Multi-Object Detection Using Single Shot Multibox Detector MobileNet for People with Visual Impairments

Indrabayu, Intan Sari Areni, Anugrayani Bustamin, Nur Latifa Jamaluddin, and Yuliani

***Abstract*—This research aims to assist people with visual impairments live their daily lives using intelligent technology based on computer vision. In the early stages, this research focused on detecting and estimating the barrier distance for the blind where the object of the barrier is poles and the motorcycles. The input data is obtained from a smartphone camera hung around the respondent's neck using POVIE while walking towards the object for various distances. Dynamic movement from users is the challenge in this research: processing data from a moving camera. The data is divided into training and testing data. The detection methods used are Single Shot Multibox Detector (SSD) and Mobilenet. Meanwhile, the Pinhole Model algorithm is used to estimate the distance between the obstacle object and the position of the blind person. The output of this application is sound using the text-to-speech library on Android. The best motorcycle detection system achieved accuracy of 100%, and for pole, detection obtained an accuracy of 98.66%.**

***Index Terms*— motorcycle detection, pole detection, pinhole model, SSD**

I. Introduction

In 2020, it was estimated that 160.7 million people of productive age have visual impairments [1]. It is estimated that this number will increase significantly with the growth of the community population, where it is predicted that there will be 703 million people with visual impairments in 2050 [2].

Blind people have limitations in their activities, especially outdoors. Most of them use assistive devices such as sticks to substitute for the sense of sight and hearing as a response to the stick. However, the use of canes does not fully assist them in walking. Sticks are inaccurate and are limited to providing more information about environmental situations, especially in detecting objects or people several meters ahead [3,4]. Smart sticks have been developed in several studies, as in [5-8], and have been sold commercially [9,10], but smart sticks tend to be expensive or require much hardware that makes it difficult for users to move around.

Researchers are currently developing more computer vision-based tools as an alternative to sticks and to overcome the shortcomings of using sticks. Rapid technological developments open opportunities for innovation in intelligent applications that are more robust and easier to use. Moreover, several previous studies have also proven that technology-assisted tools make it easier for people with multiple disabilities to do activities and learn [11-13].

The utilization of computer vision has been used to detect various objects in daily life. Rahman et al. presented a system for detecting obstacles in an indoor environment with a straightforward technique. The system saves each previous floor type as a reference image. Then, the obstacles are detected by segmenting the ROI value and comparing one frame with the next frame [14]. Kadafi and Utaminingrum also detected indoor obstacle objects using blob analysis. The method used is Connected Component Labeling to get blobs on the image with a system accuracy rate of 81.25% [15].

Previous research is still carried out in the indoor environment so that further research is needed to detect obstacle objects in the outdoor environment. More research is required for the outside environment due to the dangerous obstacles faced by the blind when walking. In addition, based on the results of a questionnaire conducted by several blind people in a previous study at a blind foundation, the objects that were most frequently hit were vehicles parked on the side of the road and electricity poles [16]. For the blind people at the foundation, an application for object detection of obstacles is very useful when walking outdoors. In addition to detecting the type of object, estimation of distance or relative position can be done using computer vision so that the use of sensors, especially on sticks, is no longer needed.

One widely used computer vision algorithm is the Single-Shot Multibox Detector (SSD). SSD is one of the deep learning Convolutional Neural Networks (CNN) architectures. This algorithm has been widely used to detect objects, such as cars [17], faces [18], facial occlusions [19], hand gestures [20], and even cow image segmentation [21]. The CNN methods provide the highest performance when compared with several algorithms, i.e., Support Vector Machine (SVM), Artificial Neural Network (ANN), Entropy Degradation Method (EDM) in lung cancer classification cases [22]. SSDs can be built using several CNN networks models, but [23-25] shows that MobileNet SSDs provide the best accuracy among other models with relatively fast

Manuscript received September 14, 2021; revised January 27, 2022. This work was supported by The Ministry of Research & Technology BRIN Indonesia (PDUPT Grant number: 752/UN.4.22/PT.02.00/2021) and Artificial Intelligence and Multimedia Processing (AIMP) research group, Universitas Hasanuddin, Makassar, Indonesia.

Indrabayu is Associate Professor at Informatics Department Universitas Hasanuddin, Makassar, Indonesia (corresponding author phone: +62811447755, e-mail: indrabayu@unhas.ac.id).

Intan Sari Areni is Associate Professor at Electrical Engineering Department, Universitas Hasanuddin, Makassar, Indonesia (e-mail: intan@unhas.ac.id).

