Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 8 No. 1 (2024) 26 - 32 e-ISSN: 2580-0760

Visual Impaired Assistance for Object and Distance Detection Using Convolutional Neural Networks

Jumadi Mabe Parenreng[1], Andi Baso Kaswar[2], Ibnu Fikrie Syahputra[3]
[1,2,3]Informatic Engineering and Computer Education, State University of Makassar, Makassar, South of Sulawesi, Indonesia
[1]jparenreng@unm.ac.id, [2]a.baso.kaswar@unm.ac.id, [3]ibnufikriesyah2000@gmail.com

Abstract

Vision is a very valuable gift from God, most aspects of human needs in the body are dominated by vision. Based on data from the World Health Organization (WHO) there are around 180 million people in the world experiencing visual impairment, while the prevalence of blindness in Indonesia reaches 3 million people (1.5% of Indonesia's population), so we designed a system in the form of a prototype that could detect objects around the user and convey data in the form of sound to the user. This research discusses the application of a machine learning model using the Convolutional Neural Network method so that it can detect objects optimally. The objects that have been collected will be trained on machine learning and produce a model to be embedded in the system's main machine, namely the Raspberry PI 4B. Machine learning model training was carried out several times by changing several layer compositions until a model with optimal accuracy was obtained, however, the size of the resulting model was quite large so the researchers carried out SSDMobileNetV2 transfer learning to obtain the optimal model. The optimal model was obtained with a model accuracy of 92% and a model size of 18 MB. Object detection testing carried out in 3 test conditions resulted in an average object detection accuracy of 84.3%, and distance detection testing carried out in 10 conditions resulted in an average distance detection error of 2.1 cm. The Results show that the system was accurate and effective

Keywords: *model; machine learning; vision*

How to Cite: J. M. Parenreng, Andi Baso Kaswar, and Ibnu Fikrie Syahputra, “Visual Impaired Assistance for Object and Distance Detection Using Convolutional Neural Networks ”, J. RESTI (Rekayasa Sist. Teknol. Inf.), vol. 8, no. 1, pp. 26 - 32, Jan. 2024.
DOI: *https://doi.org/10.29207/resti.v8i1.5417*

1. Introduction

Vision is a very valuable gift from God, most aspects of human needs in the body are dominated by vision. Information on navigation, mobility, and object recognition are important aspects of vision because all this important information must reach the brain, to reach the brain, it must be through sight. According to[1], Vision is also the most important aspect of life, if you experience visual impairment it can interfere with activities, both in the teaching and learning process or in social interactions. However, vision does not escape interference, one of the most common visual impairments is the blind.

Blindness itself is often interpreted as not being able to see at all (total blindness), but blindness is damage to the eye, resulting in impaired vision. According to the Indonesian Blind Association (PERTUNI) in Rosenbloom[2], the blind are those who have no vision at all (totally blind) to those who still have residual vision but are unable to use their eyesight to read plain text measuring 12 points in normal light conditions. even though assisted with glasses.

The World Health Organization (WHO) estimates that around 180 million people worldwide have visual impairments, of which 40-45 million suffer from blindness. Meanwhile, the prevalence of blindness in Indonesia reaches 3 million people (1.5% of the population in Indonesia). Indonesia is the highest in Southeast Asia (Bangladesh 1%, India 0.7%, Thailand 0.6%).[3]

The working principle of this system is to apply the computer vision method combined with ultrasonic sensors as object detectors and distance detectors, this method will be rendered into audio form which will be information for the blind. So, this system acts as a third eye for the blind. The system we propose is expected to be able to provide information to the wearer in the form of object detection and distance, thus facilitating mobility and navigation for the blind.

In their research[4], they made smart glasses for the blind using the HC-SR04 ultrasonic sensor, buzzer, and Arduino as the microcontroller. The ultrasonic sensor in this study acts as a detector for what is in front of it using a distance limit. The buzzer acts as a source of information from the detection results of the ultrasonic sensor. The way this system works uses an ultrasonic sensor as a detector and a buzzer as an information provider which is attached to a pair of glasses.

