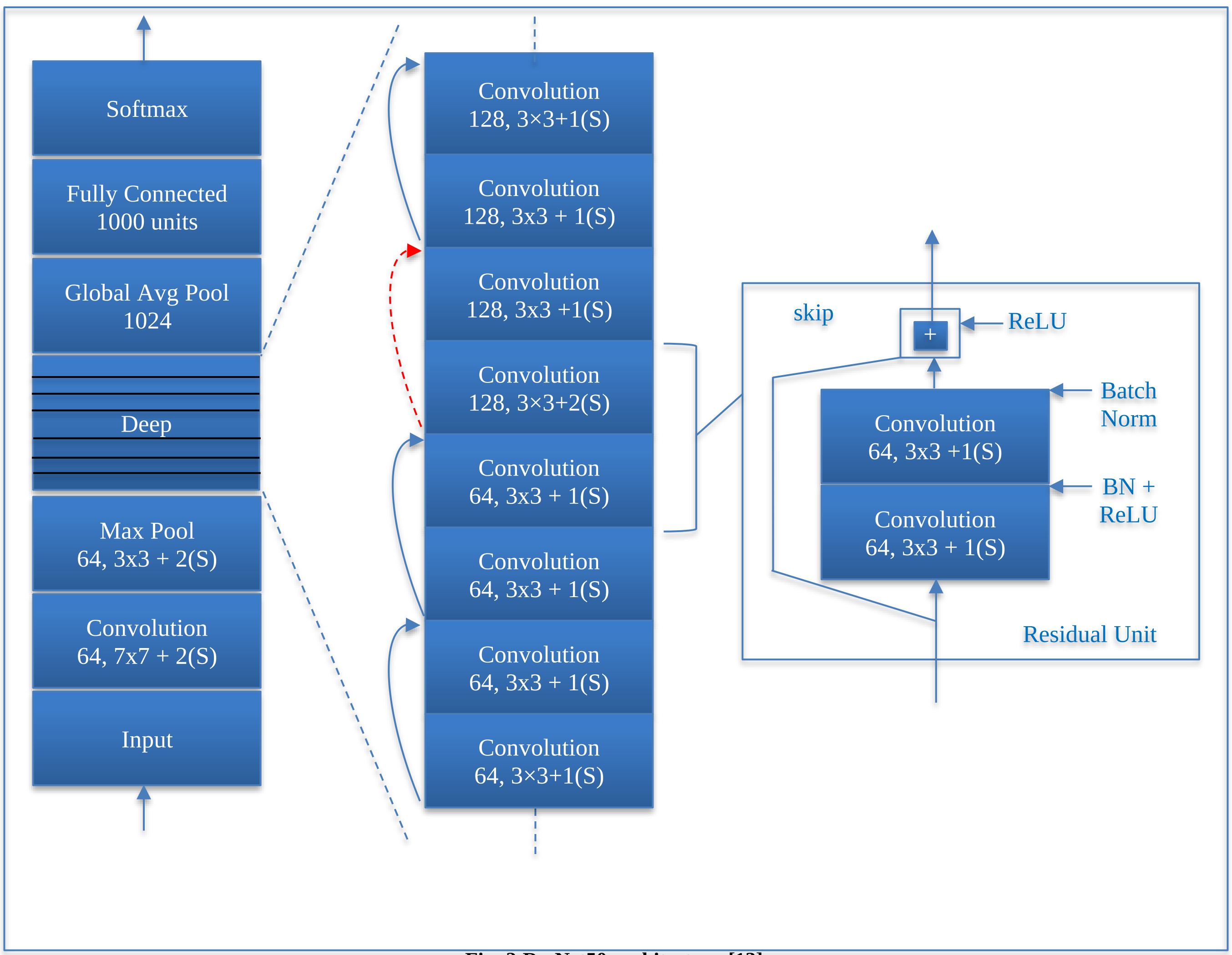


Fig. 2 ResNet50 architecture [13]

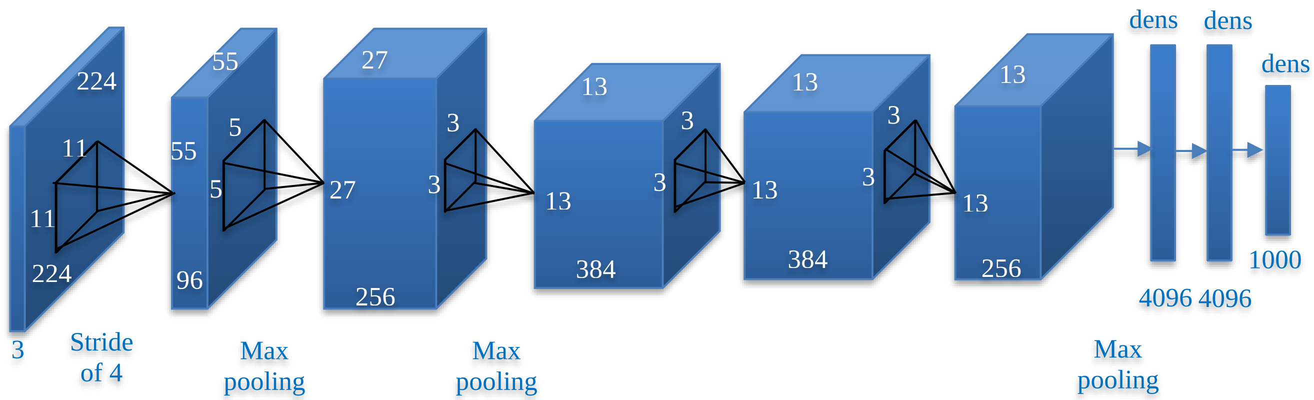


Fig. 3 AlexNet architecture [14]