Anugrayani Bustamin is Assistant Professor at Informatics Department, Universitas Hasanuddin, Makassar, Indonesia (e-mail: anugrayani@unhas.ac.id).

Nur Latifa Jamaluddin is Graduate Student of Electrical Engineering, Universitas Hasanuddin, Makassar, Indonesia (e-mail: nurlatifajamaluddin@gmail.com).

Yuliani is Undergraduate Student of Informatics, Universitas Hasanuddin, Makassar, Indonesia (e-mail: yuliani29997@gmail.com).

computation times. While the method used to estimate the distance of the obstacle object is the pinhole camera model. This method expresses the ratio of object size in pixels to the real-world equivalent to the ratio between the focal length of the camera lens and the distance of the object to the camera [26, 27].

This research develops an obstacle detection system to help blind people, poles, and motorcycles. This research develops an obstacle detection system to help blind people, poles, and motorcycles. This system uses the SSD MobileNet algorithm to detect obstructions and is applied to Android-based mobile devices. Furthermore, an estimation of the distance of the blind person to the obstacle object is also carried out. So that blind people are safer and more comfortable doing activities, especially in an outdoor environment.

II. Material and Methodology

The method used to detect is Single Shot Multibox Detector (SSD), while the pinhole camera model estimates distance. The steps taken in the SSD training process are shown in Fig. 1. The system created is in the form of an android application designed to be used with the help of a point-of-view camera holder (Povie) hung around the user's neck. The output of this application is in the form of sound using the text-to-speech library on Android.

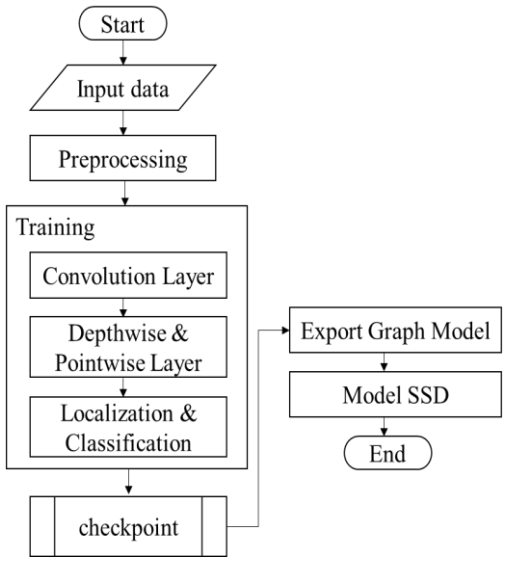


Fig. 1. SSD MobileNet Flowchart

A. Input Data

Data retrieval is done by placing the smartphone camera on the povie, which is hung on the user's neck. Pole data in videos taken by the user walking using a forward povie approaching the pole from 4 m to 2 m. The total image data to be trained is 45, and the five-video data to be tested.

Motorcycle data is taken from 4 meters to 2 meters. The data consists of images of parked motorcycles taken from six positions, namely front, back, right side, left side, right oblique, and left oblique. This data is divided into training and test data. As a representation of Indonesians' average height, this research involves four users, namely A, B, C, and D, who have different heights, respectively 148 cm, 154 cm, 167 cm, and 177 cm. The training data is 432 image data. The test data for determining the learning model of training results using the SSD MobileNet is 160 image data. The real-time system implementation test data is 448 image data. Fig. 2 shows the six-position scenario of motorcycles as an input system.

Fig. 2. The examples of data input

B. Preprocessing Data

The initial stage begins by inputting image training data that be preprocessed with several stages described as follows.

B.1. Resize Image

The pole data used has an initial size of 4000x3000 pixels, then resized manually to 900x675 pixels. Meanwhile, motor data with an initial size of 1920x1080 and 1920x1088 pixels were resized to 300x300 pixels. The purpose of resizing is to reduce the time-consuming training process.

B.2. Labelling Image

Image labeling is the initial stage where the input dataset is given a label or identifier (mark) to store image information. Labeling is done manually using the labeling application. Two classes created, namely the pole class and the non-pole class.

This labeling is saved in an XML file in PASCAL VOC format. Next, the XML file is converted to a CSV file which generates new information containing the width and height as the size of the image in pixels; *Xmin*, *Xmax*, *Ymin*, and *Ymax* as the location of the pixel bbox object points in the image; as well as to object classes that store object image data.