In the research conducted[5], they made a wearable device in the form of glasses that will aim as a navigation system for the blind using ultrasonic sensors, wayfinding algorithms, Visual SLAM, POI-Graph, and route following. The way this system works is by making walking routes for the blind using a combination of way-finding algorithms, Visual SLAM, POI-Graph, and route following, then information will be given via sound, and ultrasonic sensors here play a role in detecting distance.

In the research conducted[6], they made smart glasses using stereo cameras, gyro sensors, vibration motors, and buzzers. All electronic elements are mounted on a pair of glasses, and vibration motors and buzzers are mounted on the left and right sides. The way it works is that the stereo camera is tasked with detecting objects, the gyro sensor aims to distinguish between the floor and objects, and the vibration and buzzer that are paired on both sides will sound if one side gets closer to the object.

In the research conducted[7], they made glasses connected to Google Vision to inform what was in front of blind people. This system consists of Raspberry Pi ZERO W, Raspberry Pi camera, and Google Vision. The way this system works is to use Google Vision as an object detector that is in front of the blind and then inform it via voice but must use the internet which will later be connected to the microcontroller so that Google Vision can be used.

Their research[8] created a distance detection system using vibrations applied to a stick. This system uses 2 Arduino and Raspberry Pi microcontrollers, as well as ultrasonic sensors and infrared sensors. 4 ultrasonic sensors are attached to the stick, and 1 infrared sensor. This system detects the distance using an ultrasonic sensor and then sends information in the form of vibrations.

2. Research Methods

Initially, the system was designed embedded to guarantee sensor and system compatibility. the biggest difficulty was finding embedded devices that could run tensor flow and were small enough to fit on glasses. after reviewing several components, the Raspberry Pi 4, Raspberry Pi v2 8MP camera, and sensor HC-SR04 were chosen. specifications of the components used are listed in Table 1.

Table 1. Software and Supporting Hardware

Components	Specification
Raspberry Pi 4	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz, 8GB LPDDR4 SDRAM, 2.4GHz and 5.0GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, 5.1V, 3A power via USB-C, USB, HDMI, GPIO.
Raspberry Pi Camera v2 8MP[9]	Sony IMX219 image sensor, fixed focus lens, 3280 x 2464-pixel static images, supports 1080p30, 720p60 and 640x480p60/90 video, 25mm x 23mm x 9mm dimensions.
HC-SR04[10]	Working voltage: 5V DC, working current: 15mA, Maximum range: 4 m, Minimum range: 2 cm, Measuring angle: 15 degrees, Trigger input signal: 10 us TTL pulse, Resolution: 1 cm, Ultrasonic Frequency: 40 kHz, Dimensions: 45*20*15mm

2.1 Convolutional Network

Convolutional Neural Network (CNN) is a part of deep learning that is used to classify images. CNN uses raw pixel data in the image as input and then learns and extracts these features, so it can conclude what object is being worked on[11].

Convolutional layers are layers that will extract raw data into several parts. Convolution extracts raw data from the input into several plots called feature maps, then performs calculations using filters from convolution to produce new output from the features map. Rectified Linear Unit (ReLU) is one part of the convolutions layer where this operation is applied every time a convolution operation is carried out which aims to be nonlinear in the model. ReLu uses “equation (1)” where all X values > 0, and produces 0 for all X values ≤ 0.

The subsampling layer is often also referred to as the pooling layer because in this layer there is only a pooling process. Pooling is done to save time and reduce the dimensions of the feature map while retaining important data from the +ReLU convolution. That process is called max pooling.

