



Embedded System

Audio-Based Assistive Device for Boxed Product Detection for the Visually Impaired

Muhammad Dzaky Yoni ¹, Dodon Yendri ^{*2}

^{1,2} Department of Computer Engineering, Faculty of Information Technology, Andalas University, Limau Manis Kec. Pauh, Padang City, West Sumatra, Indonesia

ARTICLE INFORMATION

Received: April 10, 2025

Revised: April 25, 2025

Available online: April 30, 2025

KEYWORDS

Object Detection, Visually Impaired, SSD
Mobilenetv2, Audio, Raspberry Pi, Ultrasonik Sensor

CORRESPONDENCE

E-mail: dodon@it.unand.ac.id

ABSTRACT

Visually impaired individuals face challenges in recognizing and distinguishing boxed products, which are typically identified visually. This study aims to develop an audio-based assistive system for detecting boxed product packaging to enhance the independence of visually impaired individuals. The system utilizes the *Single Shot Multibox Detector (SSD) MobileNetV2* algorithm for object detection and classification, with the Raspberry Pi 4 Model B as the main microcontroller and the Raspberry Pi Camera OV5647 for image capture. The HC-SR04 ultrasonic sensor is used to measure the distance of objects from the user, while the output information is provided through *wireless* earphones in the form of audio. Experimental results show that the system can detect and classify boxed products with an accuracy of 95%, precision of 96.8%, recall of 98%, and an F1-Score of 97.4%. This system provides real-time audio feedback, assisting visually impaired individuals in identifying products without the need for tactile exploration.

INTRODUCTION

Visual impairment can limit human activities in daily life. According to data from the World Health Organization (WHO), the number of people in the world with visual impairment is estimated at 253 million. Of these, 188 million people have mild visual impairment, 217 million people have moderate to severe visual impairment, and 36 million people are blind [1]. Limitations in vision bring significant challenges for visually impaired people in carrying out daily tasks, especially in recognizing and distinguishing products packaged in boxes.

Visual identification is the main source of information in recognizing various items, but it is not accessible to blind people. Therefore, an assistive device is needed that can provide effective assistance through audio-based detection of boxed products to increase their independence and mobility in daily life. The development of information and communication technology provides a strong foundation for designing innovative solutions for the visually impaired. Audio-based systems emerge as a promising alternative in providing information audibly, thus enabling visually impaired people to recognize and distinguish boxed products needed in daily activities, such as foodstuffs, beverages, and other consumer goods.

Several studies have been conducted in the field of object recognition to detect product packaging. One study developed a text detection system on food product packaging using a Raspberry Pi-based deep learning method that produces output in the form of text or text-to-speech through speakers. However, this research still has limitations, namely some products cannot be detected properly, where 3 out of 10 products tested did not get any detection results at all [2].

Another study used the You Only Look Once (YOLO) algorithm to detect Palembang specialties with an accuracy above 90%. However, this research has weaknesses because the detected objects do not have packaging and YOLOv3 still has difficulty in recognizing objects with medium to large sizes [3]. Furthermore, another study used YOLOv4-tiny to detect Palembang specialties with a mean average precision (mAP) value of 92.14%. However, this research is limited to one type of product, namely milk powder, despite different brands [4].

Based on some of these studies, there are several limitations, such as systems that have not been specifically designed for blind people, limited object variations, and there are still products that are not detected properly. Therefore, it is necessary to develop a system that is more flexible and in accordance with the needs of blind people.

In this research, an audio-based tool for boxed product detection for blind people is developed. This system uses Raspberry Pi 4 Model B as a microprocessor that processes images from Raspberry Camera OV5647 and detects object distance using HC-SR04 ultrasonic sensor. This system is designed to provide information through audio technology that can be heard through wireless earphones, making it easier for blind people to recognize box packaging products without having to touch them first. In addition, this system applies the Single Shot Multibox Detector (SSD) algorithm with MobileNetV2 backbone. This algorithm has an advantage over YOLO because it offers faster inference time and has a more efficient network structure. With this approach, it is expected that the developed system can increase the independence of blind people in recognizing various box packaging products more easily and accurately.