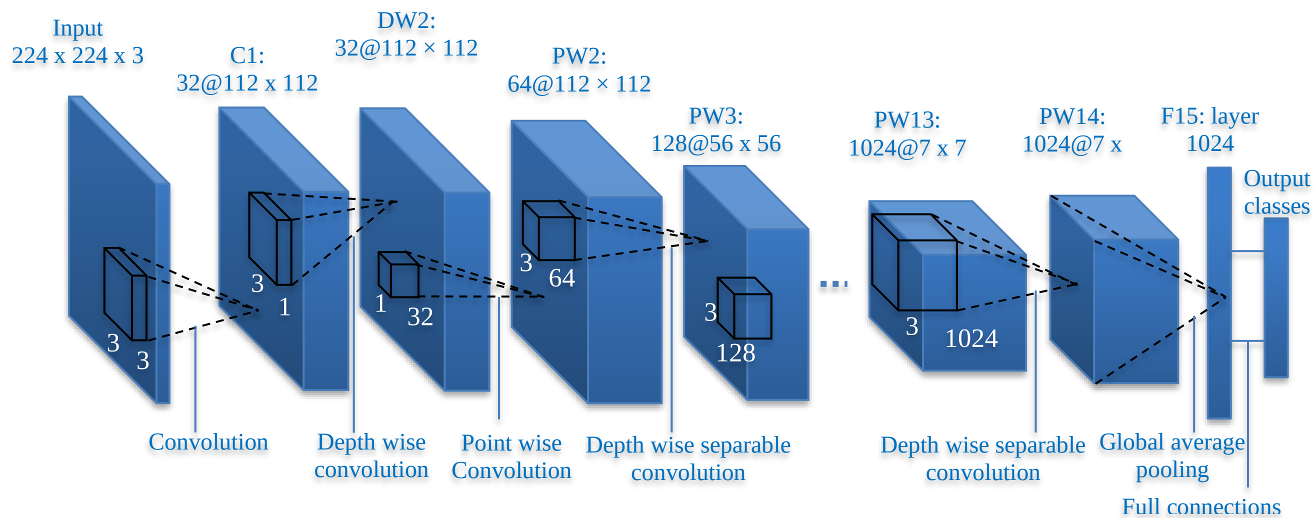


Fig. 4 MobileNet architecture [15]

2.4. MobileNet

Convolutional Neural Networks (CNNs) of the MobileNet type are intended for embedded and mobile vision applications. Its efficiency and open-source nature make it a good fit for mobile device applications. The architecture of MobileNet is summarized as follows:

1. Input: MobileNet uses an image as input.
2. Depthwise Separable Convolutions: MobileNet employs depthwise separable convolutions, a type of factorized convolution that significantly reduces computational costs. This approach involves applying a single filter per input channel (depth-wise convolution), followed by a 1 × 1 convolution.
3. Convolutional Layers: The model comprises multiple convolutional layers, some of which are followed by nonlinear activation functions.
4. Fully Connected Layers: The final layer is the fully connected layer that produces the ultimate layer.

One of the primary features of MobileNet is its efficiency because it utilizes fewer parameters than other networks, resulting in faster and less computationally demanding operations. Consequently, MobileNet is particularly appropriate for mobile applications with limited computational resources. Despite its simplicity and efficiency, MobileNet performs well across various image-classification tasks. It is an excellent foundation for compact and swift training classifiers and is particularly suitable for mobile and embedded applications.

3. Methodology

3.1. Dataset

The initial step of the approach involved preparing and loading the dataset, which comprises over 16600 images. The division of the dataset by category is shown in Table 1. The dataset was partitioned randomly in an 80:20 ratio, with 80% of the images allocated for training and the remaining 20% allocated for testing and validation, as outlined in Table 1.

The experiment encompassed diverse types, sizes, orientations, and constructions of discrete and surface-mounted electronic components.

3.2. Image Preprocessing

Before training the deep learning models, the vital stage was image preprocessing. This process typically involves several essential steps: Initially, the images were adjusted to a standard size to ensure consistency across datasets. Subsequently, normalization occurs, where pixel values are adjusted to a common range, typically 0 to 1 or -1 to 1, facilitating faster convergence during training. The color space can be altered by converting RGB images to grayscale images when color information is not essential.

To artificially expand the dataset and enhance model generalization, data augmentation techniques, such as image rotation, flipping, or noise addition, can be utilized. Image quality can be improved through noise-reduction filters or contrast enhancement. Finally, the images were converted into suitable numerical formats, such as tensors, for input into the deep learning model. These preprocessing steps serve to standardize the input data, reduce computational demands, and improve the capacity of the model to extract relevant features from the images.

3.3. Implementation Tools

Python, TensorFlow, and Keras were used as the implementation tools. These implementation tools enable users to incorporate pre-trained deep learning models such as VGG16, ResNet50, Alexnet, and MobileNet.