The fully connected layer is the last layer, where the output feature map that has gone through the convolutions layer and subsampling layer will be presented in this layer. This layer is tasked with classifying the output feature map results that have been presented from operations in the previous layer using the softmax activation function which produces probability values of 0 and 1 for each classification label that the model tries to predict[12], [13]

2.2 Architecture System

The design and how the system works is shown in Figure 1. Blind glasses are a tool to assist the mobility of blind people, in Figure 1 it is explained that these glasses consist of 3 parts, namely computer vision, ultrasonic, and output. In the computer vision section, there is a Convolutional Neural Network (CNN) process whose job is to recognize objects that have been previously trained, when the object is successfully detected, the system will send the results to the output section as information to the user. The ultrasonic section oversees detecting the left and right distances of the blind and sending the detection results to the output section. At the output, there is a voice assistant whose job is to provide information to the blind.

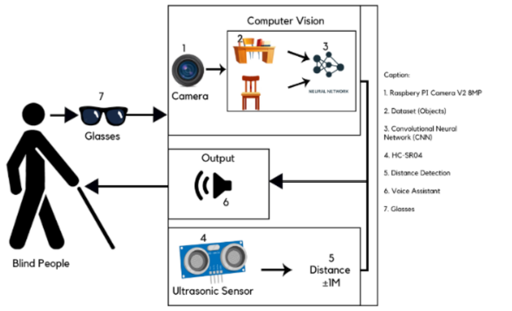

Figure 1. Architecture System

2.3 System Schema

Based on Figure 2 is a schematic of the entire system that will be made in this study, the explanation is as follows:

Part 1, The Raspberry PI Camera V2 8MP are used because they are mini but the resulting images are good.[9] It is useful for collecting images of objects known as datasets and detecting objects that are in front of the user.

Part 2, There are 3 types of objects we will use in this research: People, Tables, and Chairs. Objects are useful as datasets to be trained on machine learning s using the method of Convolutional Neural Network (CNN).

Part 3, Convolutional Neural Network (CNN) is a part of deep learning that is used to classify images. CNN uses raw pixel data in the image as input and then learns and extracts these features, so it can conclude what object is being worked on.[12], [13] CNN is a method that will be used when training datasets in machine learning[14]. This produces a model that will be embedded into the system so that it can detect objects.[11]

Part 4, HC-SR04 Ultrasonic Sensor as a distance detector in this system. This sensor will be paired on the left and right of the glasses. The maximum distance it can detect is 4 meters[10], [15]

Part 5, Earphones as media convey the main information to the users of this system. The earphones will receive information from the results of object detection and the distance.

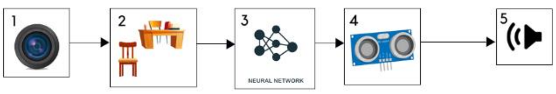

Figure 2. System Schema

3. Results and Discussions

Our test results are divided into 2, namely machine learning model training testing and testing scenarios. The goal is to observe the differences in each possibility being tested.

3.1 Machine Learning Model Training Test

We collected a total of 1000 datasets using Raspberry PI V2 Camera. This dataset will be divided into 3 object categories, namely people, chairs, and tables. We carried out this test twice to compare the results between testing without using the transfer learning method and testing using the transfer learning method.

Testing without using the Transfer Learning Method: In Table 2, these results were obtained after doing the 7 model training tests. The 6th scenario was the optimal model for this research, but the size of the model is very big.

Testing using the Transfer Learning Method: Look at Table 3, this test only requires 1 try to get an optimal model with a small size, and compatible to put in Raspberry 4.

Training without using the transfer learning method requires 7 tests to get optimal results while training using the transfer learning method requires 1 test to get optimal results.

