International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I5P201

Volume 71 Issue 5, 1-8, May 2023

Original Article

Various Obstacles Detection Systems using Single Shot Multi-Box Detector (SSD) for Autonomous-Driving Vehicles

Indrabayu[1], Taslinda[2], Rizka Irianty[3], Sitti Wetenriajeng Sidehabi[4]

[1,2,3]Department of Informatics, Hasanuddin University, Indonesia
[4]Jurusan Otomasi Sistem Permesinan, Politeknik ATI Makassar, Indonesia

[1]Corresponding Author : indrabayu@unhas.ac.id

Received: 17 April 2023 Revised: 01 May 2023 Accepted: 06 May 2023 Published: 25 May 2023

Abstract *- One of the most important features of an autonomous vehicle is obstacle detection. The vehicle should be able to precisely and timely detect the presence of an obstacle to avoid a collision. This study aims to design and build an obstacle detection system to detect four types of obstacles (cars, motorcycles, people, and potholes) using the Single Shot Multi-box Detector (SSD) method and mobilenet v2 architecture. The input is video data extracted into frames and taken using a dash camera installed in the car. The dataset contains 720 images for each obstacle object. The training parameters are num_steps=20000 and batch_size=16. The result shows that the SSD method can be implemented properly for detecting and classifying obstacles in real-time. From the testing stage, the system obtains accuracy of 93.88%, 97.22%, 95.83%, and 94.44% at speeds of 10 km/h, 20 km/h hour, 30 km/hour, and 40 km/hour, respectively.*

Keywords *- Autonomous driving, Mobilenet v2, Obstacle detection, Real-time, SSD.*

1. Introduction

An autonomous driving vehicle is a car that can drive without the control of humans. Autonomous vehicles move with a series of Artificial Intelligence (AI) and consist of a collection of systems that work together to integrate environmental perception, path planning, and decision-making [1]. The principle purpose of autonomous vehicles rests in making sense of the complex and dynamic driving environment [2]. To achieve this goal, autonomous cars use a variety of sensors, cameras, radars, etc., to detect conditions in front of the car and make decisions like human drivers [3]. The main sensor is the GPS (Global Positioning System), which determines the car's exact position. Meanwhile, cameras and radars are installed around the car as the eye to monitor the surroundings. Infrared and ultrasonic waves are used to calculate the distance to other vehicles [4] accurately.

Autonomous driving is a technology that has the potential to change people's lifestyles and help people in their daily lives by providing reliable and safe transportation services [5]. Autonomous driving eases the driver's burden in navigating the car, especially for those with mobility impairments or elderly drivers [6],[7]. When driving a car, there are many obstacles that can hinder smooth driving. The majority of traffic accidents in Indonesia were caused by human factors (61%), followed by road condition factors (30%) and vehicle factors (9%) [8]. Autonomous cars are better at obtaining and maintaining situational awareness on the road [9]. They are expected to help eliminate the occurrence of traffic accidents caused by humans and reduce traffic congestion, economic loss caused by accidents, and pollution emissions [9], [10].

Victoria Transport Policy Institute suggests that it will be at least 2045 before half of the new vehicles are autonomous and 2060 before half of the vehicle fleet is autonomous [11]. Research on autonomous vehicles and related fields has been extensive these past years. The related fields include vehicle type classification, vehicle counting, obstacle detection, road signs detection, routing scheme, road safety and ethics, etc. For instance, Amiruddin et al. [12] designed a system to classify vehicles according to their categories, consisting of motorcycles, light vehicles, and heavy vehicles. Similarly, Sutrisno et al. [13] classified cars based on Car Make and Model Recognition (CMMR). Information on vehicle type and model can be utilized to analyze future transportation needs and improve transportation safety.

Meanwhile, work by Indrabayu et al. [14] detected and calculated the number of motor vehicles and modified motorcycles passed on a highway. Vehicle counting gives real-time traffic situation information to help autonomous cars plan the travel route. Karthikeyan and Sathiamoorthy [15] used a deep reinforced learning algorithm to control the steering angle of an autonomous vehicle. On the other hand, Kouonchie [16] focuses more on Vehicle-to-Infrastructure communication to maximize fuel-saving by determining their position and motion through traffic information in real-time.