C. Training Process Using SSD

C.1. Convolution Layer

Convolution is a way to combine two series of numbers to produce a third series of numbers. There are two series of numbers in the input and the filter's kernel, while the third is the output. The input and the kernel both have a series of numbers in the form of a matrix. In the input, the number series is obtained based on the color level in each pixel. While in the kernel or filter, the number series is adjusted according to the researcher's needs. Several types of kernels are commonly used, including identity operations, edge detection, sharpen, box blur, and Gaussian blur.

The convolution layer is formed by running a filter over it. A filter is another block or cube with a smaller height and width, but the same depth swept over the base or original image. Filters are used to determine what pattern will be detected, which is then convoluted or multiplied by the value in the input matrix; the value in each column and row in the matrix depends on the type of pattern to be detected.

The convolutional process for pole detection is illustrated in Fig. 3 to understand the process well. Samples series are used as an input with a size of 416x416. The researchers use a sample series of the input size of 6x6 and use a kernel or filter for vertical edge detection operations with a size of 3x3. In contrast to the basic SSD architecture, this study uses MobileNet as the basic replacement architecture for VGG16.

It aims to reduce computation so that the object recognition process can be carried out on mobile devices. The convolution layer in the MobileNet architecture is divided into two processes, namely depthwise and pointwise convolution. The two processes are called depthwise separable convolution blocks.

- In the first layer, a 3-dimensional image (RGB) with a size of 300x300 goes through a 2-dimensional convolution process using 32 filters measuring 3x3x3. The convolution produces 32 feature maps with a size of 150x150. The convolution process can be seen in Fig. 3.

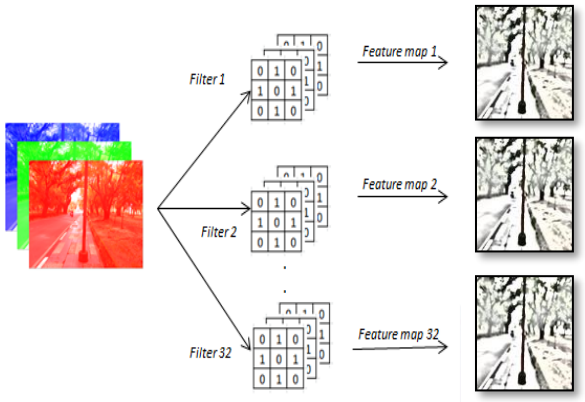


Fig. 3. Two-dimensional convolution process on a pole object

The convolution result in the previous layer, 150x150x32, is used as input data in the second convolution process. In this layer, the depthwise process is carried out, where each filter channel will be multiplied by each input data on each channel. This layer uses a 3x3x32 filter, resulting in a 150x150x32 feature map. This process is shown in Fig.4.

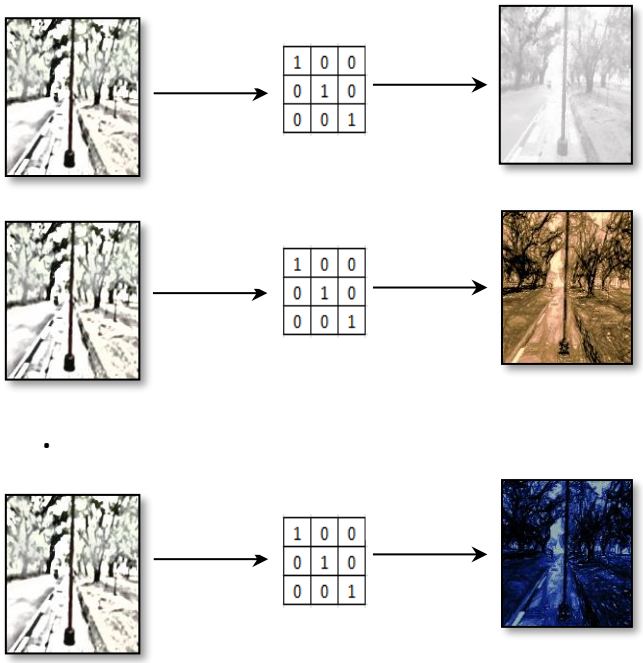


Fig. 4. Convolution process depthwise

In the next layer, a pointwise process is carried out where the results of the depthwise process measuring 150x150x32 as input data are multiplied by a 1x1x32 filter 64 times to obtain a 150x150x64 feature map, as shown in Fig. 5.

