Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.6, December 2025, pp. 3567~3574
e-ISSN: 2548-6861

Detection of Misoriented Polarized Electronic Components on PCBs Using HOG Features and Neural Networks

Eko Rudiawan Jamzuri [1]*, Habyb Nur Ikhsan [2]*
* Department of Electrical Engineering, Politeknik Negeri Batam
ekorudiawan@polibatam.ac.id [1], nhabyb@gmail.com [2]

Article Info

Article history:

Received 2025-09-24
Revised 2025-11-12
Accepted 2025-11-22

Keyword:

Component Orientation Detection, Histogram of Oriented Gradients, Neural Networks, Printed Circuit Board, Polarized Electronic Components

ABSTRACT

Mounting misorientation on polar electronic components in printed circuit boards (PCBs) can cause malfunctions in electronic devices. This study proposes an automatic detection system that utilizes the Histogram of Oriented Gradients (HOG) feature and employs classification using an artificial neural network. The research was conducted by collecting data from PCB images featuring polar components, such as diodes, electrolytic capacitors, and transistors. Once the components are identified, the HOG features are extracted to generate feature vectors used in artificial neural network training. The experiment results show that this system can detect component orientation errors with a high degree of accuracy, achieving accuracy values of 99.5% for transistor components, 97% for electrolyte capacitors, and 93.6% for diodes. Additionally, F1 values and high precision are achieved for all three types of components. The ReLU activation function has been shown to perform best among other activation functions. While the results are promising, further research is necessary to automate the identification of component locations without relying on manual cropping processes.

I. INTRODUCTION

A printed circuit board (PCB) is a crucial component of electronic devices. It is composed of a board with a layer of copper that facilitates the placement and connection of electronic components. PCBs provide mechanical support and electrical connections, making them fundamental to the functionality of complex electronic circuits [1]. Electronic components on PCBs can be classified into surface-mount devices (SMD) and through-hole technology (THT) components. SMDs are smaller and typically installed using machines, while THT components, which require precise placement of their leads into PCB holes, are more challenging to install. Although there are machines available for THT installation, they are costly and limited to specific components [2]. Consequently, THT component installation still largely relies on manual labor.

Manual installation of THT components poses a risk of errors, particularly for polarized components like diodes, electrolytic capacitors, and transistors, which must be installed with the correct orientation. Incorrect orientation can lead to abnormal functioning of the electronic circuits. Moreover, misorientation errors result in significant rework costs and production delays. In a typical electronics manufacturing industry, the cost of reworking a single PCB can range from $5 to $50, depending on complexity, not including downtime costs. This economic impact underscores the need for detection systems that can reduce defect rates and improve manufacturing efficiency.

Post-assembly inspection is crucial for reducing the defect ratio in PCBs. A standard method is visual inspection, which verifies the conformity of installed components with the PCB design. This inspection can be automated using Automated Optical Inspection (AOI), a tool based on cameras and machine vision that identifies various defects in PCBs, including solder issues, component deficiencies, and other physical damages [3].

AOI primarily focuses on component detection to ensure that components are correctly placed on their predetermined footprints. This can be achieved using object detection methods such as You Only Look Once (YOLO) [4], [5], [6], [7], the PCBDet architecture [8], or Graph Network [9]. While

these methods can verify component placement, they cannot determine if the component's orientation is correct. Additional detection methods are required to verify the orientation of installed components.

PolarNet is a method developed to detect component orientation [10]. It consists of three main components: the Siamese Feature Extractor (SFE), Indicator Attention Generator (IAG), and Rotation Detector (RD). SFE extracts features from images, while IAG generates attention maps that help the convolutional network focus on polar markers and other relevant regions. RD uses these attention maps to predict rotation probabilities through fully connected layers. This approach enables PolarNet to detect component rotation on PCBs by leveraging polar marker information to enhance accuracy. However, PolarNet's reliance on polar markers means it may fail to detect the orientation of components with less distinct markers, such as transistors. Additionally, using multiple components, such as SFE, IAG, and RD, increases the model's complexity, requiring more computational resources and longer training times.

This study proposes a component orientation detection system that utilizes Histogram of Oriented Gradients (HOG) features in conjunction with a Neural Network classifier. HOG is a feature extraction algorithm that does not require data training, making it computationally efficient. Computational resources are primarily utilized for the neural network classifier, specifically a Multi-Layer Perceptron (MLP). These characteristics make our approach particularly suitable for deployment in resource-constrained industrial environments where real-time processing and reliability are critical.

We will provide a detailed description of this research in the next section. Section 2 describes the materials used and the methods proposed in this study. Section 3 explains how to verify the research results and the experimental outcomes. The paper concludes with Section 4, which summarizes the overall research results and outlines potential future developments.