Object detection is one of the vital elements in autonomous systems as it recognizes the existence and location of an object on the road. In the case of autonomous vehicles, object detection needs to be done in real-time to make quick decision-making and perform immediate action. The reaction time of an autonomous vehicle from obstacle recognition to applying brakes was found to be about 0.5 seconds [17]. Therefore, designing reliable object detection systems has gradually become the research focus in developing autonomous driving.

Currently, object recognition detection algorithms have developed a lot, but mainly focus on single object detection. Lim et al. proposed a real-time lightweight CNN for detecting road objects of various sizes [18], but the main focus was car objects. Sahal et al. used five classes, which are animal, person, tree, two-wheeled vehicle, and cars [26], while Yan et al. [20] used the KITTI dataset that includes 8 categories such as cars, pedestrians, and trucks. However, these works did not take into account the detection distance.

This study combines multiple object detection and compares system performance from multiple distances. The system can detect four types of obstacles (cars, motorcycles, people, and potholes) using the Single Shot Multi-box Detector (SSD) method and mobilenet v2 architecture. The SSD runs a deep Convolutional Neural Network (CNN), widely used for object detection and segmentation algorithms on an input image to produce network predictions from multiple feature maps.

2. Literature Review

2.1. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a deep learning algorithm that is enhanced from the Multi-Layer Perceptron (MLP) and designed to process data into two-dimensional shapes, such as images or sounds [21],[22]. CNN is used to classify labelled data using the supervised learning method. It imitates how human brains work that use neurons to transfer knowledge: biological neuron corresponds to an artificial neuron; kernels represent different receptors that respond to various features; activation functions mimic the function that only neural electric signals exceeding a certain threshold can be transmitted to the next neuron [22]. The CNN method has two types of layers: feature extraction and classification.

2.1.1. Feature Extraction

The feature extraction layer aims to extract features from the image so that they can be processed in subsequent layers. The feature extraction layer consists of the convolution layer, the Rectified Linear Unit (ReLU) activation function, and the pooling layer. The convolution layer consists of neurons that form a filter with pixel length and height. A filter extracts specific features from input data, while stride is the number of rows and columns traversed when sliding the convolutional filter over the input image. This process is a way to combine two series of numbers to produce a third series of numbers. In this study, the two series of numbers are contained in the input and kernel, while the third series of numbers is the output (feature map). Both input and kernel have a series of numbers in the form of a matrix.

In the convolution process, there is a parameter called padding. Padding extends the area of an image in a convolutional process by adding zeros around the edges of an image. Padding is a means of manipulating the output dimensions of the convolution layer (feature map). The purpose of padding is to preserve the original size of an image when applying a convolutional filter and enable the filter to perform full convolutions on the edge pixels [23].

The second layer in feature extraction is the pooling layer. The pooling layer combines each group of the previous layer's outputs into a single neuron. The pooling layer typically calculates either the maximum or the average value of the elements in the pooling window. This study uses average pooling, where this pooling combines the information from multiple adjacent pixels by averaging those adjacent pixels. Like convolutional layers, the pooling layer consists of a filter slid over all regions in the input according to its stride. Each layer in feature extraction has a ReLU activation function located between the convolution layer and the pooling layer. The ReLU activation function is an activation function that is zero if $x<0$ and is linear with slope 1 when $x>0$.

2.1.2. Classification

This layer consists of a fully-connected layer, which is a layer that functions in the application of multi-perceptron to perform transformations on data dimensions so that data can be classified linearly. This layer operates on a layer called the hidden state or hidden layer. This is the layer that produces the accuracy of the system.

2.2. Single Shot Multi-box Detector (SSD)

Single shot multi-box detector (SSD) is a development of the Deep Neural Network (DNN) algorithm for object detection. Based on its name, it is called ‘single shot’ because the tasks of object localization and classification are done in a single forward pass of the network; ‘multibox’ is the technique for bounding box regression; and ‘detector’ means the network is an object detector that also classifies the detected objects [24]. SSD produces a collection of bounding boxes (*bboxes*) and scores to determine the object' s class in each of these bounding boxes. SSD generates scores for the presence of each object category in each default box and produces adjustments to the box to match the object shape better.