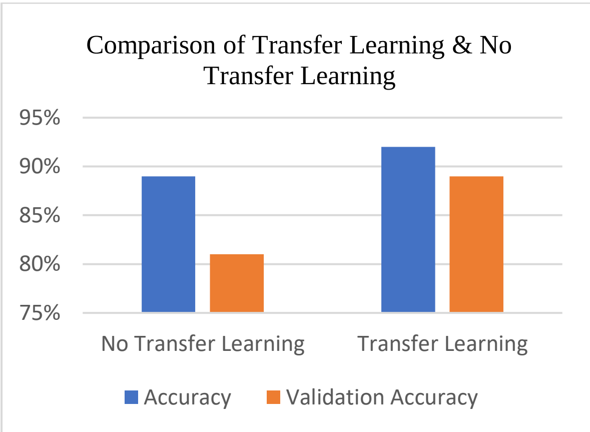

Figure 3. Comparison of Transfer Learning and no Transfer Learning

The comparison between the model without transfer learning and the SSDMobileNetv2 transfer learning model in Figure 3. looks very different in terms of accuracy and accuracy validation values. Based on the 7 test scenarios in Table 2, the optimal machine learning model was obtained in the 6th scenario, namely 89% with a validation accuracy of 81%, but the model size was quite large to deploy to the Raspberry Pi 4. This comparison was produced because of the transfer model training process[16][17]. The learning uses an existing CNN layer, namely SSDMobileNetV2, and the steps per epoch during training reach 40,000. Training the model with transfer learning produces a model with an accuracy value of 92% validation accuracy of 89% and a model size of 18mb. With these results, this model is used as the main model that will be deployed into the system by researchers.

Table 2. Machine learning model without transfer learning test data

Scenarios	*Convolutional Neural Network*				*Compile*		*Training*		*Result*		
	Operation	*Filters*	*Kernel*	*Activation*	*loss*	*Optimizer*	*Epoch*	*Steps per epoch*	*Acc*	*Val Acc*	*Size (MB)*
	Convolution	16	3x3	Relu							
	Pooling	-	2x2	-							
	Convolution	32	3x3	Relu							
1	Pooling	-	2x2	-	Categorical_Crossantropy	Adam	30	29	0.3772	0.3768	64
	Convolution	64	3x3	Relu							
	Pooling	-	2x2	-							
	Dense	16	-	Relu							
	Convolution	16	5x5	Relu							
	Pooling	-	2x2	-							
	Convolution	32	5x5	Relu							
2	Pooling	-	2x2	-	Categorical_crossantropy	Adam	30	29	0.8185	0.8188	114
	Convolution	64	5x5	Relu							
	Pooling	-	2x2	-							
	Dense	64	-	Relu							
	Convolution	32	5x5	Relu							
	Pooling		2x2	-							
	Convolution	64	5x5	Relu							
3	Pooling		2x2	-	Categorical_crossentropy	Adam	30	29	0.6619	0.8116	72
	Convolution	128	5x5	Relu							
	Pooling		2x2	-							
	Dense	64	-	Relu							
	Convolution	32	5x5	Relu							
	Pooling		2x2	-							
	Convolution	64	5x5	Relu							
4	Pooling		2x2	-	Categorical_crossentropy	Adam	30	29	0.8683	0.7971	219.8
	Convolution	128	5x5	Relu							
	Pooling		2x2	-							
	Dense	128	-	Relu							
	Convolution	32	3x3	Relu							
	Pooling		2x2	-							
	Convolution	64	3x3	Relu							
5	Pooling		2x2	-	Categorical_crossentropy	Adam	30	29	0.8523	0.8478	230
	Convolution	128	3x3	Relu							
	Pooling		2x2	-							
	Dense	128	-	Relu							
	Convolution	32	3x3	Relu							
	Pooling		2x2	-							
	Convolution	64	3x3	Relu							
6	Pooling		2x2	-	Categorical_crossentropy	Adam	30	29	0.8968	0.8188	116
	Convolution	128	3x3	Relu							
	Pooling		2x2	-							
	Dense	64	-	Relu							
	Convolution	16	3x3	Relu							
	Pooling		2x2	-							
	Convolution	32	3x3	Relu							
7	Pooling		2x2	-	Categorical_crossentropy	Adam	30	29	0.8698	0.7391	57.8
	Convolution	64	3x3	Relu							
	Pooling		2x2	-							
	Dense	64	-	Relu							

Table 3. Machine learning model without transfer learning test data

Scenarios	*Convolutional Neural Network*				*Compile*	*Training*		*Result*		
	Operation	*Filters*	*Kernel*	*Activation*	*Optimizer*	*Epoch*	*Steps per epoch*	*Acc*	*Val Acc*	*Size (MB)*
1	Convolution	32	3x3	Relu	Adam	1	40000	0.92	0.89	18
	Pooling	-	2x2	-						
	Convolution	64	3x3	Relu						
	Pooling	-	2x2	-						
	Dense	128	-	-						

3.2 Testing Scenarios

The testing design is illustrated in Figure 4. Testing with the glasses involved object detection and distance detection, while testing with the non-glasses just walking normally with eyes closed.