- *Python* is commonly used for implementing these models because of its simplicity and the wide range of available scientific and numerical libraries, such as NumPy and SciPy.
- *TensorFlow*, an open-source library created by Google, was designed for numerical computations and large-scale machine learning. It integrates numerous machine learning and deep learning models and algorithms and utilizes Python to offer a user-friendly front-end API.
- *Keras*, a high-level neural network API, is built in Python and can operate on top of TensorFlow. It prioritizes enabling swift experimentation, aiming to minimize the time it takes to move from concept to outcome, which is crucial for conducting effective research.

Table 1. Division of image data set

Components	Train	Test	Total
Capacitor	3600	900	4500
Diode	1000	250	1250
EC	2000	500	2500
IC	2080	520	2600
LED	1400	350	1750
Resistor	2000	500	2500
SCapacitor	280	70	350
Zener	920	230	1150
Total	13280	3320	16600

3.4. Data Augmentation and Preprocessing

Image data augmentation is a method aimed at artificially expanding the size of a training dataset by generating modified versions of images.

This regularization technique mitigates overfitting and enhances the generalization ability of the model [16].

The following specific augmentation steps are involved:

- Rotation: The images can be rotated by a random angle within the range of 0 °to 360 °, aiding the learning model in recognizing objects in diverse orientations.
- Random Horizontal Shift: Introducing a random horizontal shift of 20% allows the model to learn object recognition in distinct positions within the image, which is particularly beneficial for datasets in which the object is consistently centered.
- Random Vertical Shift: Analogous to horizontal shift, a random vertical shift of 20% relocates the image vertically, supporting the model in learning object recognition by varying vertical positions within the image.
- Random Zoom: Randomized zooming into the image by 20% contributes to the model's ability to learn object recognition at different scales, which is advantageous for datasets with varying object sizes.
- Horizontal and Vertical Flip: Applying horizontal or vertical flips assists the model in recognizing objects in varying orientations, which is particularly advantageous for datasets with diverse object orientations.

Implementing these augmentations significantly enhances the training data diversity, thereby improving the model performance.

3.5. Implementation Steps

To utilize VGG16, ResNet50, AlexNet, and MobileNet, the following steps are pursued:

1) Importing Essential Libraries: This includes TensorFlow, Keras, and other relevant Python libraries.
2) Loading Pre-Trained Model: Retrieving the pre-trained model architecture from Keras, potentially VGG16, ResNet50, AlexNet, or MobileNet.
3) Preprocessing the Input: Aligning the input data with the anticipated format of the model, which may involve image resizing and normalization of pixel values.
4) Compiling the Model: Specifying the optimizer and loss function for model training.
5) Training the Model: Feeding the input data to the model and adjusting the model weights based on the computed loss during training on the dataset.
6) Assessing the Model: Evaluating the model's performance on a validation or test dataset.
7) Making Predictions: Utilizing the trained model to predict outcomes for new data.

3.6. Training the Network

The initial learning rate was established at 0.002 to decelerate the learning process within the transferred layers that were not fixed. Conversely, the learning rate multiplier for the fully connected layer was augmented to expedite the learning within the new final layers of the network. This configuration of learning rate adjustments reduces the training duration. Consequently, 50 epochs were used to train all the models.

4. Results and Discussion

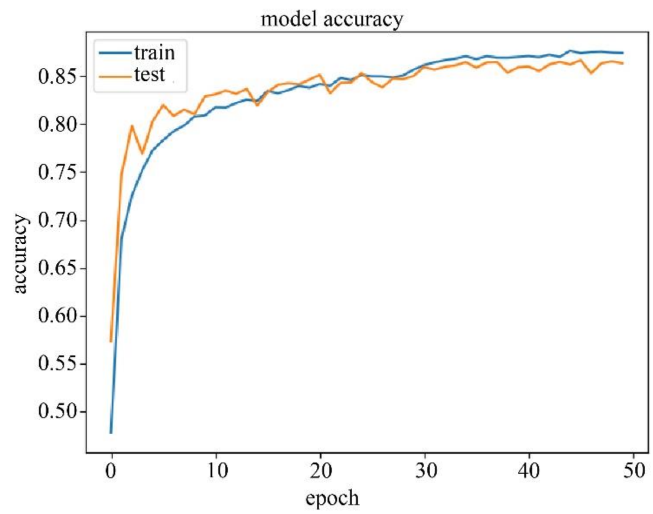
A comparative study of image classification algorithms such as VGG16, ResNet50, AlexNet, and MobileNet for research purposes can yield significant insights into their effectiveness, efficiency, and applicability to various tasks. Several compelling rationales exist for undertaking such an analysis: assessing performance by comparing accuracy, precision, recall, and F1-scores across specific datasets or tasks; evaluating computational efficiency by examining training duration, inference speed, and resource demands; investigating model complexity by considering the number of parameters, layers, and architectural distinctions; assessing transfer learning capabilities to determine each model's adaptability to novel tasks or domains; comparing feature extraction to evaluate the quality and interpretability of extracted features; testing robustness to gauge performance under diverse conditions; determining task-specific suitability to identify optimal models for particular image classification challenges; analyzing trade-offs between accuracy and computational efficiency; evaluating scalability to examine performance as dataset size or class numbers increase; comparing interpretability to assess the ease of understanding and visualizing decision-making processes; evaluating fine-tuning potential to determine the models' capacity for performance improvement through fine-tuning; and benchmarking against state-of-the-art architectures to compare these models with more recent designs. This comprehensive comparative analysis can aid researchers and practitioners in making well-informed decisions regarding model selection for specific applications and contribute to the ongoing advancement of image classification techniques.