Based on a study by Liu et al. [25], three deep learning methods have the highest performance: Faster RCNN, You Only Look Once (YOLO), and Single Shot Multibox Detector (SSD). The authors tested the detection time and accuracy using the PASCAL VOC, COCO, and ILSVRC datasets with varying input sizes. These three methods can detect objects in real-time but have different Mean Average Precision (MAP). Based on their MAPs, the SSD method has the highest accuracy compared to the Yolo and Faster RCNN methods.

3. Materials and Methods

3.1. Dataset

This study uses primary data obtained from direct data collection on the road using the MOLA N3 DDPAI Dash Camera. The camera is mounted on the windshield and positioned at a 90-tilt angle to get the best field of view. The data collection is carried out with different car speeds of 10 km/hour, 20 km/hour, 30 km/hour, and 40 km/hour in clear weather conditions. The camera resolution is 2560 x 1600 pixels. The illustration of the data collection process is shown in Figure 1.

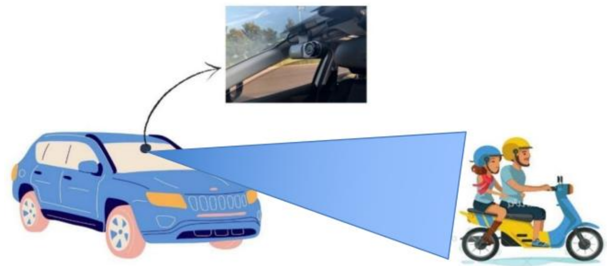

Fig. 1 Illustration of the data acquisition process

3.2. System Design

The system design consists of two main stages: training and testing. The system flowchart can be seen in Figure 2.