METHOD

This research utilizes action research methodology, in which the researcher works together with the research subject to identify problems, design solutions, implement actions, and reflectively evaluate the results.

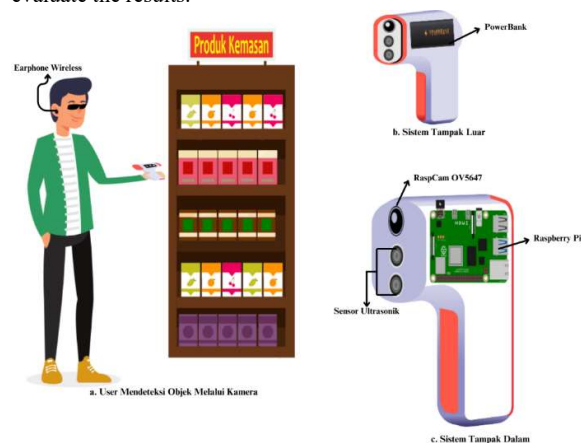


Figure 1. General System Design

Figure 1 shows a user holding a thermogun-like device powered by a power bank to detect boxed products, such as wafers, milk powder, liquid milk, whole grain cereals, noodles, sugar, ground coffee, and tea bags. The device is equipped with a Raspberry Pi Camera OV5647 that records real-time video. Each frame is processed by the Raspberry Pi using the Single Shot Multibox Detector (SSD) algorithm with Python as the programming language. The input image is extracted through a convolution layer and classified according to the available dataset. The classification results are then conveyed in the form of audio, such as the greeting "Teabag Detected," which is played through the user's wireless earphones.

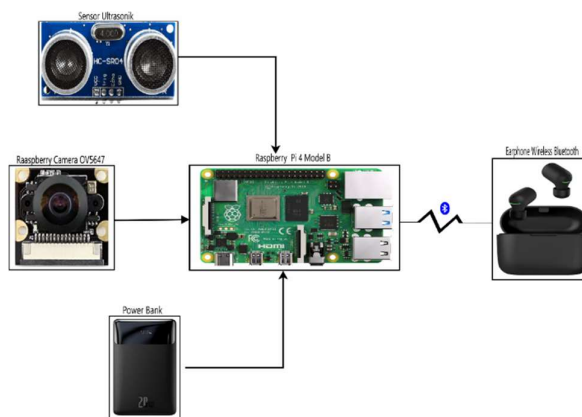


Figure 2. Schematic Design

In Figure 2, it can be seen that this system consists of several main components, namely Raspberry Pi as a microprocessor that manages image processing to audio output through wireless earphones. The Raspberry Pi Camera OV5647 is used to capture the image of the object as input, while the ultrasonic sensor measures the distance of the object to the camera within a certain time. The detection information is converted into sound output through wireless earphones. The entire system is supported by a power bank as a power source to ensure stable operation.