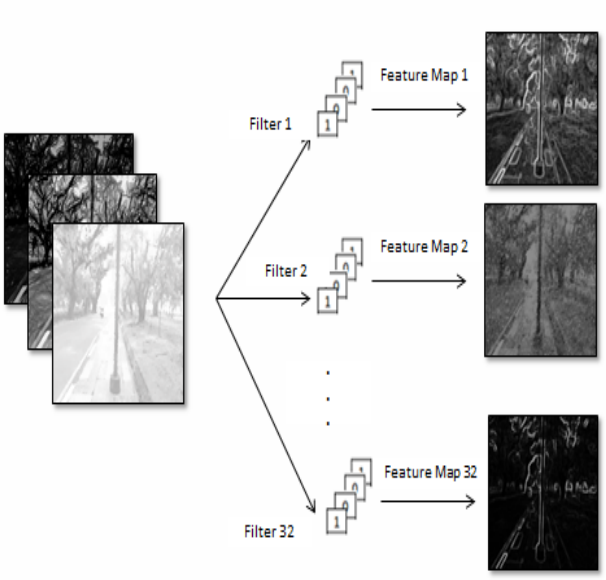


Fig. 5. Pointwise convolution process

C.2. Export Graph Model

During the training process, it will generate a checkpoint that is created automatically in the form of a graph tensor which aims to store information on the training process carried out; if the training process is complete, the next step is to export the graph tensor and make it a ready-to-use model.

C.3. Model SSD

The neural network training process produces a ready-to-use model for the testing process. The model SSD is a checkpoint file from training results and a tensor graph using the protobuf extension ".pb".

D. Estimated Distance and Relative Position of Objects

At this stage, testing will be carried out on the built system on the android-based. The flowchart of the system test can be seen in Fig. 6.

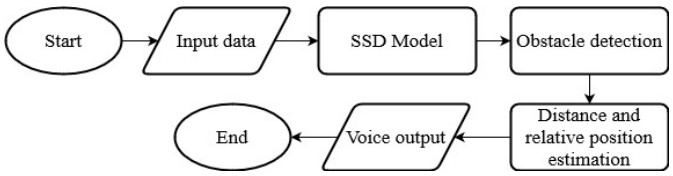


Fig. 6. Testing process flowchart

The calculation of the estimated distance is carried out using the pinhole camera model method. The pinhole camera is modeled as a closed box with a small hole punched on one side. The light rays entering the box through this pinhole form an inverted image on the opposite side in the image plane. The scenario to produce the inverted image is illustrated in Fig. 7. Objects A, B, and C at a distance (d) through the pinhole O are projected as A', B,' and C' with the same size and distance ratio as the actual object.

The stages of the pinhole camera model in this research are described as follows.

1. Measuring Actual Distance
   The first step is to determine the actual distance using a meter measuring tool and mark each distance.

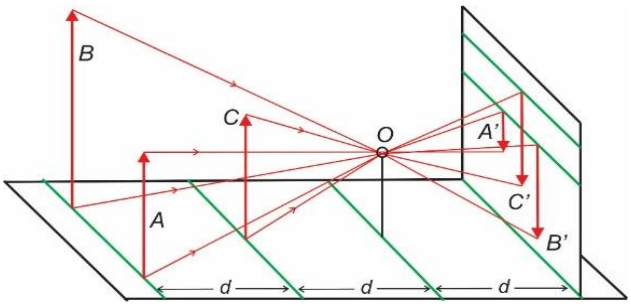


Fig. 7. Projection of objects on the pinhole camera model [28]

2. Determining thePixel Reference Coordinate Distance
This step measures the distance of the pixel reference coordinates horizontally, reflecting the object's actual distance in the image. The pixel reference point is determined by comparing the actual distance with the *y*-coordinate in the image, as illustrated in Fig. 8. Each image has coordinates (0,0) located at the top left, thus determining the pixel reference coordinates for each distance measured from the top of the image.

Fig. 8. Initialization of the pixel reference point coordinates in the image

3. Comparing the *ymax* with Each Pixel Reference Coordinate
The highest *y*-coordinate in the bbox (Y_{max}) resulting from the object detection stage is compared to each pixel reference point coordinate (K_{ac}). The Ymax coordinates of the bbox can be seen in Fig. 9.

Fig. 9. Example of Ymax bounding box

The smallest value difference is calculated by using equation (1).

$$S_j = \min\left(\sum_{i=1}^{4} | Y_{max} - K_{ac}(i)|\right) \quad (1)$$

4. Finding the Smallest Difference Value and Estimated Object Distance
Based on the results of the calculation of the difference in equation (1), to obtain the closest reference distance from the comparison of several distances, the estimated distance (J) can be calculated using equation (2). The estimated distance is obtained by multiplying the reference point (K_{ac}) and the actual distance (J_{ak}) then dividing by the highest coordinate of bbox (Y_{max}).

$$J = \frac{J_{ak} \times K_{ac}}{Y_{max}} \quad (2)$$

In addition to estimating the distance, the relative position estimation is also carried out on the motor detection system. Due to the pole position being the same from any point of view, the relative position is not observed in the pole detection system. Suppose the distance estimation is carried out by comparing the highest y-coordinate on the bbox with the y-coordinate of the reference distance, then, on the contrary. In that case, the relative position estimation compares the x-coordinate of the bbox with the x-coordinate of the blind in the image. Following are the steps for estimating the motorcycle relative position with visual impairments.