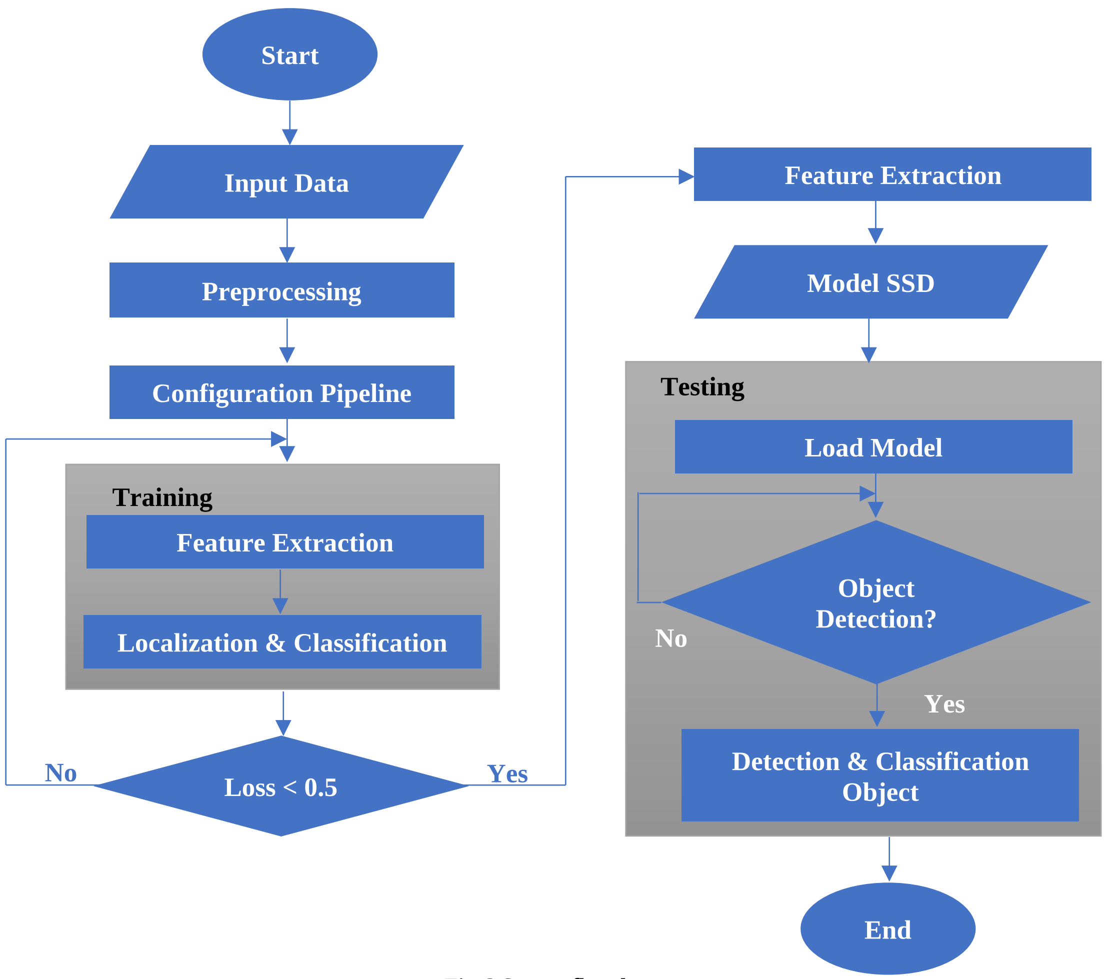

Fig. 2 System flowchart

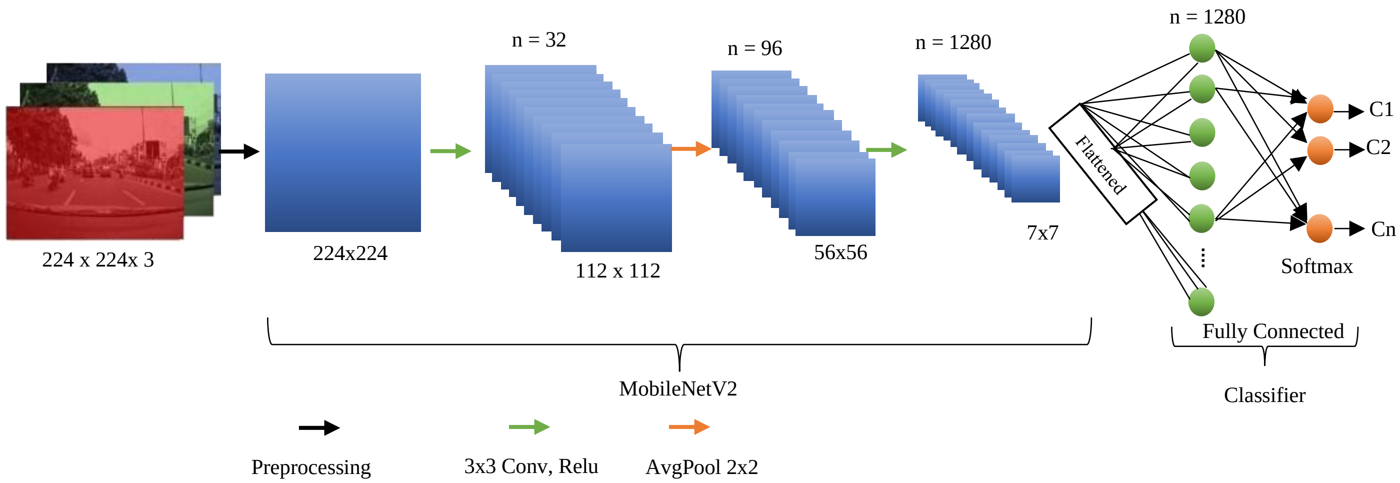

Fig. 3 Training process using MobileNetV2

3.3. Pre-processing

3.3.1. Frames Extraction

The video data has a resolution of 2560 x 1600 pixels, a frame rate of 30 fps and a duration of 60 seconds. The video is extracted into frames to be used in the next process. The data used for each object in this study is 900 frames, where there are 4 classes of obstacle objects, including car, motorcycle, person, and pothole.

3.3.2. Image Labelling

The video data has a resolution of 2560 x 1600 pixels, a frame rate of 30 fps and a duration of 60 seconds. The video is extracted into frames to be used in the next process. The data used for each object in this study is 900 frames, where there are 4 classes of obstacle objects, including car, motorcycle, person, and pothole. The image labeling process is done by providing a bounding box for each obstacle object in the frame.

3.3.3. Image Resizing with Bounding Box

Data with an initial size of 2560x1600 pixels is resized to 1280x800 pixels or half of the image's initial resolution. This is done to optimize the computing process without losing system accuracy.