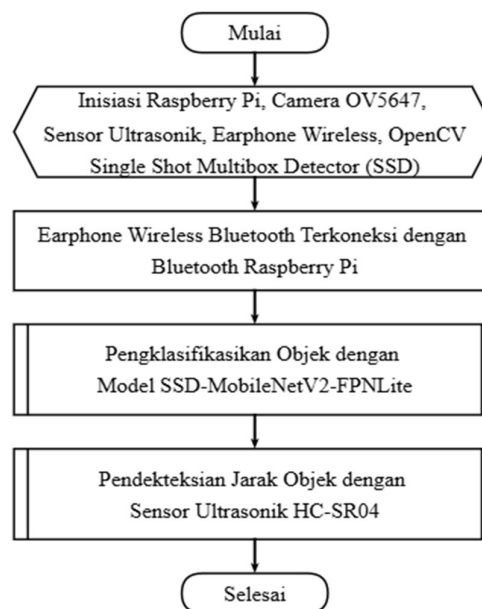


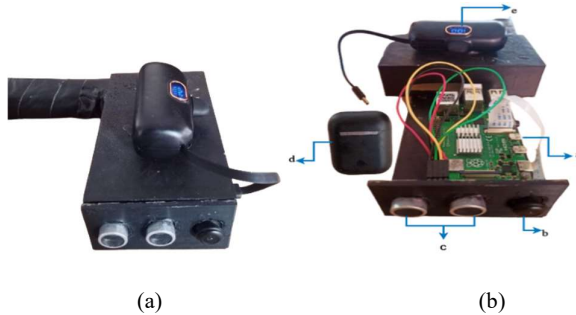
Figure 3. Flowchart of Process Design

Based on Figure 3, it is explained that the system begins with the initialization of components, including Raspberry Pi, OV5647 camera, wireless earphones, and Single Shot Multibox Detector (SSD) algorithm. The camera captures images in real-time for classification using SSD. The classification results are compared with the dataset to identify boxed products. The product type and distance information is conveyed through the wireless earphone. The process is iterative to detect the next object.

RESULTS AND DISCUSSION

Hardware Implementation

The hardware implemented in this system are Raspberry Pi 4B, Raspberry Pi Camera OV5647, Ultrasonic Sensor, Wireless Earphone, and Power Bank. Some of these components are arranged in a container that has been made as shown below,



Gambar 4. Whole System (a) External View, (b) Internal View

In **Figure 4(b)** there is an explanation of the components according to the number, namely,

- Raspberry Pi 4B, installed inside the case so that it can be integrated with two input components, namely the Raspberry Pi Camera and ultrasonic sensor.
- Raspberry Pi Camera OV5647, installed at the front of the system to facilitate object detection without being obstructed by other hardware components.
- HC-SR04 Ultrasonic Sensor, installed parallel to the Raspberry Pi Camera in order to measure the distance accurately, according to the position of the camera against the detected object.
- PowerBank, with a battery capacity of 5000mAh installed on the side of the case and provides output power of 5V 2A.
- Wireless Earphone, functions as an audio output connected to the Raspberry Pi via Bluetooth.
- PowerBank, with a battery capacity of 5000mAh is installed on the side of the case and provides 5V 2A output power.

Software Implementation

The software implementation in this system is the application of the SSD-MobileNetV2 model according to the predetermined class and implements the software so that the hardware can be connected properly to the Raspberry Pi.

- Dataset collection, taken by custom either in the form of images captured by mobile phones or from videos converted into .jpg format. The results of data collection amounted to 2000 images of box packaging.
- Dataset Preposition, is the preparation of data before processing. Some processes such as resizing images to 800x600px size, dividing training and test data with a ratio of 80:20, labeling images, and converting data into TFRecord format which is used for dataset input during the training process. The following is the amount of data per each class divided into 80:20 percent.

Table 1. Dataset Splitting

Class	Train	Validation
Wafers	200	50
Milk Powder	200	50
Liquid Milk	200	50
Wheat Cereals	200	50
Noodles	200	50
Sugar	200	50
Ground Coffee	200	50
Teabags	200	50
Total	1600	400

- Dataset Training, stage is the process of training the model using a pre-processed dataset (transfer learning). The model used is SSD-MobileNetV2 FPNLite 320x320, which is optimized for object detection on devices with computational limitations such as Raspberry Pi. This stage starts from the configuration of pipeline.config, model evaluation and up to the conversion of the model to TFLite Lite (.tflite) so that it can be used on Raspberry Pi for real-time inference.