1. Determine the position of the blind people on the frame with respect to the *x*-axis.
2. Divide the bbox into three parts, namely bbox1, bbox2, and bbox3, where the first, second, and third bbox is the left, middle, and right sides of the motorcycle, respectively. Each bbox has *xmin* and *xmax* coordinates which are then compared with the *x* coordinates of the position of the blind people. In addition, each bbox is further divided into two parts, with the x-coordinate points being *a*, *b*, and *c*, respectively, as illustrated in Fig. 10.

Fig. 10. Bbox division into three parts

3. The *x*-coordinate of the blind people is compared with the *x*-coordinate point of the bbox to determine whether the blind people is aligned with the right, center, or left side of the motorcycle.

E. Performance Evaluation

Two conditions are used to determine the percentage of system accuracy in detecting objects, namely the condition of the pole being detected correctly in the system and the condition of the pole being undetected.

$$Percentage\ Accuracy = \frac{JTB}{JTK} \text{ x } 100\% \quad (3)$$

where:
JTB = Number of Correct Poles
JTK = Total Number of Poles

Next, calculate the error (*E*) of the distance estimation results using the following equation:

$$E = (Jac - Jes) \quad (4)$$

where:
E = Error
Jac = Actual Distance
Jes = Estimated Distance

III. Results and Discussion

A. Pole Detection

In the process of testing the pole detection system, the data used consist of 5 videos. The results of system testing can be seen in Table 1.

TABLE I
POLE DETECTION WITH SSD BY NUMBER OF OBJECTS PER FRAME

Video	Number of Frame	Number of Detected Frame			Number of Undetected Frame		
		2m	3m	4m	2m	3m	4m
1	226	74	73	74	0	2	3
2	229	74	74	74	1	2	4
3	172	57	55	54	0	2	4
4	236	77	77	76	1	2	3
5	300	99	97	96	0	3	5

Table 1 shows the detection results from test data with four different distances: the number of detected and undetected frames and the number of frames per video. An example of a correctly detected frame as a pole is shown in Fig. 11.

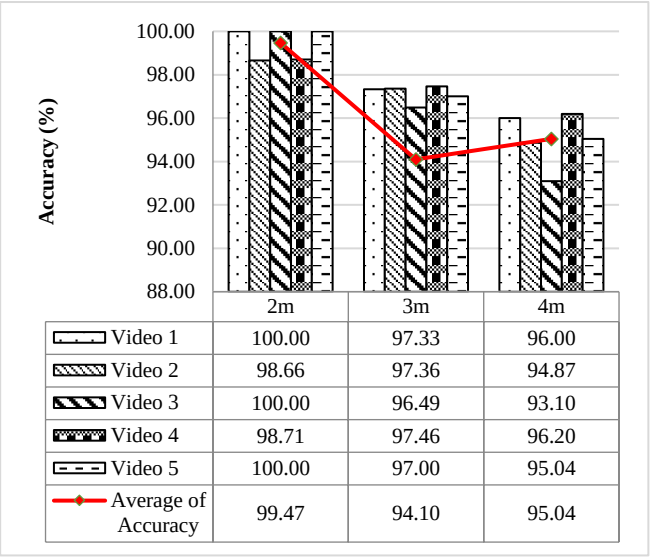
Fig. 11. Correctly detected frames

Poles correctly detected as poles will display the bounding box with Indonesian language "tiang" and confidence score. The confidence score is the value of the system's confidence in the object detected as a pole. Fig. 11 displays a confidence score of 96%, which means the system's confidence level is 96% that the object in the bounding box is a pole.

In each video, several frames are not detected, which is at 4 meters. This is because the distance of 4 meters is the furthest distance from the pole and the size of the pole is not far, so the bounding box that appears is still missing, where the bounding box sometimes appears and sometimes disappears. While at other distances, there are frames that are also not detected. For example, at 3 meters in video 5. This is because the camera moves while taking test data, so there are blurry frames as in Fig.12.

Fig. 12. Frames not detected properly

The accuracy of the pole detection results for each video with different distances is shown in Fig. 13.