4.1. Graphs and Confusion Matrix

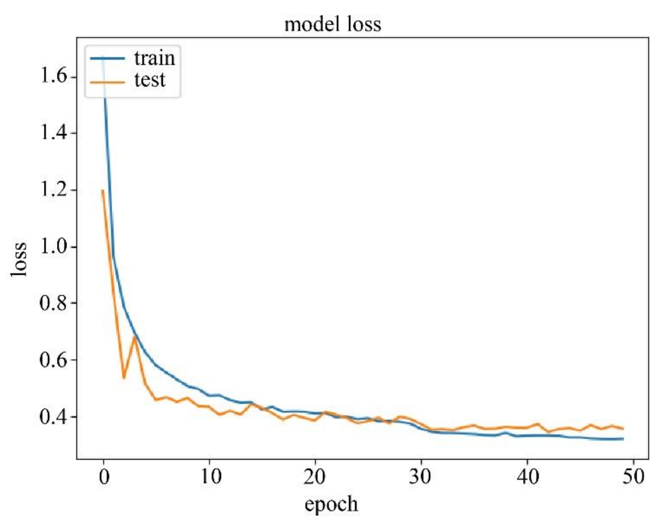
Figures 5, 6, 7, and 8 show the relationships between the iteration count and accuracy/loss for the fine-tuned deep learning models. Table 2 summarizes the training and validation accuracies of the deep learning models. After training for 32 epochs, VGG16 achieved a validation accuracy of 86.28%. The graphical representation indicates that accuracy and loss demonstrate a harmonious and consistent pattern. After 36 training epochs, ResNet50 achieved a validation accuracy of 86.13%. Graphical representations of the accuracy and loss of the ResNet50 model exhibited minimal fluctuations throughout the training process. AlexNet attained a validation accuracy of 80.71% after 20 training epochs.

Table 2. Summary of training and validation accuracy

Model	Training Accuracy	Validation Accuracy
VGG16	87.35%	86.28%
ResNet50	86.75%	86.13%
AlexNet	83.92%	80.71%
MobileNet	98.39%	98.44%