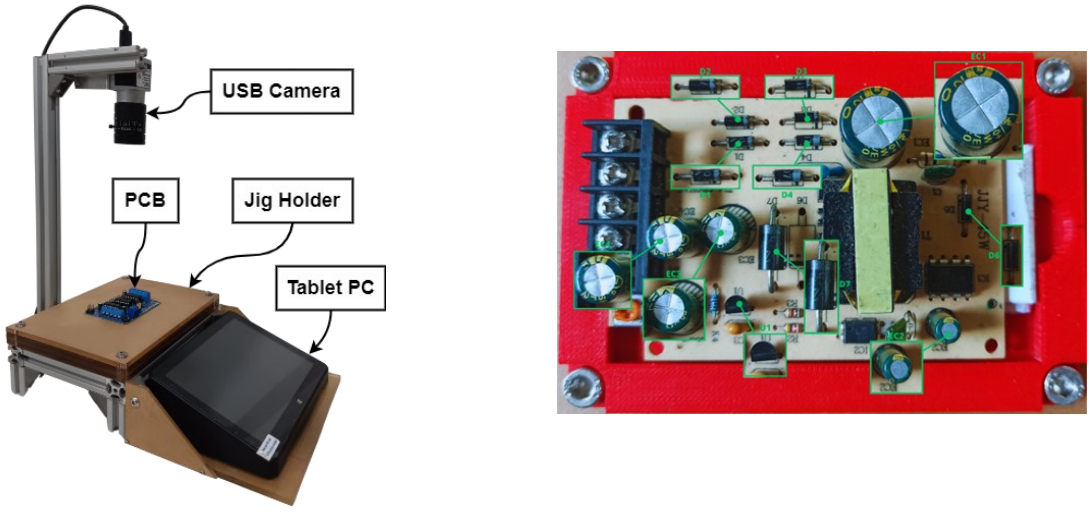
II. METHODS

This section outlines the materials and methods employed in this research. The study begins with the collection of image samples from a PCB containing several polarized components, such as diodes, electrolytic capacitors, and transistors. Once the image samples are acquired, they are cropped to isolate the Region of Interest (ROI) of the targeted components. The features of these ROIs are then extracted using a Histogram of Oriented Gradients (HOG) and subsequently classified using neural networks. The following subsections provide a detailed explanation of each stage of the research.

A. Sample Collection

The sample collection was conducted by placing PCBs on a specially designed jig. A jig is a mechanical fixture used to securely hold the PCB in place, preventing any movement during the image capture process. This jig is also equipped with a camera stand, ensuring consistent positioning for image sampling. Utilizing this setup allows for the capture of images with minimal variation. This technique was adopted from the methodology presented in [11]. Figure 1 (a) shows the equipment used for sample collection. Using this jig, we captured images of PCBs, as shown in Figure 1(b), which reveal both correctly installed components and those with mounting orientation errors, particularly in electrolytic capacitors, diodes, and transistors.

The sample collection process is illustrated in Figure 2(a). First, the PCB sample is positioned on the jig, and images are



(a) (b)

Figure 1. (a) Device for collecting dataset and (b) sample PCB for experiment.

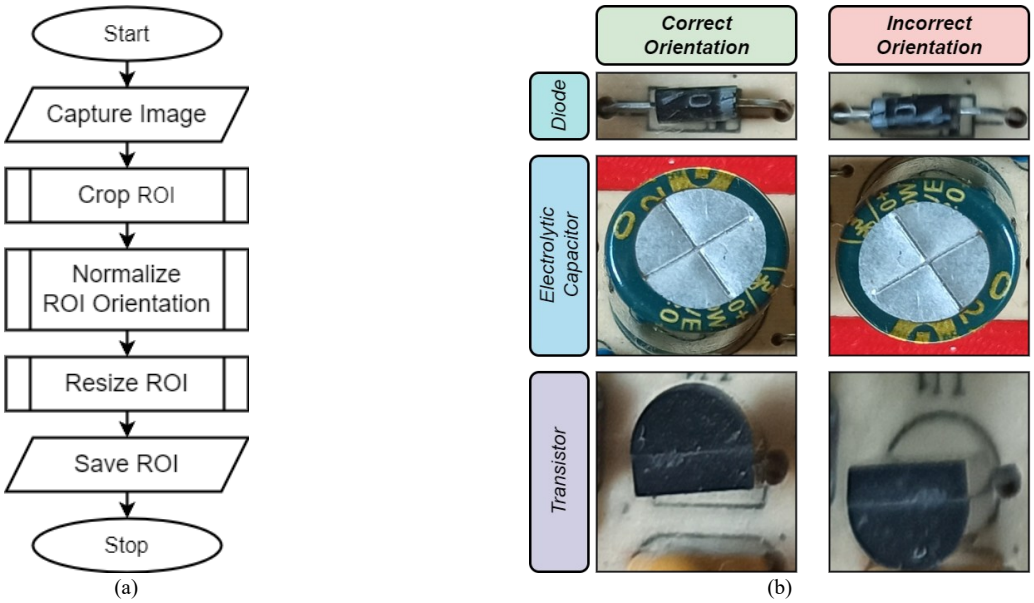


Figure 2. (a) Process for collecting samples and (b) resulted ROI image.