Testing and Analysis

Testing and Analysis of Raspberry Pi Camera OV5647 Against Distance and Processing Time

Table 2. Distance and Processing Time during Object Detection

Class	Distance (cm)	Description	Runtime (ms)
Sugar	10	Not detected	-
	20	Detected	159.9
	30	Detected	165.5
	40	Detected	168.4
	50	Not detected	-
Ground Coffee	10	Not detected	-
	20	Detected	164.5
	30	Detected	159.1
	40	Not detected	-
	50	Not detected	-
Noodles	10	Detected	156.2
	20	Detected	154.1
	30	Detected	166.5
	40	Not detected	-
	50	Not detected	-
	10	Detected	163.9

Class	Distance (cm)	Description	Runtime (ms)
Milk Powder	20	Detected	158.0
	30	Detected	162.0
	40	Not detected	-
	50	Not detected	-
Liquid Milk	10	Not detected	-
	20	Detected	175.3
	30	Detected	164.2
	40	Not detected	-
Wheat Cereals	10	Not detected	-
	20	Detected	168.8
	30	Detected	160.4
	40	Detected	153.0
Teabags	10	Detected	172.2
	20	Detected	169.5
	30	Detected	160.9
	40	Detected	157.4
Wafers	10	Detected	162.7
	20	Detected	187.8
	30	Detected	154.1
	40	Not detected	-
	50	Not detected	-

Object detection testing using the raspberry pi camera OV5647 starts from a distance of 10 - 50 cm. After testing, the best detection occurs at a distance of 20-30 cm, with the most accurate Wheat Cereal and Teabag, while some objects are difficult to detect above 30 cm. Testing of the Raspberry Pi Camera OV5647 showed an average prediction time of 94.11 ms. At longer distances, some data was not recorded.

Model Testing and Analysis of Light Intensity

Table 3. Testing Box Packaging Detection against Lightning Intensity

Light Intensity (Lux)	Detection Result	Description
0		Objects Not Detected

Light Intensity (Lux)	Detection Result	Description
10		Some Objects Detected and Some Misdetected
20		Some Objects Detected and Not Misdetected
50		All Objects Detected

Table 3 shows the results of testing object classification and detection under various lighting conditions, with lux values varying from 0 to 50. Based on the test results, a stable lighting condition for object detection starts from lux 50, because at this point all objects have been detected according to their class. In addition, no misclassification or missing objects were found, so the system can work with optimal accuracy under these lighting conditions.

Testing and Analysis of Object Detection Data Delivery

Table 4. FPS Testing on Object Detection System

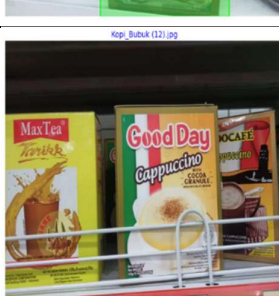
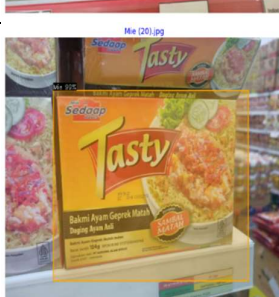
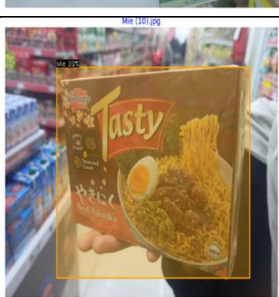
Model	Detection Result	FPS Value
Without Model		19.6
		19.4
SSD-Mobile NetV2-FPNLite 320x320		7
		6.4

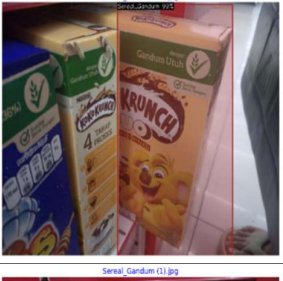
Table 4 shows the FPS test results based on the object detection system components. When only displaying camera frames without detection, the FPS is high (19.1-19.6). After the MobileNetV2 FPNLite SSD model was implemented, the FPS decreased due to the computational load. The addition of sound notifications slowed down the FPS further (6.4-7) due to the lag when audio was played. Despite this, the system continued to perform well, with the main focus on detection accuracy and voice notifications being useful to the user.