3.3.4. Data Augmentation

Data augmentation creates artificial data to produce more variations of the train data. This study uses horizontal flip, scale, brightness, and contrast as data augmentation techniques.

Table 1. Training parameters

Parameter	Value
Num_class	4
Type	ssd_mobilenet_v2
Batch size	16
Num steps	20000

3.4. Pipeline Configuration

Following the pre-processing stage, the next step is configuring the pipeline to build the pre-trained model. The model is the mobilenetv2 SSD that is initialized using the training parameters shown in Table 1.

3.5. Model Training

This study uses the MobileNet V2 architecture to reduce the amount of computation for a more efficient object recognition process. SSD performs object detection by creating a bounding box. Meanwhile, MobileNet V2 works by extracting features from images. In object detection, SSD is used for image localization to determine object position, and MobileNet V2 is used to classify the objects.

This study uses a 2x2 filter applied with strides 1 and 2 in the pooling layer. After the pooling layer, the bottleneck feature is applied as the last activation maps before the fully-connected layers. Bottleneck is a block contained in the mobilenet v2 architecture. Bottleneck blocks are commonly known as inverted residual blocks (stride 1) and bottleneck residual blocks (stride 2). The MobileNet V2 model has a total of 7 bottleneck blocks. In the first layer, a 3-dimensional (RGB) image with a size of 224x224 goes through a 2-dimensional convolution process using a 3x3 filter. This convolution produces 32 feature maps with a dimension of 112x112. The convolution results from the previous layer will be used as input data in the following layers for each bottleneck block. Figure 3 illustrates the training process.

3.6. Loss

During the training process, it will produce a loss value that can be used as a metric to know whether the model works correctly. The lower the loss value, the better the system performance. In this study, the training process can be stopped if the loss value meets the specified requirements. But if the loss value is unsatisfactory, the training process will continue until it produces a low loss value.

3.7. Export Graph Model

The next process is the export graph model or export inference graph on the checkpoints generated from the training process. This process will generate a file with the extension protobuf (file. pb). This file will be used in the object detection process.

3.8. SSD Model

The end result of the neural network training produces a model ready for testing. This model is in the form of a checkpoint file, and tensor graph data is included in the protobuf file.

3.9. System Testing

The system testing stage aims to test how accurately a model performs from the training results. The testing process goes through the same stages as the training process but with different data. The data is classified using the model from the training results. The system testing process is also carried out in real-time.

System performance can be evaluated from the testing results. In the testing stage, system performance is analyzed using the multi-class confusion matrix method. The confusion matrix method is the base for calculating the precision, recall, F1-score, and accuracy values. There are four conditions in the confusion matrix:

- True Positive (TP) is a condition where the system successfully detects cars, motorcycles, people, and potholes
- True Negative (TN) is a condition where the system can ignore objects other than cars, motorcycles, people, and potholes
- False Positive (FP) is a condition where the system incorrectly detects negative objects such as cars, motorcycles, people, or potholes.
- False Negative (FN) is a condition where the system is unable to detect cars, motorcycles, people, and potholes that are present in the frame.

4. Results and Discussion

This chapter presents the results of the testing stage, where cars, motorcycles, people, and potholes obstacle detection is carried out using the SSD algorithm with the pre-trained model and mobilenetv2 as the architecture. The testing stage uses the model obtained from the training process to detect objects in real-time.

Finding the optimal parameters to generate the most suitable model for the system is necessary. In this case, experiments were carried out on several parameters to see which parameter values were most appropriate for the training process. The following are the parameters used in the training process:

- *Num_steps*. This parameter determines how many steps are in the training iteration of the algorithm. One 'step' corresponds to one batch of data processing. An experiment was conducted using *Num_steps* values of 20000 and 50000 to determine the best step number for the system. The result showed that *Num_steps* 20000 is the optimal value.
- *Batch_size*. Batch size is the number of data samples divided into each batch for use in the training process. An experiment has been carried out to determine the batch size parameter using values 8, 16, and 32. From the experiment, the value 16 is the optimal value.

Based on the experimental results, system testing was carried out using *num_steps* = 20000 and *batch_size* = 16. A dash camera is mounted on the inner part of the car windshield glass or behind the car's rear-view mirror. There are four scenarios with variations in car speed 10 km/h, 20 km/h, 30 km/h, and 40 km/h, each taken with a test distance of 20 meters. Show those parameters *num_steps* = 20000 and *batch_size* = 16 yielded the smallest loss value of 0.2. Examples of the classification results can be seen in Figure 4.