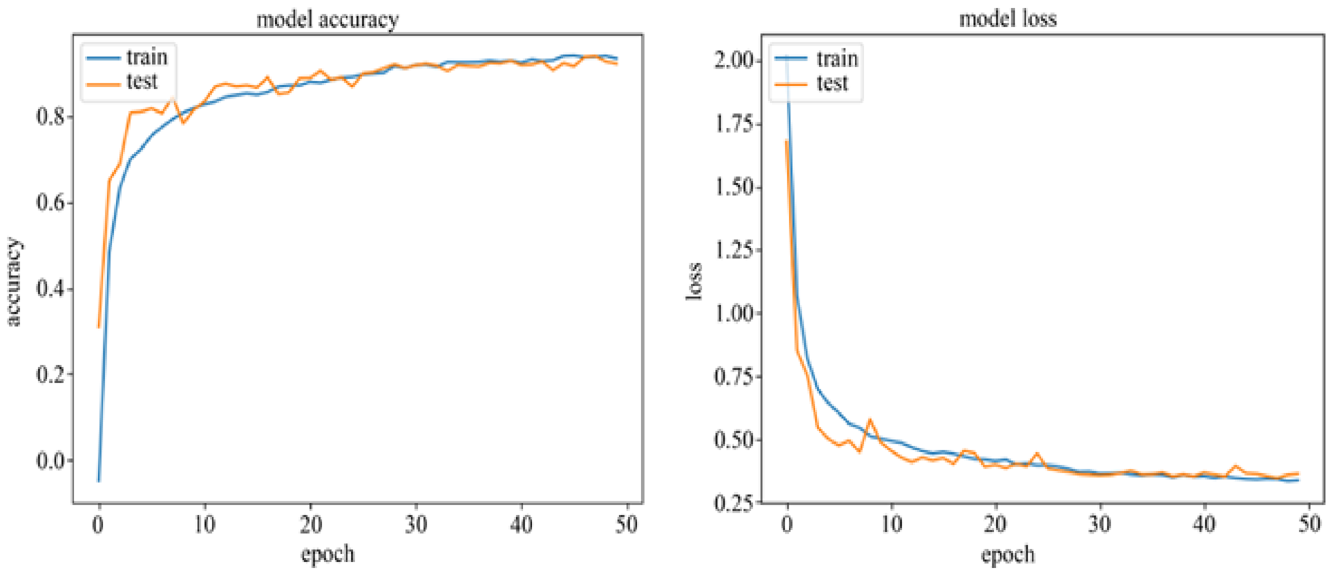
(a) Model accuracy



(b) Model loss

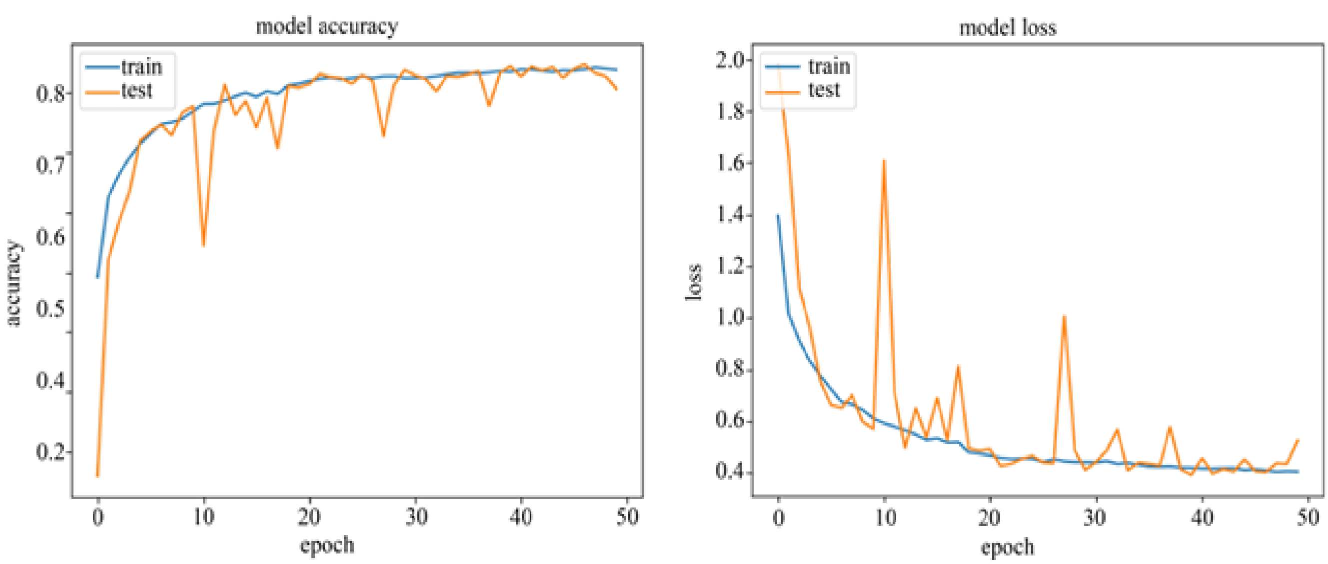
Fig. 5 VGG16 training results

The charts depicting the accuracy and loss for the AlexNet model illustrate persistent fluctuations throughout the training duration, ultimately impacting the model's final validation accuracy. To substantiate the assertion that MobileNet is a superior framework for classifying Surface Mount Device (SMD) components, a comprehensive evaluation is necessary. This evaluation should encompass a comparative analysis with other widely utilized deep-learning architectures, including ResNet, VGG, and Inception. The comparison should focus on key performance metrics, such as accuracy, precision, recall, and F1-score, while also considering computational efficiency and inference speed. Rigorous error analysis is essential, involving the identification of common misclassification patterns, examination of confusion matrices, and investigation of how factors such as image quality, illumination, and component orientation affect classification accuracy. It is crucial to discuss instances of misclassification by presenting specific examples, analyzing the underlying causes, and proposing potential remedies or enhancements. The evaluation should also consider dataset characteristics, examining MobileNet's performance across various SMD component categories and its ability to generalize to novel or unseen component types. Assessing model robustness by testing performance under diverse conditions and evaluating transfer-learning capabilities for adapting to new SMD component types will provide valuable insights. Comparing MobileNet with conventional computer vision techniques and discussing the balance between model size, computational demands, and classification accuracy will help to elucidate its advantages and limitations. By addressing these aspects, a more comprehensive and compelling argument for MobileNet's efficacy in SMD component classification can be presented while acknowledging potential shortcomings and areas for improvement. MobileNet achieved a validation accuracy of 98.44% after training for 25 epochs. The accuracy and loss graphs show that they exhibit relatively smooth and complementary behavior.