Testing and Analysis of Object Classification Based on Test Data

Table 5. Test Results of Object Classification with Test Data

True Class	Detection Result	Predicted Class	Description
Sugar		No Prediction	False Negative

True Class	Detection Result	Predicted Class	Description
Sugar		Sugar	True Positive
Ground Coffee		Ground Coffee	True Positive
Ground Coffee		No Prediction	False Negative
Noodles		Noodles	True Positive
Noodles		Noodles	True Positive
Milk Powder		Milk Powder	True Positive

True Class	Detection Result	Predicted Class	Description
Milk Powder		Milk Powder	True Positive
Liquid Milk		Liquid Milk	True Positive
Liquid Milk		Liquid Milk	True Positive
Wheat Cereal		Wheat Cereal	True Positive
Wheat Cereal		Wheat Cereal	True Positive
Teabag		Teabag	True Positive

True Class	Detection Result	Predicted Class	Description
Teabag		Teabag	True Positive
Wafer		No Prediction	False Negative
Wafer		Liquid Milk	False Positive

Table 6. Model Performance Calculation

Class	Accuracy	Precision	Recall	F1-Score
Sugar	0.95	1.00	0.95	0.974
Ground Coffee	0.85	0.895	0.944	0.919
Noodles	1.00	1.00	1.00	1.00
Wheat Cereals	0.95	0.95	1.00	0.974
Milk Powder	1.00	1.00	1.00	1.00
Liquid Milk	1.00	1.00	1.00	1.00
Teabags	0.95	0.95	0.95	0.974
Wafers	0.90	0.947	0.947	0.947

Tests were conducted using 160 samples of boxed product images, with each class totaling 20 samples. Evaluation of the model's performance was performed by calculating the accuracy, precision, recall, and F1-score metrics for each class, the results of which can be seen in **Table 6**. The calculations were performed using the confusion matrix, which allows in-depth analysis of the model's performance in correctly classifying each product category.

Based on the test results, the model showed high performance with an average accuracy of 95%, precision of 96.8%, recall of 98%, and F1-score of 97.4%. These values indicate that the model is able to recognize and differentiate products with a low error rate. The high accuracy indicates that most of the predictions

generated by the model match the actual labels. Meanwhile, the precision value of 96.8% indicates that the model rarely gives wrong predictions for certain classes. Recall of 98% indicates that almost all samples that should have been detected were correctly classified by the model. In addition, the F1-score value of 97.4% shows a good balance between precision and recall, so the model has a reliable performance in detecting and recognizing boxed packaging products optimally.

CONCLUSION

Based on the results of implementation, testing and analysis carried out on this system, it can be concluded that:

1. This system can detect and classify box packaging objects with 8 different classes using RaspCam OV5647 and SSD-MobileNetV2 FPNLite Model in real-time with the accuracy achieved is 95%, precision 96.8%, recall 98%, and F1-Score 97.4%.
2. The system can detect and classify objects optimally at a distance of 20-30 cm with lighting starting from 50 lux.

RECOMMENDATION

Based on the results of implementation, testing and analysis carried out on this system, there are suggestions for this system to perform better, namely:

1. Adding variations to the Ground Coffee class because it still has a large false positive from other classes.
2. Replace the newer Raspi Camera which has a higher resolution, so that the object detection results are more satisfactory.
3. Adding an On/Off Switch to easily control the system for users.
4. Adding the types of objects to be detected, so as to expand the capabilities of the system.
5. Replace the newer Raspberry pi version that supports 8GB RAM and above, so that it is lighter for the object detection process.