	2m	3m	4m
Video 1	100.00	97.33	96.00
Video 2	98.66	97.36	94.87
Video 3	100.00	96.49	93.10
Video 4	98.71	97.46	96.20
Video 5	100.00	97.00	95.04
Average of Accuracy	99.47	94.10	95.04

Fig. 13. Pole Object Detection Results Accuracy

Fig. 13 shows the results of the accuracy of each different distance for pole detection based on equation (3). It shows that the highest average accuracy is 99.47 % in 2 meters. Moreover, the distance estimation between the pole object and the user is carried out, the results of which can be seen in Fig. 14.

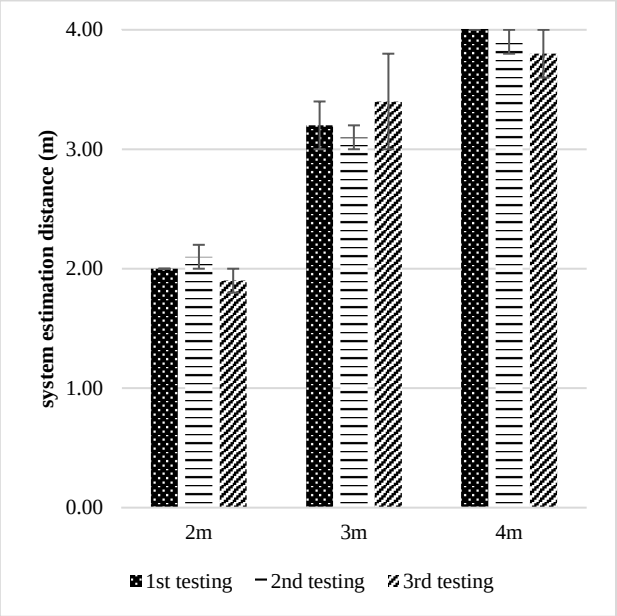


Fig. 14. Pole object distance estimation error

The average error at 2 meters, 3 meters, and 4 meters is 0.1m, 0.25m, and 0.175m, respectively. Some estimation errors are caused by the position of the povie hanging on the user's neck, which often shifts or moves the first time it is used so that a stable povie is needed when used. In addition, the video capture results become unstable because the smartphone camera used moves along while the user is walking.

B. Motor detection

In the initial process, the test image data is used as a reference in determining the model with the optimal learning rate for the system implementation in real-time. Five different learning rates are used to find the most optimal one in the training process. The results of the learning rate test can be seen in Table 2.

TABLE II
MOTOR DETECTION SYSTEM ACCURACY BASED ON LEARNING RATE

Distance	*Learning rate* (LR)					Mean Accuracy (%)
	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	
2m	82.76	94.83	98.28	99.14	94.83	93.97
3m	58.62	92.24	99.14	96.55	89.66	87.24
4m	41.38	89.66	91.38	88.79	82.76	78.79

Based on the test results, the highest detection was obtained using a learning rate of 10^{-3}. Optimal detection results do not depend on an increased number of learning rates. However, the precision and optimal learning rate is a learning rate that is neither too low nor too high. If the learning rate is too low, the change in the weight value becomes smaller so that the learning process becomes very slow; the computational process also becomes longer so that the results are not optimal if the target steps used have the same number. Meanwhile, if the learning rate is too high, the weight change becomes too large, making it difficult to reach the optimal point in the training process. In addition, the learning process also becomes longer.

Based on the learning rate testing process, the training result model implemented on Android smartphones has a learning rate of 10^{-3}. The model is converted into Android, which is then used to detect motorcycles in real-time. In this test, two different motorcycles were used: firstly, motorcycle as a sport motorcycle and secondly as a street motorcycle. The system accuracy based on distance and position can be seen in Table 3.

TABLE III
SYSTEM ACCURACY IN ESTIMATING DISTANCE

User	Type of Motorcycle	Distance		
		4m	3m	2m
A	1	92.90	100	100
	2	92.90	100	100
B	1	100	100	100
	2	100	100	100
C	1	85.70	92.90	100
	2	92.90	100	100
D	1	92.90	100	100
	2	100	100	100

Motorcycle detection results show the highest accuracy at 2 meters and the lowest at 4 meters. Decreasing accuracy is caused by the decreasing pixel density of the motorcycle object in the image at a distance.