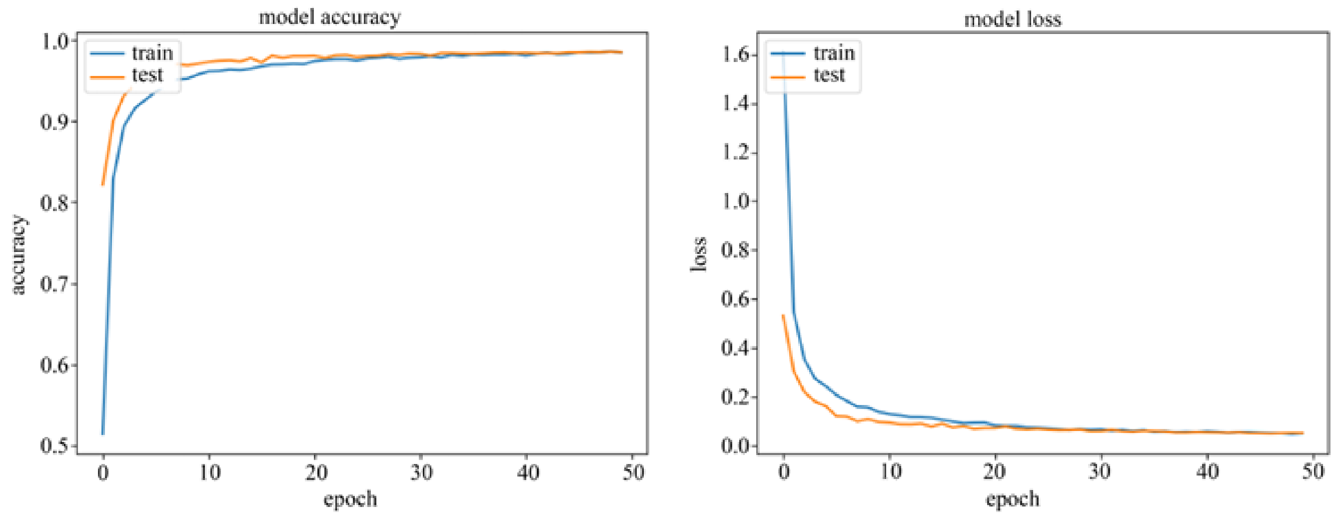
(a) Model Accuracy **(b) Model Loss**

Fig. 6 ResNet50 training results



(a) Model Accuracy **(b) Model Loss**

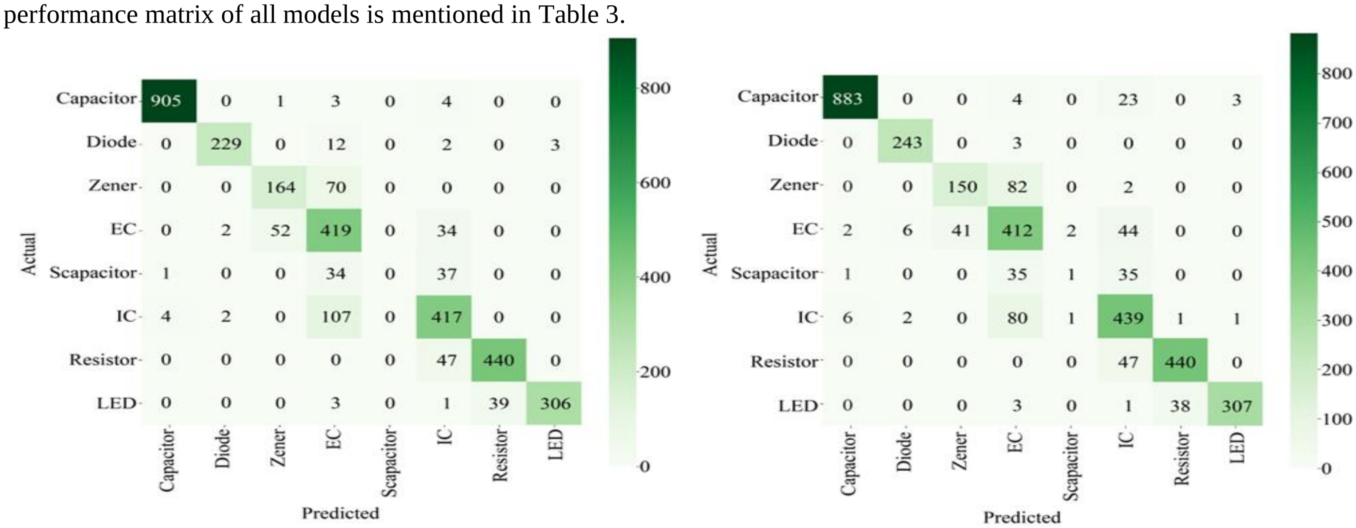
Fig. 7 AlexNet training results



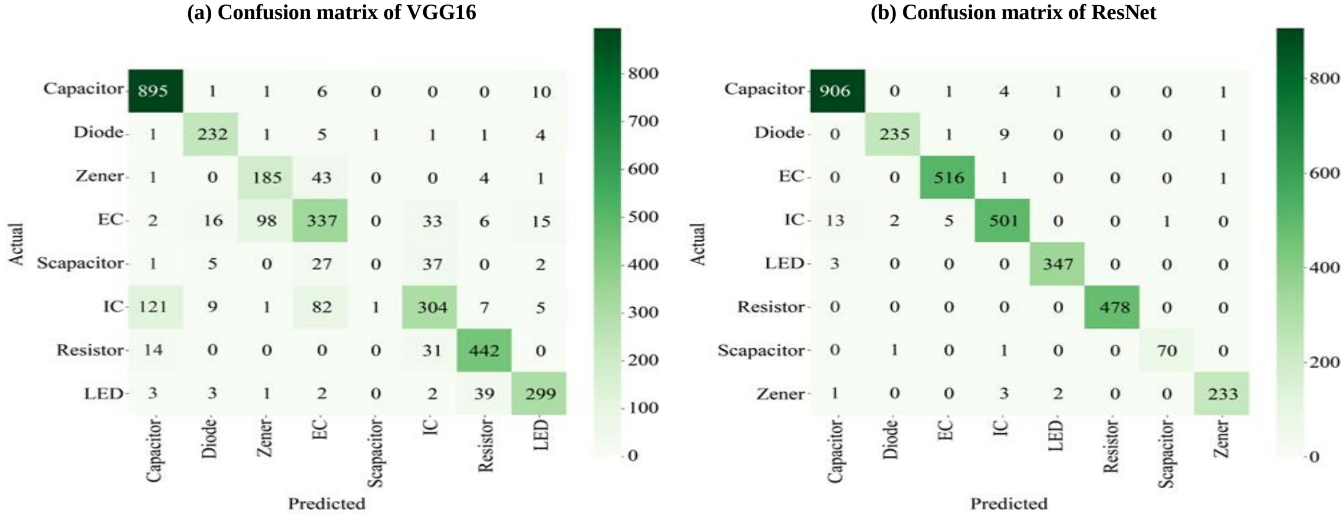
(a) Model Accuracy **(b) Model Loss**

Fig. 8 MobileNet training results

Figure 9 shows the confusion matrices for the VGG16, ResNet50, AlexNet, and MobileNet classifications. The overall performance matrix of all models is mentioned in Table 3.



