Computer Science and Information Technologies
Vol. 7, No. 3, November 2026, pp. 346~352
ISSN: 2722-3221, DOI: 10.11591/csit.v7i3.p346-352

Detection and translation of logic gate images into Boolean functions

Muhammad Shidqii Taqiyyuddin[1], Muhammad Subali[2]
[1]Department of Electrical Engineering, Faculty of Technology and Engineering, Gunadarma University, Jakarta, Indonesia
[2]Faculty of Informatics Engineering, Cendekia Abditama University, Tangerang, Indonesia

Article Info

Article history:

Received Aug 27, 2025
Revised Jun 15, 2026
Accepted Jul 13, 2026

Keywords:

Artificial intelligence
Boolean function
Logic gate
Object detection
YOLOv5s

ABSTRACT

Artificial intelligence (AI) has become an essential tool in solving complex problems effectively and efficiently. In digital electronics, translating logic gate circuits into Boolean functions can be challenging, especially for more complex structures. This study presents the design of a detection and translation system for logic gate images into Boolean functions using the You Only Look Once (YOLOv5) object detection model. A dataset of 800 images was collected using a smartphone camera under varied lighting conditions and preprocessing to ensure robustness. The dataset was divided into 70% for training, 20% for validation, and 10% for testing. Training was conducted using YOLOv5s with batch size 32, 100 epochs, and pre-trained weights. The trained model achieved strong results with an overall mean average precision (mAP@0.5) of 0.922, precision of 0.98, and recall of 0.984. The confusion matrix confirmed accurate detection across all classes, with minimal misclassification. Furthermore, the translator system successfully converted recognized objects into correct Boolean expressions, with results validated in multiple test cases. This demonstrates that the proposed system can reliably automate circuit image-to-Boolean translation, bridging image recognition and symbolic computation.

Corresponding Author:

Muhammad Shidqii Taqiyyuddin
Department Electrical Engineering, Faculty of Technology and Engineering, Gunadarma University
Jakarta, Indonesia
Email: taqiyy17@gmail.com

1. INTRODUCTION

Artificial intelligence (AI), particularly computer vision, has transformed numerous domains by enabling automated solutions for complex visual recognition tasks [1]. Within the field of digital electronics, the translation of logic gate circuits into Boolean functions represents a fundamental yet intricate process, particularly when circuit designs reach higher levels of complexity. Foundational studies and textbooks on digital logic, such as those by Maini [2] and Mano and Ciletti [3], provide the theoretical basis for understanding logic gate symbols, circuit representations, and their corresponding Boolean expressions. These works establish the essential principles upon which automated approaches, such as those using AI and computer vision, can be developed.

Previous studies have attempted to address this challenge by introducing computer vision and deep learning techniques for recognizing and interpreting logic gates. For instance, Datta *et al*. [4] explored the detection of logic gates from document images using mathematical morphology, while Yun *et al*. [5] investigated deep neural networks for automatic recognition of engineering diagrams. Similarly, Aviles-Cruz *et al*. [6] utilized augmented reality and smartphones to identify integrated logic circuits, but this approach differs from diagram-based symbol recognition. Despite these advances, existing approaches still

face limitations in accurately recognizing and translating logic gate symbols under varying lighting conditions, orientations, and drawing styles, highlighting the need for a more robust detection framework.

The rise of convolutional neural networks (CNNs) and object detection frameworks such as the You Only Look Once (YOLO) family has provided new opportunities to tackle these challenges. YOLOv1 introduced by Redmon *et al*. [7] demonstrated the feasibility of real-time object detection, followed by further improvements in YOLOv4 [8] and YOLOv5 [9]. These models offer high accuracy and computational efficiency, making them suitable for applications involving real-world image recognition. Studies have applied YOLO variants in defect detection [10]–[11], safety monitoring [12]–[16], symbol recognition [17]–[18], agricultural sector [19]–[21], electrical circuit analysis [22]–[24], and sign language [25]. However, the application of YOLOv5 in translating logic gate symbols into Boolean functions remains underexplored.

