Indonesian Journal of Engineering and Technology (INAJET)
e-ISSN: 2623-2464
Vol.7 No.1 September 2024
https://journal.unesa.ac.id/index.php/inajet

Traffic Sign Recognition Using Detector-Based Deep Learning Method

Alfito Mulyono[1], Ervin Yohannes[2]
[1,2]Jurusan Teknik Informatika, Fakultas Teknin, Universitas Negeri Surabaya, Surabaya, 60231, Indonesia
[1]alfito.20038@mhs.unesa.ac.id
[2]ervinyohannes@unesa.ac.id

Abstract

Traffic is a key element in the transportation system. Traffic is an integral part of urban life and a key element in the transportation system. Traffic safety is a major concern to prevent accidents and ensure safe mobility. Traffic accidents are one of the most common occurrences. . But on the other hand, the increase in road accidents is increasing, which can be caused by people's lack of knowledge about traffic. The main solution to overcome this problem is to increase knowledge about traffic. The application of artificial intelligence, especially object detection methods with the use of detector-based deep learning methods, is one method that has proven efficient in detecting objects in real-time.

In this research, object recognition is performed using SSD (Single Shot MultiBox Detector) where the model is trained and tested for its performance in detecting traffic signs in Indonesia. From the research results, the mAP 50 and mAP 50-95 values are 89.66% and 65.49%, respectively.

Keyword: *Deep Learning, SSD, Traffic Signs.*

I. INTRODUCTION

Traffic is an integral part of urban life and a key element in the transportation system. Traffic safety is a major concern to prevent accidents and ensure safe mobility. Traffic accidents are one of the most common occurrences. Although many traffic signs have been arranged by the state to reduce the rate of accidents, accidents are still unavoidable. Many factors cause traffic accidents, among others: weather, vehicles, road conditions and driver habits. The main solution to overcome this problem is to increase knowledge about traffic by bringing the authorities in front of the community to explain some of the many traffic signs. One of the shortcomings of this method is that the public is less able to accept everything conveyed by the authorities. However, with the development of technology today can help some of the problems caused [1].

In recent years, image recognition and image processing technologies have developed rapidly. The application of artificial intelligence, particularly object detection methods, has become a major focus in the development of traffic sign recognition systems. One prominent approach is the use of detector-based deep learning methods, which have proven to be efficient in detecting objects in real-time. There are two types of architectures that are often used in object recognition using detector-based deep learning methods, namely SSD (Single Shot MultiBox Detector).

There are several previous studies that conducted traffic sign recognition using the SSD method to perform traffic sign recognition as conducted in a study entitled “Traffic Sign Board Detection Using Single Shot Detection (SSD)” [2] which resulted in a mAP of 78%.

In this research, the traffic signs used as datasets focus on traffic signs in Indonesia. Then the SSD model is tested for its performance in recognizing traffic signs in Indonesia. By conducting this research, it is expected to make a significant contribution in improving traffic safety through the implementation of object detection technology. The implementation of an efficient and accurate traffic sign recognition system is expected to help reduce the risk of traffic accidents and make a positive contribution to transportation management in urban areas.

II. LITERATURE OVERVIEW

A. SSD (Single Shot MultiBox Detector)

Liu W et al. proposed the SSD algorithm [3] in 2016, which was mainly based on the improvement of YOLO's location inaccuracy, insufficient accuracy, and low recall rate at that time. The improvement of the SSD algorithm mainly has the following points: Feature fusion is performed for feature extraction of different sizes, which improves the robustness of network training and enables deeper context learning; instead of using YOLO operation to predict objects after the full connection layer [4], [5], [6], CNN is added to the backbone network to predict directly.

Combined with the anchor mechanism in Faster R-CNN, candidate regions are obtained using different prior boxes, and the recall rate is improved. However, the disadvantage is that the accuracy of the model for small object detection is not high, and the positive and negative samples are very uneven. The schematic diagram of the SSD algorithm is shown in Fig. 1.