(a) Confusion matrix of VGG16 **(b) Confusion matrix of ResNet**



(c) Confusion matrix of AlexNet **(d) Confusion matrix of MobileNet**

Fig. 9 Confusion matrices

Figures 10, 11, 12, and 13 present random samples of the validation images, their predicted labels, and the associated predicted probabilities. The accuracy and loss charts demonstrate a reciprocal relationship. As training progresses, accuracy increases while loss diminishes, reaching a minimum at the final iteration. The validation results indicate that the fine-tuned networks effectively predicted the class of electronic components in images, regardless of their size, color, or orientation, showing exceptional performance.

Fig. 10 VGG16 image validation

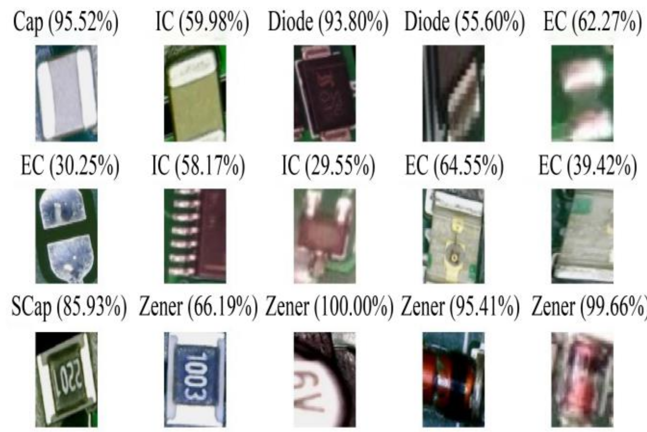


Fig. 12 AlexNet image validation

Fig. 11 ResNet50 image validation

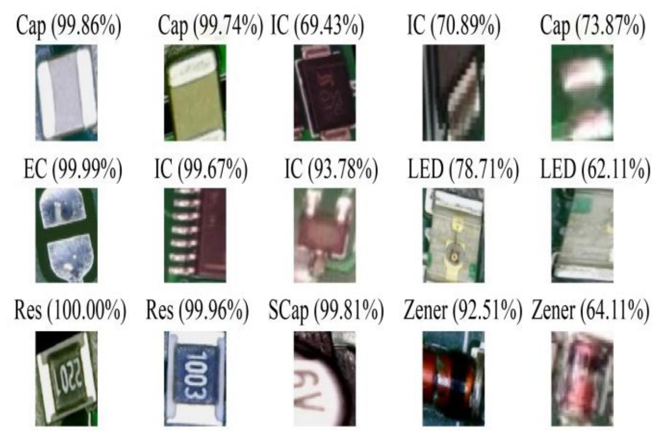


Fig. 13 MobileNet image validation

Table 3. Performance matrix of the trained model

Model	Class	Precision	Recall	Accuracy	F1 Score	Overall Acc.
VGG16	Capacitor	0.99	0.99	1	0.99	**0.965**
	Diode	0.98	0.93	0.99	0.96	
	Zener	0.76	0.7	0.96	0.73	
	EC	0.65	0.83	0.91	0.73	
	SCapacitor	0	0	0.98	0	
	IC	0.77	0.79	0.93	0.78	
	Resistor	0.92	0.9	0.97	0.91	
	LED	0.99	0.88	0.99	0.93	
ResNet50	Capacitor	0.99	0.97	0.99	0.98	**0.965**
	Diode	0.97	0.99	1	0.98	
	Zener	0.79	0.64	0.96	0.71	
	EC	0.67	0.81	0.91	0.73	
	SCapacitor	0.25	0.01	0.98	0.03	
	IC	0.74	0.83	0.93	0.78	
	Resistor	0.92	0.9	0.97	0.91	

	LED	0.99	0.88	0.99	0.93	
AlexNet	Capacitor	0.86	0.98	0.95	0.92	**0.951**
	Diode	0.87	0.94	0.99	0.91	
	Zener	0.64	0.79	0.95	0.71	
	EC	0.67	0.66	0.9	0.67	
	SCapacitor	0	0	0.98	0	
	IC	0.75	0.57	0.9	0.65	
	Resistor	0.89	0.91	0.97	0.9	
	LED	0.89	0.86	0.97	0.87	
MobileNet	Capacitor	0.98	0.99	0.99	0.99	**0.996**
	Diode	0.99	0.96	1	0.97	
	Zener	0.99	1	1	0.99	
	EC	0.97	0.96	0.99	0.96	
	SCapacitor	0.99	0.99	1	0.99	
	IC	1	1	1	1	
	Resistor	0.99	0.97	1	0.98	
	LED	0.99	0.97	1	0.98	

MobileNet achieved the highest level of accuracy, reaching 98.44%, surpassing all other deep learning models utilized in this study. In contrast, ResNet50, AlexNet, and VGG16 attained accuracies of 86.13%, 80.71%, and 86.28%, respectively, following parameter fine-tuning of reused networks. Furthermore, increasing the number of epochs does not consistently enhance the performance of the retrained network. The potential cause lies in the adverse impact on the overall dataset quality and classifier performance when combining sub-par-quality images with high-quality images. Multiple validation trials were conducted to verify the reliability of the classification model devised in this study. The general precision of the model is associated with the precision of the validation images. Inaccurate results indicate a low likelihood of accurate predictions, whereas higher precision indicates improved classification.