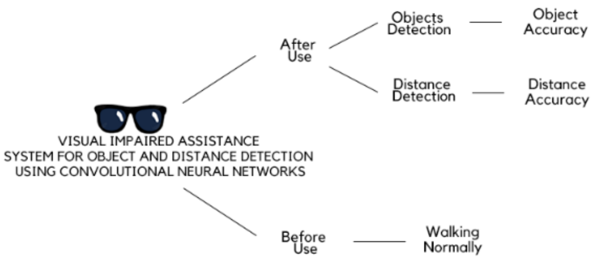

Figure 4. Testing Design

3.2.1. Object Detection Accuracy Testing

Testing was carried out with 3 conditions: the first was carried out with 1-2 each object, the second was carried out with 3-4 each object, the third was carried out with 5-6 each object

Test The Detection Accuracy of 1-2 each Object: The results of object detection accuracy testing with conditions 1-2 for each object can be seen in Figure 5. The actual total was 69 and the predicted total was 66.

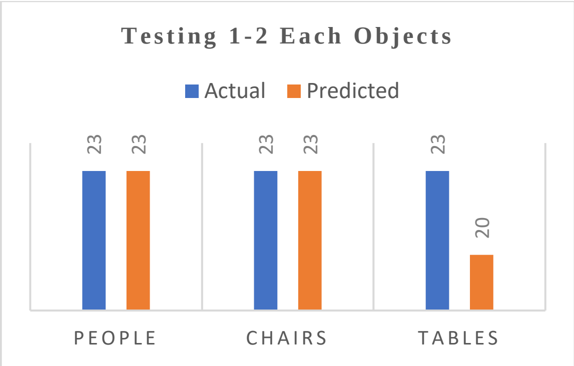

Figure 5. Testing 1-2 Each Objects

The average error in the test results is calculated using the Mean Absolute Deviation (MAD) formula. The calculation is shown in Equation 1

$$MAD = \sum_{t=1}^{23} \frac{69-66}{23} \quad (1)$$

MAD = 0.13
Percentage MAD = 0.13 * 100 = 13%

The calculation above shows an average percentage error of 13%. From the average percentage error, an accuracy of 87% is obtained, with the following calculation:

Accuracy = 100 – 13
Accuracy = 87%

Test The Detection Accuracy of 3-4 of each Object: The results of object detection accuracy testing with conditions 3-4 for each object can be seen in Figure 6. The actual total was 165 and the predicted total was 161.

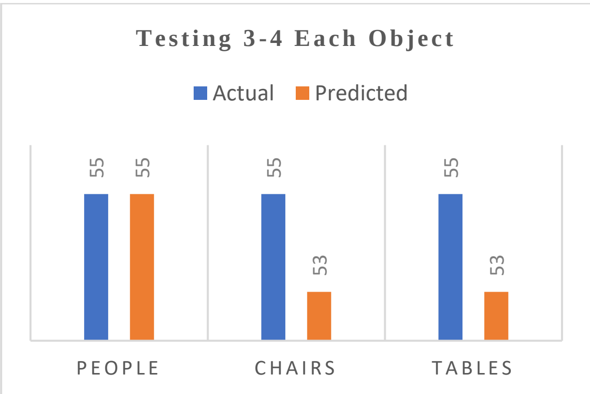

Figure 6. Testing 3-4 Each Objects

We calculated the average error in the test results using the Mean Absolute Deviation (MAD) formula. The calculation is shown in Equation 2.

$$MAD = \sum_{t=1}^{23} \frac{165-161}{23} \quad (2)$$

MAD = 0.17
Percentage MAD = 0.17 * 100 = 17%

The calculation shows an average percentage error of 17%. From the average percentage error, an accuracy of 83% is obtained, with the following calculation:

Accuracy = 100 – 17
Accuracy = 83%

Test The Detection Accuracy of 5-6 for each Object: The results of object detection accuracy testing with conditions 5-6 for each object can be seen in Figure 7. The actual total was 261 and the predicted total was 257.