REFERENCE

- [1] Kementerian Kesehatan Republik Indonesia. (2018). Infodatin: Gangguan penglihatan. Jakarta: Pusat Data dan Informasi Kementerian Kesehatan Republik Indonesia.
- [2] Yogi, M., Sari, M. I., & Handayani, R. (2023). Sistem Deteksi Teks Pada Kemasan Produk Makanan Menggunakan Metode Deep Learning Berbasis Raspberry Pi. *eProceedings of Applied Science*, 10(4).
- [3] Rahma, L., Syaputra, H., Mirza, A. H., & Purnamasari, S. D. (2021). Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once). *Jurnal Nasional Ilmu Komputer*, 2(3), 213-232.
- [4] Saputra, R. D., & Fudholi, D. H. (2023). Model Mobile untuk Deteksi Objek pada On-Shelf Availability Produk Retail. *AUTOMATA*, 4(2).
- [5] Hidayat, Asep AS, Suwandi, Ate. (2013). Pendidikan anak berkebutuhan khusus tunanetra peserta didik berkebutuhan khusus dengan hambatan penglihatan dalam perspektif pendidikan dan layanan (Cet. Ke-1). Jakarta: Luxima.
- [6] Pramana, E. (2023, Februari 11). Bantu Tunanetra, Ilmuwan Ciptakan Tongkat dengan Teknologi AI. Diambil kembali dari JawaPos.com:

- <https://www.jawapos.com/teknologi/01435889/bantu-tunanetra-ilmuwan-ciptakan-tongkat-dengan-teknologi-ai>
- [7] Klimchuk, M. R., & Krasovec, S. A. (2007). *Desain kemasan: Perencanaan merek produk yang berhasil mulai dari konsep sampai penjualan*. Penerbit Erlangga.
- [8] Nilsson, N. J. (1998). *Artificial intelligence: a new synthesis*. Morgan Kaufmann.
- [9] Yansens, C. *Apa Itu Artificial Intelligence (AI)? Cara Kerja, Komponen, Jenis, Tujuan*. Diambil kembali dari Tanya Digital: <https://tanyadigital.com/artificial-intelligence-adalah/>
- [10] Coelho, L. P., & Richert, W. (2015). *Building machine learning systems with Python*. Packt Publishing Ltd
- [11] Ahmad, A. (2017). *Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning*. Yayasan Cahaya Islam, Jurnal Teknologi Indonesia
- [12] Batubara, N. A., Awangga, R. M., & Pane, S. F. (2020). Perbandingan Faster R-CNN dengan SSD Mobilenet Untuk Mendeteksi Plat Nomor, Vol.1. Kreatif Industri Nusantara, 9-10.
- [13] Le, H. H. (2015). Division Intro. Retrieved from Academia: https://www.academia.edu/27143145/Division_intro
- [14] Primartha, R. (2018). Belajar Machine Learning Teori dan Praktik. Bandung: Informatika Bandung, 10, 20-30.
- [15] Wahyu, R. (2019, Juli 6). Deep Learning Object Detection Menggunakan Tensorflow Dan Metode Convolutional Neural Network (Cnn). Diambil kembali dari medium.com: <https://medium.com/@rismitawahyu/deep-learning-object-detection-menggunakan-tensorflow-dan-metode-convolutional-neural-network-cnn-9c09267474e3>
- [16] Saha, S. (2018, Desember 15). A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. Diambil kembali dari SaturnCloud: <https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/>
- [17] Zufar, M., & Setiyono, B. (2016). Convolutional neural networks untuk pengenalan wajah secara real-time. *Jurnal Sains Dan Seni Its Vol. 5 No. 2* (2016) 2337-3520 (2301-928X Print), A72-A77.
- [18] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* (pp. 21-37). Springer International Publishing.
- [19] C. Liu, Wei; Anguelov, Dragomir; Erhan, Dumitru; Szegedy, Christian; Reed, Scott; Fu, Cheng Yang; Berg, “SSD: Single Shot MultiBox Detector,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9905 LNCS, pp. 1–17, 2016, doi: 10.1007/978-3-319-46448-0_2.
- [20] Ramalingam, B., Elara Mohan, R., Balakrishnan, S., Elangovan, K., Félix Gómez, B., Pathmakumar, T., ... & Baskar, C. (2021). Stetro-deep learning powered staircase cleaning and maintenance reconfigurable robot. *Sensors*, 21(18), 6279.
- [21] Badgujar, W. b. (2021, Juni 17). *Creating MobileNets with TensorFlow from scratch*. Diambil kembali dari medium.com:

- <https://medium.com/@sumeetbadgujar/creating-mobilenets-with-tensorflow-from-scratch-c34ec79a59d2>
- [22] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- [23] Team, P. (t.thn.). *MOBILENET V2*. Diambil kembali dari pytorch.org: https://pytorch.org/hub/pytorch_vision_mobilenet_v2/
- [24] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).
- [25] Tryatmojo, B., & Maryati, R. I. S. (2019). Akurasi Sistem Face Recognition OpenCV Menggunakan Raspberry Pi dengan Metode Haar Cascade. *Jurnal Ilmiah Informatika*, 7(02), 92-98.
- [26] File:OpenCV Logo with text.png. (2022, Februari). Diambil kembali dari WikimediaCommons:https://commons.wikimedia.org/wiki/File:OpenCV_Logo_with_text.png
- [27] Pattanayak, S., Pattanayak, J. S., & John, S. (2017). *Pro deep learning with tensorflow* (pp. 153-278). New York, NY, USA: Apress.
- [28] Santra, B. K. (2012). Genetic Algorithm and Confusion Matrix for Document Clustering. *International Journal of Computer Science Issues*.
- [29] Downey, A. (2012). *Think python*. "O'Reilly Media, Inc."
- [30] Wikimedia Commons. (2023, Februari). Diambil kembali dari File:Python-logo-notext.svg: <https://commons.wikimedia.org/wiki/File:Python-logo-notext.svg>
- [31] Soen, G. I. E., Marlina, M., & Renny, R. (2022). Implementasi Cloud Computing dengan Google Colaboratory pada Aplikasi Pengolah Data Zoom Participants. *JITU: Journal Informatic Technology And Communication*, 6(1), 24-30.
- [32] Johnston, B. (2020, Oktober 1). *Google Colab*. Diambil kembali dari Princeton.edu: <https://mcgrawect.princeton.edu/tool/google-colab/>
- [33] Fahri, U., Pd, S., & TI, M. (2016). Raspberry Pi. *Akademi Manajemen Komputer Dan Informatika: Halaman*, 24.
- [34] Australia, I. (2019, Agustus 22). Raspberry Pi 4 Model B. Diambil kembali dari indomedia: <https://indomedia.com.au/raspberry-pi-4-model-b/>
- [35] Pi, R. Raspberry Pi 4 Tech Specs. Diambil kembali dari Raspberry Pi: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>
- [36] Lazada. Raspberry Pi 3/4 Generation OV5647 Camera 130 Degree Fisheye Wide-angle Camera Module 3B+ 5 Million Pixels. Diambil kembali dari Lazada:<https://www.lazada.co.id/products/raspberry-pi-34-generation-ov5647-camera-130-degree-fisheye-wide-angle-camera-module-3b-5-million-pixels-i6370208039s12071706955.html?spm=a2o4j.tm80150940.3312045370.1416dD6SSD6SSY4.416dD6SSD6SSY4>
- [37] Uctronics. (T.Thn.). 5mp Ov5647: Download Full Datasheet (Specs, Pinouts, Registers & Diagrams). Diambil kembali dari uctronics.com: <https://www.uctronics.com/5mp-ov5647-download-full-datasheet-pdf>
- [38] Sakti, E. (n.d.). Cara Kerja Sensor Ultrasonik, Rangkaian, & Aplikasinya. Diambil kembali dari ElangSakti.com: <https://www.elangsakti.com/2015/05/sensor-ultrasonik.html>
- [39] Widharma, I. G. S., Saputra, K. D. W., Purnamayana, I. M. S., Anam, S., & Prawiradnyana, I. G. P. B. (2020). Sensor Ultrasonik dalam Water Level Controller. *no. December*, 1-11.
- [40] Carlson, D. (2017). Wireless Charging for Bluetooth Earphones.

BIOGRAFI PENULIS



Muhammad Dzaky Yoni

Lahir pada 15 Maret 2001 di Padang, Sumatera Barat. Sekarang sedang menjalani pendidikan kuliah di Departemen Teknik Komputer, Fakultas Teknologi Informasi, Universitas Andalas.