5. Conclusion

This study evaluated the efficacy of fine-tuned deep learning models by examining their average accuracy and validation image utilization. After arranging the parameters of the adapted deep neural networks, accuracy rates of 98.44%, 86.28% for VGG16, 86.13% for ResNet50, and 80.71% for AlexNet were achieved when utilizing and large dataset. Variations in the characteristics of the components, such as type, size, orientation, and structure, did not significantly affect the overall accuracy of the retrained networks. Implementing data augmentation methods such as rotation, random shifts, zoom, and flips has been found to significantly increase the variety of training data, potentially enhancing the model's performance.

The accuracy of the model is directly related to the accuracy of the validation images, with a higher accuracy indicating better classification. According to the findings in this study, this technology closes the divide between tangible prototypes and their conceptual depictions, enabling engineers to make well-informed choices throughout the design and development phases. It has the potential to advance reverse engineering methodologies and improve product excellence.

These technologies have the potential to improve automated assembly processes, inventory tracking, and quality assurance in the field of electronic production.

References

[1] SeungGeun Youn, YounAe Lee, and TaeHyung Park, "Automatic Classification of SMD Packages using Neural Network," *IEEE/SICE International Symposium on System Integration*, Tokyo, Japan, pp. 790-795, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[2] D.U. Lim, Y.G. Kim, and T.H. Park, "SMD Classification for Automated Optical Inspection Machine Using Convolution Neural Network," *Third IEEE International Conference on Robotic Computing*, Naples, Italy, pp. 395-398, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] Stanisław Hożyń, "Convolutional Neural Networks for Classifying Electronic Components in Industrial Applications," *Energies*, vol. 16, no. 2, pp. 1-22, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Emel Soylu, and İbrahim Kaya, "Classification of Electronics Components Using Deep Learning," *Sakarya University Journal of Computer and Information Sciences*, vol. 7, no. 1, pp. 36-45, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Longfei Zhou, and Lin Zhang, "A Novel Convolutional Neural Network for Electronic Component Classification with Diverse Backgrounds," *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 13, no. 1, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Qiang Zhang et al., “Image Defect Classification of Surface Mount Technology Welding Based on the Improved Resnet Model,” *Journal of Engineering Research*, vol. 12, no. 2, pp. 154-162, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] Van-Truong Nguyen, and Huy-Anh Bui, “A Real-Time Defect Detection in Printed Circuit Boards Applying Deep Learning,” *EUREKA: Physics and Engineering*, no. 2, pp. 143-153, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Mee Chun Loo et al., “CNN Aided Surface Inspection for SMT Manufacturing,” *15th International Conference on Developments in eSystems Engineering*, Baghdad & Anbar, Iraq, pp. 328-332, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Rui Huang et al., “A Rapid Recognition Method for Electronic Components Based on the Improved Yolo-V3 Network,” *Electronics*, vol. 8, no. 8, pp. 1-18, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Sifundvolesihle Dlamini, Chung-Feng Jeffrey Kuo, and Shin-Min Chao, “Developing a Surface Mount Technology Defect Detection System for Mounted Devices on Printed Circuit Boards Using a Mobilenetv2 with Feature Pyramid Network,” *Engineering Applications of Artificial Intelligence*, vol. 121, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Haihong Pan et al., “A New Image Recognition and Classification Method Combining Transfer Learning Algorithm and Mobilenet Model for Welding Defects,” *IEEE Access*, vol. 8, pp. 119951-119960, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] Aqsa Hassan et al., “An Empirical Analysis of Deep Learning Architectures for Vehicle Make and Model Recognition,” *IEEE Access*, vol. 9, pp. 91487-91499, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Jason Adam, Resnet-50 API, 2019. [Online]. Available: https://jason-adam.github.io/resnet50/

[14] Aqeel Anwar, Difference between Alexnet, Vggnet, Resnet and Inception, Medium, 2019. [Online]. Available: https://towardsdatascience.com/the-w3h-of-alexnetvggnet-resnet-and-inception-7baaaecccc96

[15] Mohanad A. Deif et al., “Diagnosis of Oral Squamous Cell Carcinoma Using Deep Neural Networks And Binary Particle Swarm Optimization On Histopathological Images: An AIoMT Approach,” *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1-13, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Young-Gyu Kim et al., “SMD Defect Classification by Convolution Neural Network and PCB Image Transform,” *IEEE 3rd International Conference on Computing, Communication and Security*, Kathmandu, Nepal, pp. 180-183, 2018. [CrossRef] [Google Scholar] [Publisher Link]