Fig. 4 Classification result

4.1. Scenario 1

The first scenario tests the system performance at a 10km/hour speed with a distance of 20 meters. Table 2 shows the confusion matrix and the results of testing scenario 1.

Table 2. Scenario 1 evaluation

	Predicted					TP	FP	FN	TN	Precision (%)	Recall (%)	F-Score (%)
	Class	Car	Motorcycle	Person	Pothole							
Actual	Car	8	0	0	0	8	0	0	16	100	100	100
	Motorcycle	0	3	2	0	3	1	2	19	75	60	66.67
	Person	0	0	5	0	5	2	1	17	71.43	83.33	76.92
	Pothole	0	0	0	6	6	0	0	18	100	100	100
Accuracy = 93.88%												

Table 3. Scenario 2 evaluation

	Predicted					TP	FP	FN	TN	Precision (%)	Recall (%)	F-Score (%)
	Class	Car	Motorcycle	Person	Pothole							
Actual	Car	7	0	0	0	7	0	0	11	100	100	100
	Motorcycle	0	3	1	0	3	0	1	14	100	75	85.71
	Person	0	0	3	0	3	1	0	14	75	100	85.71
	Pothole	0	0	0	4	4	0	0	14	100	100	100
Accuracy = 97.22%												

Table 4. Scenario 3 evaluation

	Predicted					TP	FP	FN	TN	Precision (%)	Recall (%)	F-Score (%)
	Class	Car	Motorcycle	Person	Pothole							
Actual	Car	6	0	0	0	6	0	0	6	100	100	100
	Motorcycle	0	2	1	0	2	0	1	9	100	66.67	80
	Person	0	0	1	0	1	1	0	10	50	100	66.67
	Pothole	0	0	0	2	2	0	0	10	100	100	100
Accuracy = 95.83%												

Table 5. Scenario 4 evaluation

	Predicted					TP	FP	FN	TN	Precision (%)	Recall (%)	F-Score (%)
	Class	Car	Motorcycle	Person	Pothole							
Actual	Car	5	0	0	0	5	0	0	4	100	100	100
	Motorcycle	0	1	1	0	1	0	1	7	100	50	66.67
	Person	0	0	1	0	1	1	0	7	50	100	66.67
	Pothole	0	0	0	1	1	0	0	8	100	100	100
Accuracy = 94.44%												

Scenario 1 shows that for car, motorcycle, person, and pothole, there are 8, 3, 5, and 6 correct predictions, respectively. There are 2 detection errors in the motorcycle class, where they are classified as a person. Motorcycles and persons obtain f1-score values of 66.67% and 76.92%, respectively, while car and pothole objects obtain 100.00% accuracy. The total accuracy is 93.88%.

4.2. Scenario 2

The second scenario tests the system performance at a 10km/hour speed with a distance of 20 meters. Table 3 shows the confusion matrix and the results of testing scenario 2. Scenario 2 shows 7, 3, 3, and 4 true positive predictions for car, motorcycle, person, and pothole class, respectively. In this scenario, there is one error detection where the motorcycle is classified as a person. The motorcycle and person classes have an f1-score of 85.71%, while the car and pothole classes have an f1-score of 100.00%. The total accuracy is 97.22%.

4.3. Scenario 3

The third scenario tests the system performance at a 10km/hour speed with a distance of 30 meters. Table 4 shows the confusion matrix and the results of testing scenario 3. Scenario 3 shows that for car, motorcycle, person, and pothole, there are 6, 2, 1, and 2 correct predictions, respectively. There is one detection error for the motorcycle object, where the motorcycle is classified as a person. The motorcycle and person objects have f1-score values of 80.00% and 66.67%, respectively, while the car and pothole objects have 100.00% f1-score values. The overall accuracy is 95.83%.

4.4. Scenario 4

The first scenario tests the system performance at a 10km/hour speed with a distance of 40 meters. Table 5 shows the confusion matrix and the results of testing scenario 4. Scenario 4 shows that the system made 5 correct predictions for cars and 1 for motorcycles, persons, and potholes. There is one detection error for the motorcycle object, where the motorcycle is classified as a person. The motorcycle and person objects have f1-score values of 66.67%, while the car and pothole objects have 100% f1-score values. The overall accuracy is 94.44%.