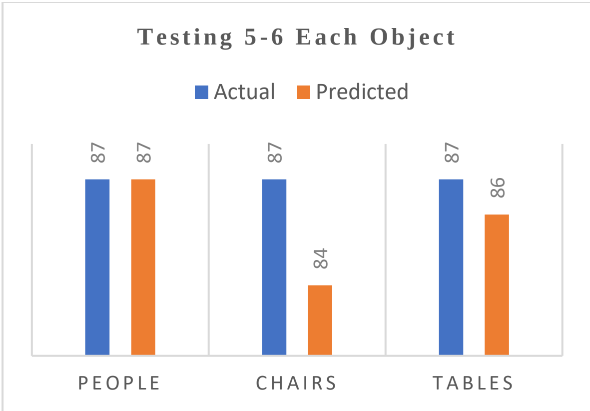

Figure 6. Testing 5-6 Each Objects

The average error in the test results was calculated using the Mean Absolute Deviation (MAD) formula. The calculation is shown in Equation 3.

$$MAD = \sum_{t=1}^{23} \frac{261-257}{23} \quad (3)$$

MAD = 0.17
Percentage MAD = 0.17 * 100 = 17%

The calculation shows an average percentage error of 17%. From the average percentage error, an accuracy of 83% is obtained, with this calculation:

Accuracy = 100 – 17
Accuracy = 83%

Table 4. Data on the average accuracy of object detection testing

Detection Testing	Detection Accuracy
Testing 1-2 each object	87%
Testing 3-4 each object	83%
Testing 5-6 each object	83%
Total Average	84.3%

After carrying out 3 accuracy testing conditions, accuracy values were obtained from these three conditions. To get the overall accuracy value from this test, we calculated the average of the three accuracy values as in Table 4.

3.2.2. Object Detection Accuracy Testing

Distance detection testing is carried out based on manual distance calculations and from ultrasonic sensors. Distance accuracy testing was carried out in up to 10 scenarios. The following are the results of accuracy distance testing in Table 5. The maximum difference between the actual distance and the predicted distance is 3 cm.

Table 5. Distance accuracy testing data

Scenario	Detection Accuracy		Difference
	Actual	Predicted	
1	10	9.6	1.6
2	20	18.5	1.5
3	30	29.4	0.6
4	40	38.3	1.7
5	50	47.5	2.5
6	60	57.0	3
7	70	68.4	1.6
8	80	77.3	2.7
9	90	87.5	2.5
10	100	97.1	2.9
Total	550	531	20.0

We calculate the average percentage error of the test results the calculation is shown in Equation 4

$$Average\ Error = \frac{\sum difference}{number\ of\ scenarios} \quad (4)$$

$$Average\ Error = \frac{20.6}{10}$$

$$Average\ Error = 2.1cm$$

From the calculation, the average error from tests carried out after 10 test scenarios is 2.1cm. The results of this calculation were obtained after testing with each distance, a value of 2.1cm is considered effective for distance detection, because the maximum detection error value in this test is 3cm

3.3 Discussions

The dataset in this study collected 700 data which was collected using an 8Mp Raspberry Pi camera. The collected dataset is cropped and duplicated in the form of a horizontal flip. The dataset is given a label to make it easier during the machine learning model training process. The dataset label is divided into 3 categories, namely: People, Chairs, and Tables. The labelled dataset is uploaded to Google Drive for training on Google Collaborator.