This study proposes the development of a system capable of detecting and translating logic gate symbol images into Boolean functions using YOLOv5. A custom dataset of 800 images was collected under varied lighting conditions and backgrounds to enhance generalizability. The dataset consists of standardized logic gate symbols generated with Draw.io, providing robustness under diverse capture conditions. Images were annotated and processed using Roboflow to generate training, validation, and testing subsets. The trained YOLOv5 model was then integrated into a translation system capable of recognizing logic gate components (inputs, operators) and converting them into corresponding Boolean expressions.

The main contributions of this work are: i) the creation of a diverse dataset of printed logic gate symbols; ii) the implementation of YOLOv5 for accurate detection of logic gate components; iii) the design of a translation system that maps detected gates into Boolean functions; and iv) validation of the system performance through quantitative metrics (precision, recall, and mean average precision or mAP) and qualitative evaluation of translated expressions. This research bridges the gap between computer vision and digital logic design, offering a tool that simplifies the process of Boolean function derivation from visual circuit representations. The findings contribute to both the academic field of AI in electronics and practical applications in education and digital system design.

2. METHOD

2.1. Dataset acquisition

The dataset used in this study consisted of 800 images of digital logic gate symbols. The symbols were first created digitally using the Draw.io application and then printed with an EPSON L15150 printer to ensure standardized and structured shapes. These printed sheets were subsequently photographed using a POCO M5 smartphone camera under varied lighting conditions, angles, and backgrounds to introduce natural diversity and simulate real-world variability. This strategy ensured that, while the symbols remained consistent in design, the captured images reflected practical variations encountered in real environments, making the dataset more robust for training the detection model. The dataset covered seven types of logic gates (AND, OR, XOR, NAND, NOR, XNOR, NOT) along with two consistent input symbols (Input 1 and Input 2). Each image contained at least one logic gate and two input elements, reflecting typical circuit representations used in digital electronics.

2.2. Data preprocessing

Before training, several preprocessing steps were applied to prepare the dataset for YOLOv5 training:

- Resizing: all images were resized to 256×256 pixels. This size was chosen as a trade-off between preserving sufficient detail of logic gate symbols while keeping the computational load manageable on the available hardware.
- Normalization: pixel values were normalized to a [0,1] scale, improving convergence during training.
- Annotation: each object (logic gates and inputs) was annotated using bounding boxes following the YOLO 1.1 format. Annotations were performed using Roboflow, which also ensured consistent label formatting.
- Data augmentation: several augmentation techniques such as random rotations, flipping, scaling, and brightness adjustments were applied to increase dataset variability. Augmentation helped improve the robustness of the model by simulating real-world variations such as tilted drawings or uneven illumination.

After preprocessing, the dataset was split into: i) 70% for training (560 images); ii) 20% for validation (160 images); and iii) 10% for testing (80 images). This split ensured balanced training and reliable evaluation of model performance.

2.3. Hardware and software requirements

The training and implementation were conducted on a laptop with the following specifications: i) processor: AMD Ryzen 5 7535HS with Radeon Graphics (3.30 GHz); ii) memory: 16 GB RAM, 1 TB

SSD; iii) GPU: NVIDIA GeForce RTX 2050, 4 GB VRAM; iv) operating system: Windows 11 64-bit (version 24H2); and v) software and tools:

- Anaconda Navigator (v2.3.1) for environment management,
- Jupyter Notebook (v6.4.12) as the development interface,
- Python (v3.12.8) as the programming environment,
- Visual Studio Code (v1.100.2) for additional coding tasks,
- Google Chrome (v136.0.7103.114) for web-based tools,
- Roboflow for dataset preprocessing and annotation,
- Draw.io and iLoveIMG for creating logic gate symbols and image adjustments,
- GitHub (Ultralytics repository) for YOLOv5 model source code,
- NVIDIA CUDA Toolkit and cuDNN libraries to enable GPU acceleration, and
- PyTorch as the deep learning framework backend.

The choice of these hardware and software tools ensured efficient training within the constraints of a mid-range GPU system.