captured using a camera. After obtaining the images, the specific area of the component to be detected is cropped, referred to as the Region of Interest (ROI). This cropping yields the ROI cutout, which is then normalized in orientation. This step is necessary because some components are installed either vertically or horizontally. The normalization process aligns the component orientation horizontally, ensuring uniform aspect ratios across images. Additionally, the ROI images are resized to consistent dimensions, with a standard size of 50×50 pixels. During this process, ROI images are manually labeled to indicate whether they contain polarity errors. The images are then categorized into directories: "correct" for correctly oriented components and "incorrect" for those with orientation errors, as shown in Figure 2(b). These labeled ROI images are subsequently used as training data for polarity error detection, as described in the following subsection.

The dataset collected for this study comprises 11,000 images from a 24V, 3A power supply PCB. The diode dataset comprises 6,000 images (3,000 correct and 3,000 incorrect), the electrolytic capacitor dataset contains 4,000 images (2,000 correct and 2,000 incorrect), and the transistor dataset includes 1,000 images (500 correct and 500 incorrect). The proportional difference in dataset sizes directly reflects the number of each component type present on the PCB: six diodes, four capacitors, and one transistor. Despite the size differences, all component types maintain a balanced 50:50 split between correct and incorrect orientations, preventing class imbalance during training and ensuring unbiased performance evaluation. The dataset was divided into two sets using an 80:20 ratio, with 8,800 images allocated for training and 2,200 images reserved for testing.

B. Component Orientation Detection

The process begins by obtaining training data from ROI images collected and stored during the sample collection process. Next, feature extraction is performed using the Histogram of Oriented Gradients (HOG) algorithm. This process generates vector features that represent the gradient orientation characteristics of the ROI images. These vector features are then used to train a neural network to distinguish between correctly and incorrectly installed components. The performance of the trained neural network is subsequently evaluated using specific metrics to determine the effectiveness of the polarity fault detection model. The following subsections provide detailed explanations of HOG feature extraction and neural network classification.

C. Histogram of Oriented Gradients Feature Extraction

Histogram of Oriented Gradients (HOG) is an algorithm used to extract features from images based on the orientation of color gradients, where a gradient represents the change in color intensity within an image. Initially developed for human detection in combination with Support Vector Machine (SVM) classifiers, HOG has since been applied to various fields, including biometric recognition systems and biomedical imaging. In biometric systems, HOG features have been used to recognize blood vessel patterns in the palm [12] and on the fingers [13], as well as ear shape patterns using Random Forest classifiers [14]. In the biomedical field, HOG features aid in disease diagnosis by extracting features from X-ray medical images [15], [16].

In this study, we use HOG to extract features from component images. The HOG feature extraction was implemented using the scikit-image library integrated with the Orange Data Mining software. Polar components typically

have orientation markers that contrast sharply with the base color, and they often possess asymmetrical shapes that serve as orientation references. The distinct color differences in these markers create pronounced color gradients. The HOG algorithm captures features that represent the gradient orientation of the ROI images. Feature extraction begins by converting the ROI images into grayscale. Gradients are then calculated on these grayscale images using Equations 1 and 2, resulting in gradients along the x-axis and y-axis.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \cdot Y \quad (1)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \cdot Y \quad (2)$$

The gradients along the x-axis and y-axis are then converted into oriented gradients using Equations 3 and 4, where the oriented gradient is represented by its magnitude, g, and direction, θ. The direction θ is normalized between 0 and 180 degrees. These magnitude and direction values are then used to construct a histogram, which statistically represents the distribution of gradient magnitudes and directions. The gradient image is divided into small sub-regions or cells, with each cell consisting of 16×16 pixels. For each cell, a histogram is calculated by accumulating the gradient magnitudes based on their orientation directions. In this study, eight bins are used to describe the distribution of orientations. After obtaining histogram features from each cell, the next step is normalizing these cells to eliminate the effects of lighting variations. This normalization process yields a set of 72 feature vectors that represent the HOG feature of the image.