After estimating the distance, the next step is to estimate the object's position and the user's relative position to the object. At this stage, the detected bounding box is divided into three parts to determine whether the blind person is aligned with the motorcycle's right, middle, or left side. In addition, the position of the motorcycle is indicated using numbers 1 to 6, respectively, indicating the position of the left side, right side, left oblique, right oblique, front, and back position of parked motorcycles. The testing results of the motorcycle's position and its relative position to the user can be seen in Tabel 4.

The detection results based on the motorcycle's position did not significantly differ. Still, low accuracy was obtained at positions 5 and 6, namely the front and rear positions of the motorcycle. The lack of training data causes this because the front and the back of motorcycle positions are not divided into three relative positions as in other motorcycle positions. In testing the relative position, the farther the position of the motorcycle is from the user, the position of the motorcycle on the frame tends to be in the middle, whether the blind person is parallel to the right, center, or left side of the motorcycle so that the lowest accuracy is at 4 meters. Notification of the user's relative position to the motorcycle is carried out at the optimal distance for the visually impaired to estimate the steps to avoid a parked motorbike, which is 3 meters.

TABLE IV
THE PERCENTAGE OF SYSTEM ACCURACY IN ESTIMATING MOTORCYCLE POSITION AND ITS RELATIVE POSITION TO USERS

User	Type of Motorcycle	Relative position by distance			Position					
		4m	3m	2m	1	2	3	4	5	6
A	1	64.29	100	92.86	88.89	83.33	88.89	72.22	33.33	83.33
	2	71.43	92.86	92.86	83.33	72.22	61.11	100	66.67	33.33
B	1	85.71	92.86	92.86	77.78	83.33	72.22	100	50	66.67
	2	85.71	85.71	92.86	100	100	83.33	100	50	66.67
C	1	78.57	92.86	100	83.33	83.33	77.78	94.44	16.67	33.33
	2	78.57	92.86	92.86	100	100	100	100	33.33	100
D	1	64.29	92.86	92.86	88.89	94.44	77.78	100	33.33	83.33
	2	85.71	100	92.86	100	100	100	100	66.67	66.67

IV. Conclusion

In this research, we have detected poles and motorcycles using the Single Shot Detector (SSD) MobileNet method and estimated the distance of these objects to the blind people using the pinhole camera model method. The pole detection system produces the highest average detection accuracy at 2 m with a distance estimation error of 0.1 m. Meanwhile, motorcycle detection obtained the best results at a learning rate of 10^{-3} with a percentage of 93.75%. The highest average detection accuracy based on distance is at 2 meters with 100%. Overall, the system works optimally at 2 meters to 4 meters. The motorcycle detection system works optimally to detect the type of sports motorcycle on the user with a height of 154 cm, where the system obtains 100% accuracy.

Meanwhile, pole detection obtained an accuracy of 98.66%, 97.36%, and 94.87%, respectively, at distances of 2m, 3m, and 4m. In the future, another type of obstacle can use to make the varying of data. Moreover, it is also necessary to test with other methods to compare each algorithm's accuracy and computation time to improve performance.

References

[1] A. P. Marques et al., "Global economic productivity losses from vision impairment and blindness," EClinicalMedicine Journal, vol. 35, no. 2021, pp. 1–9, 2021.

[2] P. Ackland, S. Resnikoff, and R. Bourne, "World blindness and visual impairment: despite many successes, the problem is growing," Community Eye Health Journal, vol. 30, no. 100, pp. 71–73, 2017.

[3] W. C. S. S. Simões and V. F. de Lucena, "Blind user wearable audio assistance for indoor navigation based on visual markers and ultrasonic obstacle detection," in *IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2016, pp. 60–63.

[4] H. M. U. Munir, F. Mahmood, A. Zeb, F. Mehmood, U. S. Khan, and J. Iqbal, "The voice enabled stick," in *20th International Conference of Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, 2017, pp. 1–5.

[5] M. A. Alshajajeer M. T. Almousa, and Q. A. Al-Haija, "Enhanced white cane for blind people people," Journal of Applied Computer Science and Mathematics, vol. 12, no. 2, pp. 9–13, 2017.

[6] M. H. Husin and Y. K. Lim, "InWalker: smart white cane for the blind," Disability and Rehabilitation: Assistive Technology, vol. 15, no. 6, pp. 701–707, 2020.

[7] Z. M. Yusof et al., "Design and analysis of a smart blind stick for visual impairment," Indonesian Journal of Electrical Engineering and Computer Science, vol. 11, no. 3, pp. 848–856, 2018.

[8] H. Sharma, M. Tripathi, A. Kumar and M. S. Gaur, "Embedded Assistive Stick for Blind people Persons," in *9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Bengaluru, India, 2018, pp. 1–6.