2.4. Model training

The YOLOv5s model was chosen due to its balance between accuracy and computational efficiency, making it suitable for small object detection tasks such as logic gate recognition. The training setup was defined as:

- Input resolution: 256×256 pixels.
- Batch size and epochs: a batch size of 32 was used, with training conducted for 100 epochs. Each epoch represented a complete pass through the training set, with images shuffled at every iteration to improve generalization.
- Pre-trained weights: the model used yolov5s.pt pre-trained weights as initialization, leveraging transfer learning to accelerate convergence and improve accuracy with limited training data.
- Loss functions: YOLOv5 applies three main losses: box loss (for bounding box regression), objectness loss (for detecting the presence of objects), and classification loss (for distinguishing between logic gate types).
- Optimizer and learning rate: the default SGD optimizer with momentum was used, with an adaptive learning rate schedule as defined in the YOLOv5 configuration.
- Auto-anchors: automatic recalculation of anchor boxes was enabled to optimize detection of small, structured objects such as logic gate symbols.

2.5. Evaluation metrics

To evaluate model performance, the following metrics were used:

- Precision (P): the proportion of correctly predicted positive detections among all detections.
- Recall (R): the proportion of correctly detected objects among all ground-truth objects.
- mAP@0.5: mAP at IoU threshold 0.5, the standard metric for object detection performance.
- mAP@0.5:0.95: mAP across multiple IoU thresholds (0.5 to 0.95), providing a more stringent evaluation. These metrics were calculated separately for training, validation, and testing datasets.

2.6. System implementation

Once the model was trained, it was integrated into a translation system capable of detecting logic gates and input symbols from images and converting them into Boolean functions. The system workflow included:

- Image input: users provide images via a graphical interface. To ensure consistent processing, the system currently accepts a maximum of three input images. This constraint allows the first and second images to represent parallel inputs, while the third image serves as the operator, enabling the construction of valid binary logic expressions.
- Object detection: the trained YOLOv5 model identifies logic gates and input symbols within the image.
- Symbol mapping: detected objects are mapped to their corresponding Boolean symbols (e.g., AND→"·", OR→"+", NOT→"¬").
- Expression construction: the system parses the detected components to construct a valid Boolean expression.

This system bridges the gap between visual representations of circuits and their symbolic Boolean equivalents, demonstrating practical applicability of AI-based detection in digital electronics.

3. RESULTS AND DISCUSSION

3.1. Training process overview

The training process of the YOLOv5s model is recorded in the graphs shown in Figure 1. Based on Figure 1, the results indicate that the YOLOv5s model did not experience overfitting during the training phase. The model began to plateau or reduce its learning rate after approximately 30 epochs until the end of training.