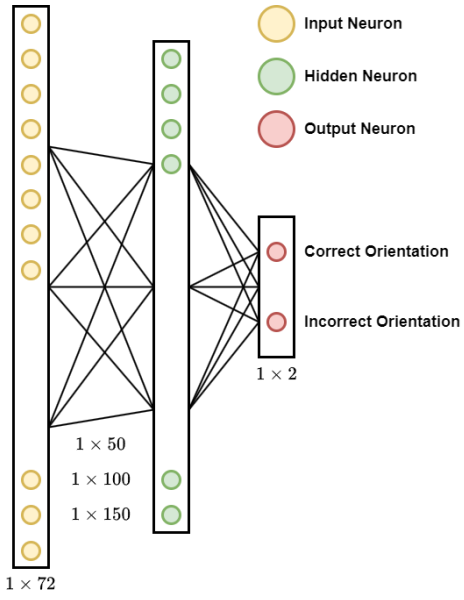


Figure 3. Neural Network Architecture for Classification.

$$g = \sqrt{x^2 + y^2} \quad (3)$$

$$\theta = atan2(y, x) \quad (4)$$

D. Classification Using Neural Network

In this study, we implement a neural network algorithm to recognize patterns derived from HOG features. The pattern recognition process using neural networks involves two main stages: training and testing. The training stage is designed to enable the neural network to learn and recognize patterns from input data represented by HOG features. Conversely, the testing stage evaluates the neural network's ability to accurately predict patterns in new, unseen test data.

The neural network was implemented using Orange Data Mining software for model training and testing. The neural network we designed employs a Multi-Layer Perceptron (MLP) architecture consisting of one input layer, one hidden layer, and one output layer, as illustrated in Figure 3. The input layer comprises 72 neurons, corresponding to the size of the feature vectors obtained after the HOG extraction process. In the hidden layer, we experimented with three different configurations, using 50, 100, and 150 neurons. The output layer contains two neurons, which are responsible for predicting the orientation class—either correct or incorrect. Three activation functions were tested: Rectified Linear Unit (ReLU), TanH, and Logistic Function. The ReLU activation function, as described by Equation (5), produces a positive output when the input is positive and zero when the input is negative, with an output range of 0 to 1. The TanH function, given by Equation (6), generates outputs ranging from -1 to +1, allowing for the representation of negative values. Finally, the logistic function, expressed as Equation (7), outputs a value between 0 and 1, which is typically used to represent probabilities and is often applied in the output layer of neural networks.

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (6)$$

$$f(x) = \frac{1}{1 + e^{-x}} \quad (7)$$

E. Evaluation Method

We validated the neural network's output to assess the performance of the proposed method. The validation process is designed to measure the system's accuracy in detecting whether component images are correctly or incorrectly oriented. As recommended by [17], validation requires a unique dataset that is distinct from the training phase. Consequently, we employed the Leave-One-Out Cross-Validation (LOO-CV) method, which ensures that each test dataset is entirely separate from the training dataset. LOO-CV

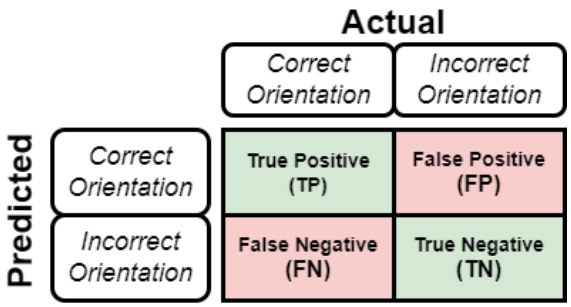


Figure 4. Confusion Matrix for Classification.

is commonly used to evaluate the generalization capability of classification algorithms by sequentially using each data point in the dataset as test data. At the same time, the remainder serves as training data. This process is repeated for every data point, and the overall accuracy is calculated as the average of the individual results, following the approach described in [18].

We evaluated the algorithm's performance using several metrics: Classification Accuracy (CA), Precision, Recall, F1 Score, and Matthews Correlation Coefficient (MCC). These metrics are calculated based on the results of predictions compared to the actual values in the dataset, which are organized in a truth table. The confusion matrix, depicted in Figure 4, is used to represent this comparison, summarizing correct and incorrect predictions for each class [19]. In Figure 4, True Positive (TP) represents the number of component images correctly predicted as having the correct orientation. False Negative (FN) denotes the number of images incorrectly predicted as having the incorrect orientation when they are correct. True Negative (TN) represents images correctly identified as having the incorrect orientation, while False Positive (FP) indicates images incorrectly identified as having the correct orientation. CA, Precision, Recall, F1, and MCC are all derived from these values.