From the results, the highest performance was obtained when the object was 20 meters distance from the camera. This opens room to improve system accuracy related to object distance. Furthermore, the results show that cars and potholes were classified successfully from any distance.

However, motorcycles often got misclassified as people. This error occurred because the two objects have some similar features. The motorcycle dataset contained some images of people riding motorcycles; hence motorcycles are prone to misclassification. Figure 5 shows an example of incorrect classification.

Fig. 5 Example of system misclassification

5. Conclusion

This study aims to build an obstacle detection and classification system for autonomous driving technology development using the Single Shot Multibox Detector algorithm and mobilenet v2 architecture. The objects to detect and classify are cars, motorcycles, persons, and potholes. To find the best model for the dataset, it is necessary to specify the optimal parameters for model training. The experimental results showed that parameters *num_steps* = 20000 and *batch_size* = 16 yielded the smallest loss value of 0.2.

In the testing stage, there were four scenarios with variations in car speed, each taken at a distance of 20 meters. The system obtained an accuracy of 93.88%, 97.22%, 95.83%, and 94.44% at speeds of 10 km/h, 20 km/h hour, 30 km/hour, and 40 km/hour, respectively. From the results, the highest performance was obtained when the object was 20 meters distance from the camera. The results also show that cars and potholes were classified successfully from any distance, but motorcycles often got misclassified as people. This error occurred because the two objects have some similar features. The results show that the SSD algorithm can be appropriately implemented in detecting and classifying road obstacles in real time but still needs improvement related to detection distance.

References

[1] Gourav Bathla et al., “Autonomous Vehicles and Intelligent Automation: Applications, Challenges, and Opportunities,” *Mobile Information Systems*, vol. 2022, pp. 1-36, 2022. [Crossref] [Google Scholar] [Publisher Link]

[2] Scott Drew Pendleton et al., “Perception, Planning, Control, and Coordination for Autonomous Vehicles,” *Machines*, vol. 5, no. 1, 2017. [Crossref] [Google Scholar] [Publisher Link]

[3] Saeed Asadi Bagloee et al., “Autonomous Vehicles: Challenges, Opportunities, and Future Implications for Transportation Policies,” *Journal of Modern Transportation*, vol. 24, pp. 284-303, 2016. [Crossref] [Google Scholar] [Publisher Link]

[4] N. Deepika, and V. V. Sajith Variyar, “Obstacle Classification and Detection for Vision Based Navigation for Autonomous Driving,” *In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2092–2097, 2017. [Crossref] [Google Scholar] [Publisher Link]

[5] Darsh Parekh et al., “A Review on Autonomous Vehicles: Progress, Methods and Challenges,” *Electronics*, vol. 11, no. 14, p. 1-18, 2022. [Crossref] [Google Scholar] [Publisher Link]

[6] Ching-Yao Chan, “Advancements, Prospects, and Impacts of Automated Driving Systems,” *International Journal of Transportation Science and Technology*, vol. 6, no. 3, pp. 208-216, 2017. [Crossref] [Google Scholar] [Publisher Link]

[7] Corey D. Harper et al., “Estimating Potential Increases in Travel with Autonomous Vehicles for the Non-Driving, Elderly and People with Travel-Restrictive Medical Conditions,” *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 1-9, 2016. [Crossref] [Google Scholar] [Publisher Link]

[8] J. I. Nantika, Penyebab Kecelakaan 61% Karena Kecerobohan Pengendara, 2021. [Online]. Available: https://mediaindonesia.com/megapolitan/399413/penyebab-kecelakaan-61-karena-kecerobohan-pengendara

[9] Caner Filiz, “Can Autonomous Vehicles Prevent Traffic Accidents?,” *Accident Analysis and Prevention*, 2020. [Crossref] [Google Scholar] [Publisher Link]

[10] Sergiu Coicheci, and Ioan Filip, “Self-Driving Vehicles: Current Status of Development and Technical Challenges to Overcome,” *In 2020 IEEE 14th International Symposium on Applied Computational Intelligence and Informatics (SACI), IEEE*, pp. 000255–000260, 2020. [Crossref] [Google Scholar] [Publisher Link]