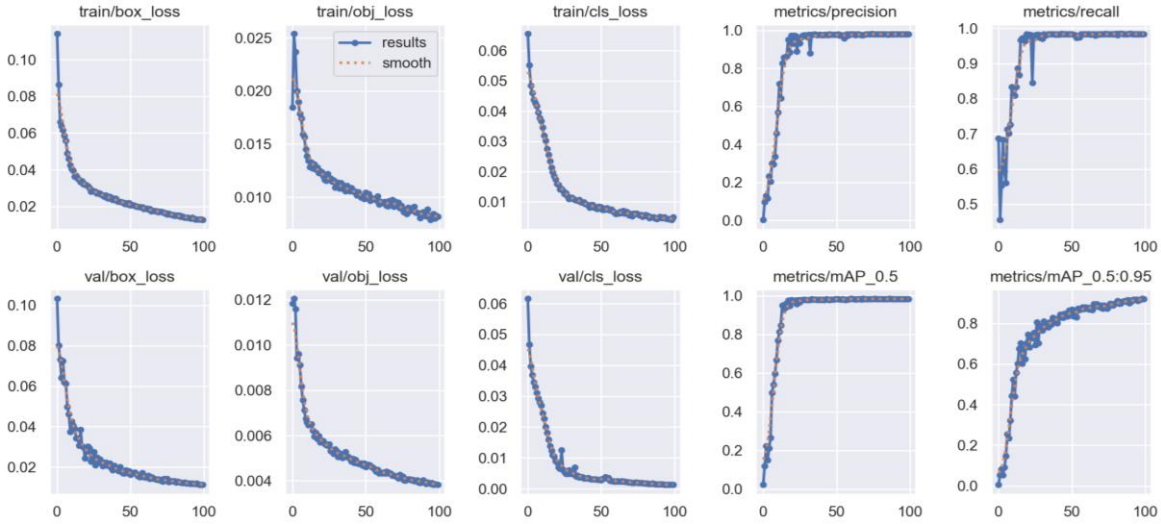


Figure 1. Training results of the YOLOv5s model

3.2. Validation process overview

The validation results of the YOLOv5s training process are presented in Table 1. According to Table 1, the YOLOv5s model achieved high precision and recall values, which indicates that the model was able to correctly detect and predict all logic gate objects and their inputs within the images. For the mAP at IoU threshold 0.5 (mAP@50), the model achieved values above 0.9 across all classes, demonstrating its ability to accurately detect objects with IoU ≥0.5 in all images. Meanwhile, for mAP at IoU thresholds ranging from 0.5 to 0.95 (mAP@50–95), the model obtained values above 0.9 for most classes, except for AND, Input 1, Input 2, and XOR. These lower mAP@50–95 results suggest that the model still struggles with precise bounding box placement for certain classes, although its performance is still considered satisfactory.

In addition, the validation results are also presented in the form of a confusion matrix, as shown in Figure 2. From Figure 2, it can be observed that the model occasionally misclassified Input 1, Input 2, and XOR as background. Moreover, some regions that were actually background were misclassified by the model as XOR, NOT, Input 1, or Input 2. However, aside from these cases, the model demonstrated perfect detection performance for the other classes.

Table 1. The performance of YOLOv5s on the validation set

Class	Images	Instances	Precision (P)	Recall (R)	mAP@50	mAP@50-95
All	160	420	0.98	0.984	0.982	0.922
AND	160	20	0.995	1	0.995	0.899
Input 1	160	140	0.964	0.958	0.98	0.83
Input 2	160	140	0.953	0.95	0.964	0.812
NAND	160	20	0.993	1	0.995	0.98
NOR	160	20	0.991	1	0.995	0.949
NOT	160	20	0.995	1	0.995	0.969
OR	160	20	0.992	1	0.995	0.987
XNOR	160	20	0.993	1	0.995	0.982
XOR	160	20	0.944	0.95	0.926	0.892

3.3. System translator results

The translation system was tested using a total of 24 cases, five of which are presented in Table 2. Based on the results in Table 2, it can be observed that the YOLOv5s model successfully detected and predicted all objects correctly. The translations produced by the system also matched the expected outputs.

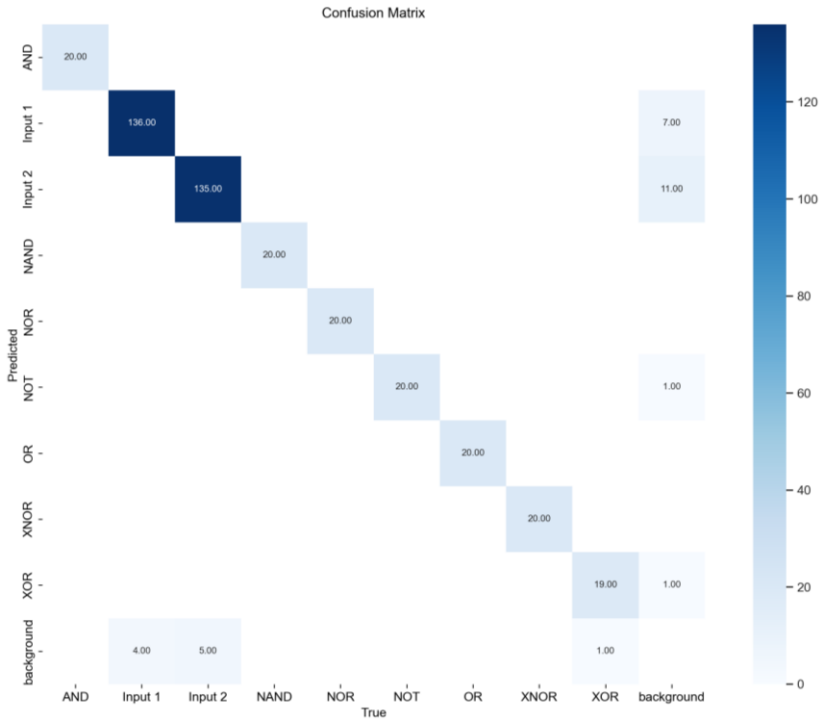
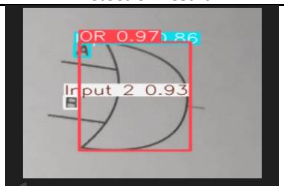
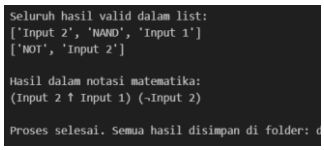