Classification Accuracy (CA) is a fundamental metric in machine learning, particularly crucial in fields where even minor errors are unacceptable, such as in biomedical applications [20]. CA is typically measured using test data that has not been involved in the training process [21]. Although CA is widely used to evaluate the performance of classification algorithms, it has some limitations. One fundamental limitation is that it does not differentiate between types of errors and is heavily influenced by the class distribution within the dataset. The formula for calculating CA is provided in Equation (8). Generally, a higher CA value indicates better performance of the classification algorithm.

$$CA = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Precision, on the other hand, refers to a model's ability to identify positive samples among all correctly predicted positive samples. Precision is calculated as the ratio of True Positives (TP) to the total number of predicted positives, as shown in Equation (9). Precision is essential for understanding the impact of image pre-processing, which often involves various operators that may affect the model's performance [22].

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

Recall measures a model's ability to identify all relevant positive samples within a dataset correctly. It is calculated as the ratio of True Positives (TP) to the total number of actual positive samples, as illustrated in Equation (10). Recall is significant in scenarios involving critical applications, such as medical diagnosis and fraud detection, where missing a positive case can have serious consequences [23].

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

Precision and Recall are often combined into a single metric known as the F1 Score, which is the harmonic mean of Precision and Recall. The F1 Score provides a balanced measure of a model's accuracy, particularly in cases where the data is unbalanced, as shown in Equation (11) [24].

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{11}$$

The Matthews correlation coefficient (MCC) is the most important metric for binary classification, as it provides a comprehensive evaluation of classifier performance [25]. MCC combines sensitivity, specificity, precision, and negative prediction values into a single value that reflects the model's overall performance. The MCC value ranges from -1 to +1, where -1 indicates total disagreement between the prediction and the actual result, while +1 indicates a perfect prediction. This metric can provide a more balanced picture of the model's performance, especially in cases where the class distribution is unbalanced or different types of errors must be accounted for. The MCC value can be obtained from Equation (12).

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (FN + TN) \times (TP + FN) \times (FP + TN)}} \tag{12}$$

III. Results and Discussion

This section presents the evaluation results obtained using the proposed method in this study. We begin by discussing the results of feature extraction from the component images, followed by the classification performance of the neural network. Detailed explanations of these results are provided in the subsequent subsections.

A. Result of HOG Feature Extraction

The feature extraction process, utilizing the Histogram of Oriented Gradients (HOG) algorithm, was applied to the Region of Interest (ROI) images. This process generates

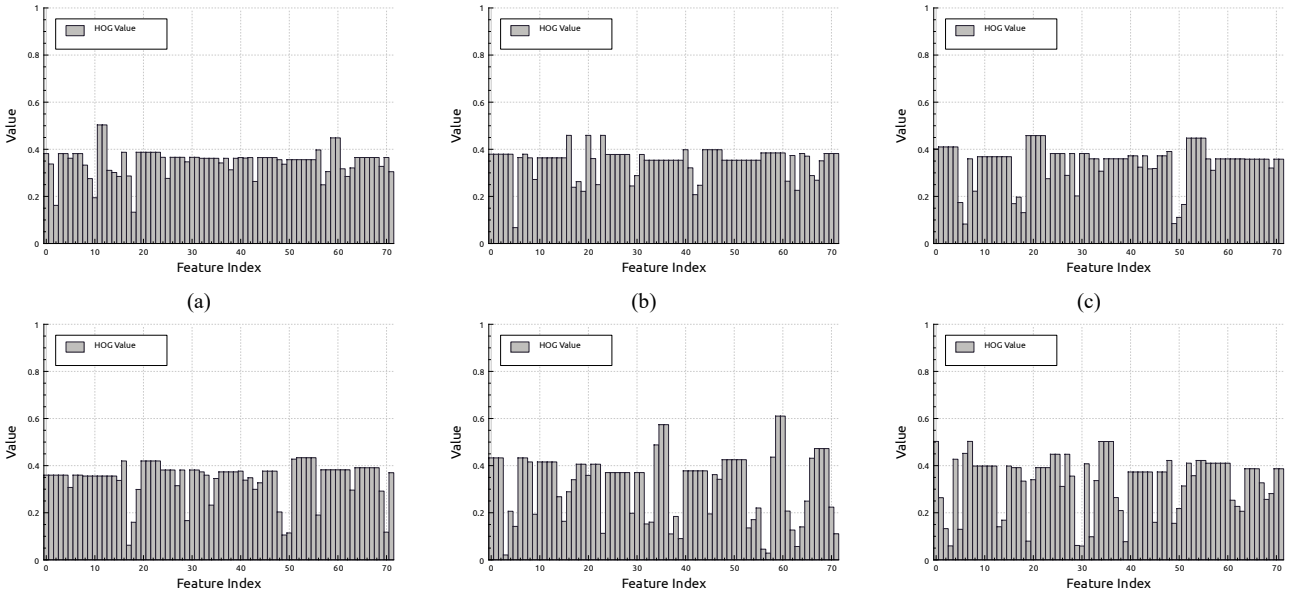


Figure 5. Histogram of Oriented Gradient from Component ROIs.