Machine learning model training requires several supporting libraries which will be imported into Google Collaborator. Load the dataset that has been uploaded to Google Drive, then train the machine learning model until you get a model with the best accuracy. Researchers carried out machine learning model training in up to 7 scenarios to obtain the best model. The 6th scenario produces an optimal model with an accuracy value of 89% and validation of 81%, but the model size is quite large. Therefore, researchers trained a machine learning model using the SSDMobileNetV2 transfer learning method, resulting in a more optimal model with 92% accuracy, and validation accuracy of 89%, and a model size of 18 Mb. This model will be used in this research and is saved in .tflite format [17], [18].

This research was designed in the form of a system prototype that was adapted to the user's needs in the form of glasses. The system prototype glasses are made from 3mm thick white acrylic material. The prototype glasses are designed to accommodate several supporting hardware inside. After the series of prototype glasses is complete, the training model will be deployed to the Raspberry Pi on the prototype glasses until the system is ready for testing.

System testing is carried out in 2 stages, namely test scenarios that will test object detection and distance detection, as well as system prototype testing. The results of object detection accuracy testing carried out with 3 test conditions, obtained an average detection accuracy of 84.3%, this accuracy is considered good for this prototype system. The results of the distance detection test carried out in 10 test scenarios produced an average error value of 2.1cm. This result is considered efficient because the maximum distance detection error in this test was 3cm. Meanwhile, the whole results for testing the prototype system are good, the sound feedback when an object is detected and the minimum distance measured are very good.

4. Conclusion

This research was designed with a prototype form of glasses aimed at making it easier for the visually impaired (users). The prototype glasses function well as intended by this research. The data received is in the form of responsive sound when objects are detected and distance. This research has carried out tests to measure accurate object detection, distance detection, and overall system prototype performance. This research produced an object detection accuracy value of 84.3%, based on 3 test conditions. The distance detection test results with an average distance detection error of 2.1, as well as the overall system prototype test results, are good and responsive. for future work you must add lots of objects as datasets so that this system can be more suitable for the visually impaired, apart from that, also use equipment that has very strong performance but is small so that it fits in this system.

Acknowledgement

This Research was completed because it was supported by the State University of Makassar, Faculty of Engineering, Informatic Engineering and Computer Education. Special thanks to the Network Security Laboratory and the Intelligence System Laboratory for their help and cooperation during this study

References

.[1] M. Z. Ritonga, "Level of Knowledge of Student in Sma Muhammadiyah 01 Medan To Decreased Visual Activity," *Jimki*, vol. 8, no. 3, pp. 35–40, 2021.

[2] A. A. Rosenbloom, "Low vision," *Optom. Vis. Sci.*, vol. 50, no. 3, p. 257, 2020, doi: 10.1097/00006324-197303000-00014.

[3] R. dr. S. Tirtonegoro, "Penyakit Mata Penyebab Utama Kebutaan Di Indonesia," *rsupsoeradji.id*, Nov. 13, 2019. https://rsupsoeradji.id/penyakit-mata-penyebab-utama-kebutaan-di-indonesia/ (accessed Apr. 19, 2022).

[4] R. Agarwal *et al.*, "Low cost ultrasonic smart glasses for blind," *2018 9th IEEE Annu. Inf. Technol. Electron. Mob. Commun. Conf. IEMCON 2018*, pp. 210–213, 2018, doi: 10.1109/IEMCON.2017.8117194.

[5] J. Bai, S. Lian, Z. Liu, K. Wang, and D. Liu, "Virtual-Blind-Road Following-Based Wearable Navigation Device for Blind People," *IEEE Trans. Consum. Electron.*, vol. 64, no. 1, pp. 136–143, 2018, doi: 10.1109/TCE.2018.2812498.

[6] J. H. Kim, S. K. Kim, T. M. Lee, Y. J. Lim, and J. Lim, "Smart glasses using deep learning and stereo camera," *2019 IEEE 8th Glob. Conf. Consum. Electron. GCCE 2019*, vol. 2, pp. 294–295, 2019, doi: 10.1109/GCCE46687.2019.9015357.