Figure 2. Confusion matrix results of the YOLOv5s model

Table 2. The performance of the translation system

Test case name	Detection result	Translation result
1 input object OR		
1 input object Input 1		
1 input object non-logic gate		
2 inputs: 1 logic gate object and 1 non-logic object		
3 inputs: 2 logic gate objects and 1 non-logic object		

4. CONCLUSION

This research successfully designed and implemented a system for detecting and translating logic gate circuit images into Boolean functions using YOLOv5. The custom dataset of 800 images ensured robustness under varied conditions. The YOLOv5s model achieved strong results (mAP@0.5=0.922, precision=0.980, recall=0.984), with confusion matrix analysis confirming reliability. The translator system accurately generated Boolean expressions from detected components, demonstrating feasibility of bridging image recognition and symbolic computation. Future work includes extending the dataset to more complex multi-gate circuits and integrating simplification algorithms for Boolean minimization.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

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

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Muhammad Shidqii Taqiyyuddin	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	
Muhammad Subali		✓							✓	✓	✓	✓		

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

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available at: https://universe.roboflow.com/muhammad-shidqii-taqiyyuddin/logic-gate-t1fli

REFERENCES

[1] R. Szeliski, *Computer vision: algorithms and applications*. Cham: Springer International Publishing, 2022, doi: 10.1007/978-3-030-34372-9.
[2] A. K. Maini, *Digital electronics: principles, devices and applications*. Chichester, U.K.: John Wiley & Sons Ltd., 2007.
[3] M. M. Mano and M. D. Ciletti, *Digital design*, 4th ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2007.
[4] R. Datta, P. D. S. Mandal, and B. Chanda, "Detection and identification of logic gates from document images using mathematical morphology," in *2015 5th National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, NCVPRIPG 2015*, 2015, pp. 1–4, doi: 10.1109/NCVPRIPG.2015.7490040.
[5] D. Y. Yun, S. K. Seo, U. Zahid, and C. J. Lee, "Deep neural network for automatic image recognition of engineering diagrams," *Applied Sciences*, vol. 10, no. 11, p. 4005, Jun. 2020, doi: 10.3390/app10114005.
[6] C. Aviles-Cruz, J. Villegas-Cortez, A. Zuniga-Lopez, I. Osuna-Galan, Y. Perez-Pimentel, and S. Cordero-Sanchez, "Logic gate integrated circuit identification through augmented reality and a smartphone," in *Intelligent Computing-Proceedings of the Computing Conference*, 2019, pp. 73–85, doi: 10.1007/978-3-030-22871-2_6.
[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
[8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Apr. 2020, doi: 10.48550/arXiv.2004.10934.
[9] Ultralytics, "YOLOv5." *GitHub*. Accessed Jun. 02, 2025. [Online]. Available: https://github.com/ultralytics/yolov5
[10] C. Li, W. Pan, R. K. L. Su, and P. C. Yuen, "Multiple structural defect detection for reinforced concrete buildings using YOLOv5s," *HKIE Transactions Hong Kong Institution of Engineers*, vol. 29, no. 2, pp. 141–150, Jun. 2022, doi: 10.33430/V29N2THIE-2021-0033.
[11] X. Li, C. Wang, H. Ju, and Z. Li, "Surface defect detection model for aero-engine components based on improved YOLOv5," *Applied Sciences*, vol. 12, no. 14, p. 7235, Jul. 2022, doi: 10.3390/app12147235.