feature vectors that encapsulate the gradient orientation characteristics of the ROI images. These vectors serve as input for the neural network during the classification stage. Figure 5 illustrates the results of the HOG feature extraction, depicting the distribution of gradient orientations for components with both correct and incorrect orientations. The distinct differences in gradient orientation distributions between correctly and incorrectly oriented components are shown in Figure 5. Figures 5 (a), (c), and (e) illustrate the histograms of the correct orientation of the diode, electrolytic capacitor, and transistor, respectively. Furthermore, Figures 5(b), (d), and (f) display the histograms of the incorrect orientation of these components. Figure 5 demonstrates that HOG features effectively capture the orientation information of the components.

B. Neural Network Classification Result

The performance of the neural network algorithm was evaluated by measuring several metrics, as described in Section 2.2. To identify the optimal configuration, we conducted tests with various settings, differentiating the number of neurons in the hidden layer and the type of activation function used. Specifically, we experimented with three neuron configurations in the hidden layer: 50, 100, and 150 neurons. For the activation function, we tested three types: ReLU, TanH, and Logistic.

Table 1 summarizes the results of the neural network performance tests conducted on diode components, electrolytic capacitors, and transistors. The test results indicate that the neural network, when used to predict HOG features, can effectively detect orientation errors in components. The lowest performance observed during testing showed values of CA = 0.862, F1 = 0.862, Precision = 0.862, Recall = 0.862, and MCC = 0.723. In contrast, the highest performance achieved values of CA = 0.995, F1 = 0.995, Precision = 0.995, Recall = 0.995, and MCC = 0.990. Upon comparing the results from the three components, it was found that the diode component exhibited the lowest performance, with a maximum MCC value of only 0.871, compared to MCC values above 0.9 for the other components. This result is further evidenced by the diode component's CA score of 0.936, whereas the CA scores for the other two components were above 0.95.

The activation function was found to have a significant influence on the neural network's performance. The results show that the ReLU activation function consistently delivered the highest performance across all three components, while the Sigmoid activation function resulted in the lowest performance. On the other hand, the number of neurons in the hidden layer did not significantly impact the network's performance. For instance, the best performance for the diode component was observed with 150 neurons and the ReLU activation function. However, this pattern did not hold for the other components. For the electrolytic capacitor, the highest performance was achieved with just 50 neurons and the ReLU activation function, yielding an MCC of 0.940. In the case of the transistor component, the optimal configuration was 100 neurons with the ReLU activation function, achieving an MCC of 0.990.

IV. CONCLUSIONS

This study addresses the verification of component orientation in the installation of diodes, electrolytic capacitors, and transistors on PCBs. A neural network-based classification method is proposed to verify the orientation of these three polarized components. The neural network

TABLE I
RESULT OF NEURAL NETWORK PERFORMANCE OVER DIFFERENT CONFIGURATION

Component Name	Number of Neurons	Activation Function	CA	F1	Precision	Recall	MCC
Diode	50	ReLU	0.929	0.929	0.929	0.929	0.858
	100	ReLU	0.932	0.932	0.932	0.932	0.863
	150	**ReLU**	**0.936**	**0.936**	**0.936**	**0.936**	**0.871**
	50	TanH	0.912	0.912	0.913	0.912	0.825
	100	TanH	0.922	0.922	0.923	0.922	0.845
	150	TanH	0.928	0.928	0.929	0.928	0.857
	50	Sigmoid	0.862	0.862	0.862	0.862	0.723
	100	Sigmoid	0.880	0.880	0.880	0.880	0.760
	150	Sigmoid	0.893	0.893	0.893	0.893	0.786
Electrolyte Capacitor	**50**	**ReLU**	**0.970**	**0.970**	**0.970**	**0.970**	**0.940**
	100	ReLU	0.969	0.969	0.969	0.969	0.938
	150	ReLU	0.970	0.970	0.970	0.970	0.940
	50	TanH	0.953	0.952	0.953	0.953	0.905
	100	TanH	0.956	0.956	0.956	0.956	0.912
	150	TanH	0.964	0.964	0.964	0.964	0.927
	50	Sigmoid	0.905	0.905	0.905	0.905	0.810
	100	Sigmoid	0.915	0.915	0.915	0.915	0.830
	150	Sigmoid	0.920	0.920	0.920	0.920	0.840
Transistor	50	ReLU	0.985	0.985	0.985	0.985	0.970
	100	**ReLU**	**0.995**	**0.995**	**0.995**	**0.995**	**0.990**
	150	ReLU	0.985	0.985	0.985	0.985	0.970
	50	TanH	0.985	0.985	0.985	0.985	0.970
	100	TanH	0.980	0.980	0.980	0.980	0.960
	150	TanH	0.975	0.975	0.975	0.975	0.975
	50	Sigmoid	0.975	0.975	0.975	0.975	0.950
	100	Sigmoid	0.970	0.970	0.970	0.970	0.940
	150	Sigmoid	0.970	0.970	0.970	0.970	0.940