[11] Victoria Transport Policy Institute, Autonomous Vehicle Implementation Predictions Implications for Transport Planning, 2023. [Online]. Available: https://www.vtpi.org/avip.pdf

[12] M. Rudini Kurniawan Amiruddin et al., “An Approach for Vehicle’S Classification Using BRISK Feature Extraction,” *In 2021 3rd International Conference on Electronics Representation and Algorithm (ICERA)*, pp. 83–88, 2021. [Crossref] [Google Scholar] [Publisher Link]

[13] A. Sutrisno et al., “Reducing Area Recognition for Vehicle Model Classification Using Car’S Front Side,” *In 2021 4th International Conference on Information and Communications Technology (ICOIACT)*, pp. 290–294, 2021. [Crossref] [Google Scholar] [Publisher Link]

[14] Indrabayu et al., “A Solution for Automatic Counting and Differentiate Motorcycles and Modified Motorcycles in Remote Area,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 2, 2022. [Crossref] [Google Scholar] [Publisher Link]

[15] Paule Kevin Nembou Kouonchie, Vitalice Kalecha Oduol, and George Nyauma Nyakoe, “Roadside Unit Transmission Control for Energy Efficiency in Vehicle-to-Infrastructure Communication Network,” *International Journal of Engineering Trends and Technology*, vol. 70, no. 5, pp. 145-158, 2022. [Crossref] [Google Scholar] [Publisher Link]

[16] Chris Urmson, "Driving Beyond Stopping Distance Constraints," *In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1189-1194, 2006. [Crossref] [Google Scholar] [Publisher Link]

[17] M. Karthikeyan, and S. Sathiamoorth, “Deep Reinforcement Learning for Computerized Steering Angle Control of Pollution Free Autonomous Vehicle,” *International Journal of Engineering Trends and Technology*, vol. 69, no. 4, pp. 204-208, 2021. [Crossref] [Google Scholar] [Publisher Link]

[18] Byeonghak Lim, Bin Yang, and Hakil Kim, "Real-Time Lightweight CNN For Detecting Road Object of Various Size," *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 202-203, 2018. [Crossref] [Google Scholar] [Publisher Link]

[19] Dipali Bhabad et al., "Object Detection for Night Vision Using Deep Learning Algorithms," *International Journal of Computer Trends and Technology*, vol. 71, no. 2, pp. 87-92, 2023. [Crossref] [Publisher Link]

[20] Min Yan et al., "Improved Real-Time Joint Object Detection and Road Segmentation Multi-Task Network," *2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE)*, pp. 541-545, 2021. [Crossref] [Google Scholar] [Publisher Link]

[21] Arohan Ajit, Koustav Acharya, and Abhishek Samanta, "A Review of Convolutional Neural Networks," *In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pp. 1-5, 2020. [Crossref] [Google Scholar] [Publisher Link]

[22] Zewen Li et al., "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *In IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999-7019, 2022. [Crossref] [Google Scholar] [Publisher Link]

[23] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi, "Understanding of a Convolutional Neural Network," *In 2017 International Conference on Engineering and Technology*, pp. 1-6, 2017. [Crossref] [Google Scholar] [Publisher Link]

[24] Ashwani Kumar, Zuopeng Justin, and Hongbo Lyu, “Object Detection in Real Time Based on Improved Single Shot Multi-Box Detector Algorithm,” *EURASIP Journal on Wireless Communications and Networking*, no. 204, 2020. [Crossref] [Google Scholar] [Publisher Link]

[25] Wei Liu et al., “SSD: Single Shot Multibox Detector,” *In Computer Vision – ECCV 2016*, vol. 9905, pp. 21-37, 2016. [Crossref] [Google Scholar] [Publisher Link]

[26] Mochammad Sahal, Ade Oktavianus Kurniawan, and Rusdhianto Effendi Abdul Kadir, “Object Detection for Autonomous Vehicle Using Single Camera with Yolov4 and Mapping Algorithm," *2021 4th International Seminar on Research of Information Technology and Intelligent Systems*, pp. 144-149, 2021. [Crossref] [Google Scholar] [Publisher Link]