[7] M. Cabanillas-Carbonell, A. A. Chávez, and J. B. Barrientos, "Glasses connected to Google vision that inform blind people about what is in front of them," *2020 8th E-Health Bioeng. Conf. EHB 2020*, 2020, doi: 10.1109/EHB50910.2020.9280268.

[8] L. A. I. Cabrera-Sosa, L. E. Lopez-Garcia, A. Diaz-Sanchez, J. M. Rocha-Perez, V. H. Carbajal-Gomez, and G. Zamora-Mejia, "Design and Implementation of a White-Cane Based on the Vibration-Distance Sensing Technique for Blind People," *2020 IEEE Int. Conf. Eng. Veracruz, ICEV 2020*, 2020, doi: 10.1109/ICEV50249.2020.9289667.

[9] F. B. Setiawan, H. W. Kusuma, S. Riyadi, and leonardo H. Pratomo, "Penerapan PI Cam Menggunakan Program Berbasis Raspberry PI 4," *Cyclotr. J. Tek. Elektro*, vol. 5, no. 2, pp. 51–56, 2022.

[10] F.- Puspasari, I.- Fahrurrozi, T. P. Satya, G.- Setyawan, M. R. Al Fauzan, and E. M. D. Admoko, "Sensor Ultrasonik HCSR04 Berbasis Arduino Due Untuk Sistem Monitoring Ketinggian," *J. Fis. dan Apl.*, vol. 15, no. 2, p. 36, 2019, doi: 10.12962/j24604682.v15i2.4393.

[11] Yuhandri, Musli Yanto, and Eka Naufaldi Novri, "Application of Object Mask Detection Using the Convolution Neural Network (CNN)," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 4, pp. 922–929, 2023, doi: 10.29207/resti.v7i4.5059.

[12] M. M. Saufi, M. A. Zamanhuri, N. Mohammad, and Z. Ibrahim, "Deep learning for Roman handwritten character recognition," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 2, pp. 455–460, 2018, doi: 10.11591/ijeecs.v12.i2.pp455-460.

[13] K. Zakka, "CS231n Convolutional Neural Networks for Visual Recognition," *Stanford Univ.*, pp. 1–18, 2021, Accessed: Apr. 19, 2022. [Online]. Available: https://cs231n.github.io/convolutional-networks/.

[14] V. Tatan, "Understanding CNN (Convolutional Neural Network) | by Vincent Tatan | Towards Data Science," Dec. 23, 2019. https://towardsdatascience.com/understanding-cnn-convolutional-neural-network-69fd626ee7d4 (accessed Apr. 19, 2022).

[15] Irene S. Walingkas Meicsy E.I. Najoan Brave A. Sugiarso, "Irene S. Walingkas," *J. Tek. Elektro dan Komput.*, vol. Vol.8, no. Vol. 8 No. 3 (2019): Jurnal Teknik Elektro dan Komputer, pp. 121–131, 2019.

[16] Buyut Khoirul Umri and V. Delica, "Penerapan transfer learning pada convolutional neural networks dalam deteksi covid-19.," *Jnanaloka*, pp. 9–17, 2021, doi: 10.36802/jnanaloka.2021.v2-no2-9-17.

[17] Achmad Naila Muna Ramadhani, Galuh Wilujeng Saraswati, Rama Tri Agung, and Heru Agus Santoso, "Performance Comparison of Convolutional Neural Network and MobileNetV2 for Chili Diseases Classification," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 4, pp. 940–946, 2023, doi: 10.29207/resti.v7i4.5028.

[18] F. NURSULISTIO, "Deteksi Objek Masker Menggunakan Object Detection API dan TensorFlow Lite Model Maker," 2022, [Online]. Available: https://dspace.uii.ac.id/handle/123456789/41754%0Ahttps://dspace.uii.ac.id/bitstream/handle/123456789/41754/18523149.pdf?sequence=1&isAllowed=y.