predicts whether the component orientation is correct or reversed using feature extraction results from the HOG algorithm as input. The test results demonstrate that the proposed method effectively detects orientation errors, as evidenced by the accuracy scores obtained for each component: 0.936 for diodes, 0.970 for electrolytic capacitors, and 0.995 for transistors. Additionally, the MCC values indicate strong performance, with 0.871 for diodes, 0.940 for electrolytic capacitors, and 0.990 for transistors. The tests further reveal that the ReLU activation function significantly enhances prediction performance, achieving the highest values in CA, Precision, Recall, and MCC metrics when used in the neural network. Despite the promising results, further work should focus on automating the component location detection process to eliminate the need for manual cropping.

ACKNOWLEDGEMENTS

The Politeknik Negeri Batam fully funds this research. The author would like to thank Politeknik Negeri Batam for the resources and facilities that supported this research.

REFERENCES

[1] Q. Ling and N. A. M. Isa, "Printed Circuit Board Defect Detection Methods Based on Image Processing, Machine Learning and Deep Learning: A Survey," *IEEE Access*, vol. 11, pp. 15921–15944, 2023, doi: 10.1109/ACCESS.2023.3245093.

[2] M. Polikarpov, Y. Mehmood, D. Boiar, L. Schulte, and J. Deuse, "A Cost-Efficient Robotic Through-Hole Assembly System for Dual-Pin Component Insertion," in *2023 11th International Conference on Control, Mechatronics and Automation (ICCMA)*, Grimstad, Norway: IEEE, Nov. 2023, pp. 271–277. doi: 10.1109/ICCMA59762.2023.10375039.

[3] Y. Zhou, M. Yuan, J. Zhang, G. Ding, and S. Qin, "Review of vision-based defect detection research and its perspectives for printed circuit board," *J. Manuf. Syst.*, vol. 70, pp. 557–578, Oct. 2023, doi: 10.1016/j.jmsy.2023.08.019.

[4] Q. Ling, N. A. M. Isa, and M. S. M. Asaari, "Precise Detection for Dense PCB Components Based on Modified YOLOv8," *IEEE Access*, vol. 11, pp. 116545–116560, 2023, doi: 10.1109/ACCESS.2023.3325885.

[5] P. Kim, X. Huang, and Z. Fang, "SSD PCB Component Detection Using YOLOv5 Model," *J. Inf. Commun. Converg. Eng.*, vol. 21, no. 1, pp. 24–31, Mar. 2023, doi: 10.56977/jicce.2023.21.1.24.

[6] H. Xin, Z. Chen, and B. Wang, "PCB Electronic Component Defect Detection Method based on Improved YOLOv4 Algorithm," *J. Phys. Conf. Ser.*, vol. 1827, no. 1, p. 012167, Mar. 2021, doi: 10.1088/1742-6596/1827/1/012167.

[7] J. Li, J. Gu, Z. Huang, and J. Wen, "Application Research of Improved YOLO V3 Algorithm in PCB Electronic Component

Detection," *Appl. Sci.*, vol. 9, no. 18, p. 3750, Sept. 2019, doi: 10.3390/app9183750.

[8] B. Li, S. Palayew, F. Li, S. Abbasi, S. Nair, and A. Wong, "PCBDet: An efficient deep neural network object detection architecture for automatic PCB component detection on the edge," *Electron. Lett.*, vol. 60, no. 2, p. e13042, Jan. 2024, doi: 10.1049/ell2.13042.

[9] C.-W. Kuo, J. D. Ashmore, D. Huggins, and Z. Kira, "Data-Efficient Graph Embedding Learning for PCB Component Detection," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa Village, HI, USA: IEEE, Jan. 2019, pp. 551–560. doi: 10.1109/WACV.2019.00064.

[10] H. Ma and H. Zhang, "PCB Component Rotation Detection Based on Polarity Identifier Attention," in *Artificial Neural Networks and Machine Learning – ICANN 2023*, L. Iliadis, A. Papaleonidas, P. Angelov, and C. Jayne, Eds., in Lecture Notes in Computer Science. Cham: Springer Nature Switzerland, 2023, pp. 140–151. doi: 10.1007/978-3-031-44201-8_12.