[12] S. Puliti and R. Astrup, "Automatic detection of snow breakage at single tree level using YOLOv5 applied to UAV imagery," *International Journal of Applied Earth Observation and Geoinformation*, vol. 112, p. 102946, Aug. 2022, doi: 10.1016/j.jag.2022.102946.
[13] S. Zhang, Y. Chang, S. Wang, Y. Li, and T. Gu, "An improved lightweight YOLOv5 algorithm for detecting railway catenary hanging string," *IEEE Access*, vol. 11, pp. 114061–114070, 2023, doi: 10.1109/ACCESS.2023.3322444.
[14] Z. Lu, L. Ding, Z. Wang, L. Dong, and Z. Guo, "Road condition detection based on deep learning YOLOv5 network," in *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information, ICETCI 2023*, May 2023, pp. 497–501, doi: 10.1109/ICETCI57876.2023.10176545.
[15] F. Wu, G. Jin, M. Gao, Z. He, and Y. Yang, "Helmet detection based on improved YOLO V3 deep model," in *Proceedings of the 2019 IEEE 16th International Conference on Networking, Sensing and Control, ICNSC 2019*, May 2019, pp. 363–368, doi: 10.1109/ICNSC.2019.8743246.
[16] F. Zhou, H. Zhao, and Z. Nie, "Safety helmet detection based on YOLOv5," in *Proceedings of 2021 IEEE International Conference on Power Electronics, Computer Applications, ICPECA 2021*, Jan. 2021, pp. 6–11, doi: 10.1109/ICPECA51329.2021.9362711.
[17] T. Liu, C. Dongye, and X. Jia, "The research on traffic sign recognition algorithm based on improved YOLOv5 model," in *2023 3rd International Conference on Consumer Electronics and Computer Engineering, ICCECE 2023*, Jan. 2023, pp. 92–97, doi: 10.1109/ICCECE58074.2023.10135475.
[18] H. Zhang, X. Zhou, H. Li, G. Zhu, and H. Li, "Machine recognition of map point symbols based on YOLOv3 and automatic configuration associated with POI," *ISPRS International Journal of Geo-Information*, vol. 11, no. 11, p. 540, Oct. 2022, doi: 10.3390/ijgi11110540.
[19] P. Puri, D. Kumar, and V. Kukreja, "Enhanced detection of wheat mosaic virus using YOLOV5 model with adaptive thresholding," in *2023 4th International Conference for Emerging Technology, INCET 2023*, May 2023, pp. 1–6, doi: 10.1109/INCET57972.2023.10170635.
[20] J. Liu and X. Wang, "Tomato diseases and pests detection based on improved Yolo V3 convolutional neural network," *Frontiers in Plant Science*, vol. 11, p. 898, Jun. 2020, doi: 10.3389/fpls.2020.00898.
[21] B. Wang, Y. Yan, Y. Lan, M. Wang, and Z. Bian, "Accurate detection and precision spraying of corn and weeds using the improved YOLOv5 model," *IEEE Access*, vol. 11, pp. 29868–29882, 2023, doi: 10.1109/ACCESS.2023.3258439.
[22] J. Bayer, S. H. Turabi, and A. Dengel, "Text extraction for handwritten circuit diagram images," in *International Conference on Document Analysis and Recognition (ICDAR 2023)*, 2023, pp. 192–198, doi: 10.1007/978-3-031-41498-5_14.
[23] H. Xin, Z. Chen, and B. Wang, "PCB electronic component defect detection method based on improved YOLOv4 algorithm," in *Journal of Physics: Conference Series*, vol. 1827, Mar. 2021, p. 012167, doi: 10.1088/1742-6596/1827/1/012167.
[24] R. R. Rachala and M. R. Panicker, "Hand-drawn electrical circuit recognition using object detection and node recognition," *SN Computer Science*, vol. 3, no. 3, p. 244, May 2022, doi: 10.1007/s42979-022-01159-0.
[25] T. F. Dima and M. E. Ahmed, "Using YOLOv5 algorithm to detect and recognize American sign language," in *2021 International Conference on Information Technology, ICIT 2021-Proceedings*, Jul. 2021, pp. 603–607, doi: 10.1109/ICIT52682.2021.9491672.

BIOGRAPHIES OF AUTHORS

Muhammad Shidqii Taqiyyuddin received the Bachelor of Engineering in Informatics Engineering from Gunadarma University, Indonesia, in 2021. His research interests include artificial intelligence, web development, and computer networking. He can be contacted at email: taqiyy17@gmail.com.

Muhammad Subali received his doctoral degree in 2007. He is currently the Head of the Lecturers at the Faculty of Informatics Engineering, Cendekia Abditama University, Tangerang, Banten, Indonesia. His research interests include image processing, speech signal processing, and digital speech processing. He can be contacted at email: subali@uca.ac.id.