[9] WeWALK. (2020, September). WeWALK smart cane (Online). Available: https://wewalk.io/en

[10] The Bawa Cane Team. (2017, Februari). How to Guide a Visually Impaired Person (Online). Available: https://www.bawa.tech/

[11] G. E. Lancioni et al., "Technology-aided programs to enable persons with multiple disabilities to move through sequences of occupational activities independently," Journal of Developmental and Physical Disabilities, vol. 26, pp. 703–715, 2014.

[12] G. E. Lancioni, N. N. Singh, M. F. O'Reilly, J. Sigafoos, D. Oliva, and F. D'Amicof, "Technology-aided programs to enable persons with multiple disabilities to choose among environmental stimuli using a smile or a tongue response," Research in Developmental Disabilities, vol. 34, no. 11, pp. 4232–4238, 2013.

[13] F. Silman, H. Yaratan and T. Karanfiller, "Use of assistive technology for teaching-learning and administrative processes for the blind people people," Eurasia Journal of Mathematics, Science and Technology Education, vol. 13, no. 8, pp. 4805–4813, 2017.

[14] S. Rahman, Sana Ullah, and Sehat Ullah, "Obstacle detection in indoor environment for blind people using mobile camera," *J. Phys. Conf. Ser.* 960, 012046, 2018.

[15] A. J. A. Kadafi and F. Utaminingrum, "Mobile-based real-time detection of obstacle objects for visually impaired people using blob analysis," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 2, no. 1, pp. 423–432, 2018.

[16] Indrabayu, N. L. Jamaluddin, and I. S. Areni, "Detection and distance estimation against motorcycles as navigation aids for visually-impaired people," in *l2th International Conference on Information & Communication Technology and System (lCTS)*, Surabaya, Indonesia, 2020, pp. 224–228.

[17] X. Ziwei et al., "Face Occlusion Detection Based on SSD Algorithm," in *IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, Beijing, China, 2020, pp. 362–365.

[18] Gang Sha, Junsheng Wu, and Bin Yu, "Detection of spinal fracture lesions based on SSD," in *Proceedings of the 2020 International Conference on Aviation Safety and Information Technology (ICASIT 2020)*, Association for Computing Machinery, New York, NY, USA, 2020, pp. 539–542.

[19] Magalhães SA, Castro L, Moreira G, dos Santos FN, Cunha M, Dias J, Moreira AP, "Evaluating the single-shot multibox detector and YOLO deep learning models for the detection of tomatoes in a greenhouse," Sensors, vol. 21, no. 3569, pp.1–24, 2021.

[20] T. S. Yong, Kian Ming Lim, and Chin Poo Lee, "Wide Residual Network for Vision-based Static Hand Gesture Recognition," IAENG International Journal of Computer Science, vol. 48, no. 4, pp906–914, 2021.

[21] Rotimi-Williams Bello, Ahmad Sufril Azlan Mohamed, Abdullah Zawawi Talib, Daniel A. Olubummo, and O. Charles Enuma, "Enhanced Deep Learning Framework for Cow Image Segmentation," IAENG International Journal of Computer Science, vol. 48, no. 4, pp1182–1191, 2021.

[22] F. Taher, N. Prakash, A. Shaffie, A. Soliman, A. El-Baz., "An Overview of Lung Cancer Classification Algorithms and their Performance," IAENG International Journal of Computer Science, vol. 48, no. 4, pp1021–1027, 2021.

[23] Q. Chen, N. Huang, J. Zhou and Z. Tan, "An SSD Algorithm Based on Vehicle Counting Method," in *37th Chinese Control Conference (CCC)*, Guangzhou, China, 2018, pp. 7673–7677.

[24] X. Hu and B. Huang, "Face Detection based on SSD and CamShift," in *IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China, 2020, pp. 2324–2328.

[25] J. Huang et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, USA, 2017, pp. 3296–3297.

[26] R. K. Megalingam, V. Shriram, B. Likhith, G. Rajesh, and S. Ghanta, "Monocular distance estimation using pinhole camera approximation to avoid vehicle crash and back-over accidents," in *10th International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, India, 2016, pp. 1–5.

[27] Indrabayu, R. H. Putra, I. Nurtanio, I. S. Areni, and A. Bustamin, "Blob adaptation through frames analysis for dynamic fire detection," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 5, pp. 2189–2197, Oct. 2020.

[28] A. Majumder and M. Gopi, *Introduction to Visual Computing - Core Concepts in Computer vision, Graphics, and Image Processing*. London, UK: CRC Press, 2018, pp. 146-148.