[11] A. Munawaroh and E. R. Jamzuri, "Automatic optical inspection for detecting keycaps misplacement using Tesseract optical character recognition," *Int. J. Electr. Comput. Eng. IJECE*, vol. 13, no. 5, Art. no. 5, Oct. 2023, doi: 10.11591/ijece.v13i5.pp5147-5155.

[12] M. Wulandari, R. Chai, B. Basari, and D. Gunawan, "Hybrid Feature Extractor Using Discrete Wavelet Transform and Histogram of Oriented Gradient on Convolutional-Neural-Network-Based Palm Vein Recognition," *Sensors*, vol. 24, no. 2, p. 341, Jan. 2024, doi: 10.3390/s24020341.

[13] J. Zeng, Y. Chen, Y. Zhai, J. Gan, W. Feng, and F. Wang, "A Novel Finger-Vein Recognition Based on Quality Assessment and Multi-Scale Histogram of Oriented Gradients Feature:," *Int. J. Enterp. Inf. Syst.*, vol. 15, no. 1, pp. 100–115, Jan. 2019, doi: 10.4018/IJEIS.2019010106.

[14] M. Hasan Mutar, E. Hammodi Ahmed, M. R. Mohamed Alsemawi, H. O. Hanoosh, and A. H. Abbas, "Ear recognition system using random forest and histograms of oriented gradients techniques," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 27, no. 1, p. 181, July 2022, doi: 10.11591/ijeecs.v27.i1.pp181-188.

[15] A. M. Ayalew, A. O. Salau, B. T. Abeje, and B. Enyew, "Detection and classification of COVID-19 disease from X-ray images using convolutional neural networks and histogram of oriented gradients," *Biomed. Signal Process. Control*, vol. 74, p. 103530, Apr. 2022, doi: 10.1016/j.bspc.2022.103530.

[16] M. Jawahar *et al.*, "Computer-aided diagnosis of COVID-19 from chest X-ray images using histogram-oriented gradient features and Random Forest classifier," *Multimed. Tools Appl.*, vol. 81, no. 28, pp. 40451–40468, Nov. 2022, doi: 10.1007/s11042-022-13183-6.

[17] S. Sivamani, S. I. Chon, D. Y. Choi, and J. H. Park, "Investigating and Suggesting the Evaluation Dataset for Image Classification Model," *IEEE Access*, vol. 8, pp. 173599–173608, 2020, doi: 10.1109/ACCESS.2020.3024575.

[18] M. Magnusson, M. Andersen, J. Jonasson, and A. Vehtari, "Bayesian leave-one-out cross-validation for large data," in *Proceedings of the 36th International Conference on Machine Learning*, PMLR, May 2019, pp. 4244–4253. Accessed: Apr. 08, 2024. [Online]. Available: https://proceedings.mlr.press/v97/magnusson19a.html

[19] D. Krstinić, L. Šerić, and I. Slapničar, "Comments on 'MLCM: Multi-Label Confusion Matrix,'" *IEEE Access*, vol. 11, pp. 40692–40697, 2023, doi: 10.1109/ACCESS.2023.3267672.

[20] R. Ramola, S. Jain, and P. Radivojac, "Estimating classification accuracy in positive-unlabeled learning: characterization and correction strategies," in *Biocomputing 2019*, Kohala Coast, Hawaii, USA: WORLD SCIENTIFIC, Nov. 2018, pp. 124–135. doi: 10.1142/9789813279827_0012.

[21] E. A. Platanios, H. Poon, T. M. Mitchell, and E. Horvitz, "Estimating accuracy from unlabeled data: a probabilistic logic approach," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, in NIPS'17. Red Hook, NY, USA: Curran Associates Inc., Desember 2017, pp. 4364–4373.

[22] B. Bilalli, A. Abelló, T. Aluja-Banet, and R. Wrembel, "PRESISTANT: Learning based assistant for data pre-processing," *Data Knowl. Eng.*, vol. 123, p. 101727, Sept. 2019, doi: 10.1016/j.datak.2019.101727.

[23] Y. Yang, C. Miller, P. Jiang, and A. Moghtaderi, "A Case Study of Multi-class Classification with Diversified Precision Recall Requirements for Query Disambiguation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Virtual Event China: ACM, July 2020, pp. 1633–1636. doi: 10.1145/3397271.3401315.

[24] J. C. Fernández, M. Carbonero, P. A. Gutiérrez, and C. Hervás-Martínez, "Multi-objective evolutionary optimization using the relationship between F1 and accuracy metrics in classification tasks," *Appl. Intell.*, vol. 49, no. 9, pp. 3447–3463, Sept. 2019, doi: 10.1007/s10489-019-01447-y.

[25] D. Chicco and G. Jurman, "The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification," *BioData Min.*, vol. 16, no. 1, p. 4, Feb. 2023, doi: 10.1186/s13040-023-00322-4.