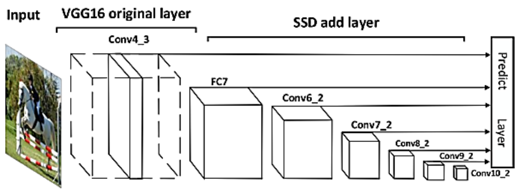


Fig. 1 *SSD Algorithm* [7]

B. Evaluation Metrics

Confusion matrix is used to evaluate the performance of a machine learning model. Confusion matrix is a matrix that displays the predictions of the actual classification and the predicted classification [8]. There are four classifications in the confusion matrix, namely True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP) derived from actual and predicted values. Where TP (True Positive) is the number of correctly classified positive samples; TN (True Negative) is the number of correctly classified negative samples; FP (False Positive) is the number of negative samples misclassified as positive; FN (False Negative) is the number of positive samples misclassified as negative [8]. An illustration of the confusion matrix can be seen in Fig. 2.

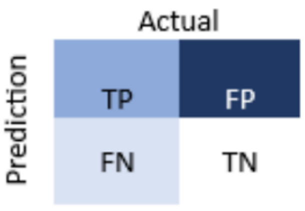


Fig. 2 *Confusion Matrix* [8]

Model performance can be calculated using precision, recall, and mAP derived from the confusion matrix. Precision is used to measure the accurately predicted positive observations out of the total predicted observations in the positive class, which is formulated as [9]:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Recall is used to measure the proportion of positive observations that are correctly classified. It is formulated as [9]:

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Average Precision is used to provide per-class average precision, and Average Recall is used to provide per-class average formulated as [9]:

$$Average\ Precision = \frac{\sum_{q=1}^{Q} \frac{TP_i}{TP_i + FP_i}}{Q} \tag{3}$$

$$Average\ Recall = \frac{\sum_{q=1}^{Q} \frac{TP_i}{TP_i + FN_i}}{Q} \tag{4}$$

Mean Average Precision (mAP) is a popular evaluation metric that gives the average AP value of all classes in a single number formulated as [9]:

$$(mAP) = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \tag{5}$$

Where Q is the number of queries in the set, q is the query for average precision. With this formula, we can explain mAP as follows: We calculate the AP values for a given query and then the average of all these AP values is the value that gives us one number, mAP. In this way, we can evaluate the performance of our model with a single number. This method is the most widely used evaluation metric in object detection algorithms [9].

III. RESEARCH METHODOLOGY

A. Dataset

The dataset used in this research is an image dataset of traffic signs in Indonesia [10] with a total of 2,096 image data with a dimension of 640x640 pixels extension .jpg consisting of 21 classes of prohibitions, instructions, warnings, and traffic lights. In this study, the division of the dataset is divided into two types, namely:

1) Training Dataset

Training datasets are used to train machine learning models. Training data is used in the learning process where machine learning algorithms optimize their functions based on the given features and labels. This is the stage where the model "learns" the patterns and relationships in the data.

2) Testing Dataset

The testing dataset is used to evaluate the final performance of the model after the training is complete. It provides an objective picture of how well the model will perform on completely new data. Once the model is trained and selected using the training dataset, the testing data is used to measure the evaluation metrics. This data is never seen by the model during training, thus providing a more accurate assessment of its ability to handle new data in the real world.

TABLE I
DATASET DIVISION

Dataset Division	**Number of data**
Training	1.468
Testing	628
Total	**2.096**

B. System Design

The system designed and used in this research aims to analyze the dataset so that it can be known that the data belongs to the appropriate traffic sign category. The following is an overview of the process of the system to be created:

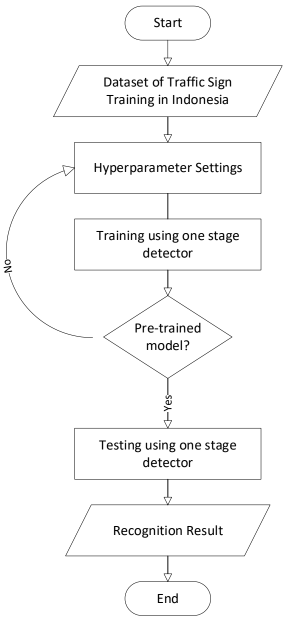
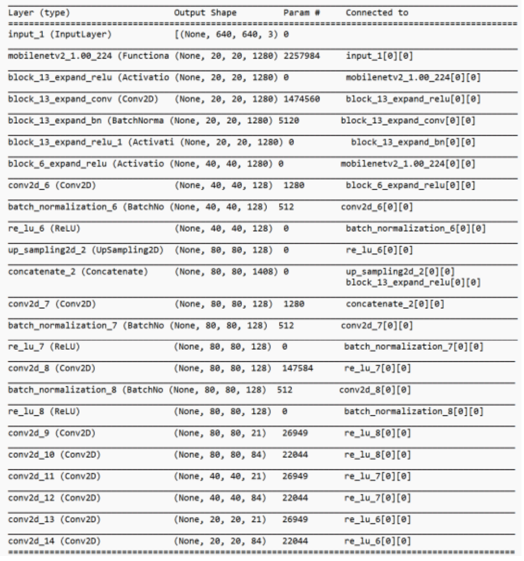

Start
Dataset of Traffic Sign Training in Indonesia
Hyperparameter Settings
Training using one stage detector
Pre-trained model?
No
Yes
Testing using one stage detector
Recognition Result
End

Fig. 3 *System Design Flowchart*

C. SSD Model Architecture

The architecture of the SSD model used in this study is shown in Fig. 4.



```
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 640, 640, 3) 0

mobilenetv2_1.00_224 (Functiona (None, 20, 20, 1280) 2257984     input_1[0][0]

block_13_expand_relu (Activatio (None, 20, 20, 1280) 0           mobilenetv2_1.00_224[0][0]

block_13_expand_conv (Conv2D)   (None, 20, 20, 1280) 1474560     block_13_expand_relu[0][0]

block_13_expand_bn (BatchNorma (None, 20, 20, 1280) 5120        block_13_expand_conv[0][0]

block_13_expand_relu_1 (Activati (None, 20, 20, 1280) 0          block_13_expand_bn[0][0]

block_6_expand_relu (Activatio (None, 40, 40, 1280) 0           mobilenetv2_1.00_224[0][0]

conv2d_6 (Conv2D)               (None, 40, 40, 128)  1280        block_6_expand_relu[0][0]

batch_normalization_6 (BatchNo (None, 40, 40, 128)  512         conv2d_6[0][0]

re_lu_6 (ReLU)                  (None, 40, 40, 128)  0           batch_normalization_6[0][0]

up_sampling2d_2 (UpSampling2D)  (None, 80, 80, 128)  0           re_lu_6[0][0]

concatenate_2 (Concatenate)     (None, 80, 80, 1408) 0           up_sampling2d_2[0][0]
                                                                 block_13_expand_relu[0][0]

conv2d_7 (Conv2D)               (None, 80, 80, 128)  1280        concatenate_2[0][0]

batch_normalization_7 (BatchNo (None, 80, 80, 128)  512         conv2d_7[0][0]

re_lu_7 (ReLU)                  (None, 80, 80, 128)  0           batch_normalization_7[0][0]

conv2d_8 (Conv2D)               (None, 80, 80, 128)  147584      re_lu_7[0][0]

batch_normalization_8 (BatchNo (None, 80, 80, 128)  512         conv2d_8[0][0]

re_lu_8 (ReLU)                  (None, 80, 80, 128)  0           batch_normalization_8[0][0]

conv2d_9 (Conv2D)               (None, 80, 80, 21)   26949       re_lu_8[0][0]

conv2d_10 (Conv2D)              (None, 80, 80, 84)   22044       re_lu_8[0][0]

conv2d_11 (Conv2D)              (None, 40, 40, 21)   26949       re_lu_7[0][0]

conv2d_12 (Conv2D)              (None, 40, 40, 84)   22044       re_lu_7[0][0]

conv2d_13 (Conv2D)              (None, 20, 20, 21)   26949       re_lu_6[0][0]

conv2d_14 (Conv2D)              (None, 20, 20, 84)   22044       re_lu_6[0][0]
==================================================================================================
```

Fig. 4 *SSD Model Architecture*

IV. RESULTS AND DISCUSSION

This research aims to analyze the mAP results obtained from the detector-based deep learning method in recognizing traffic signs in Indonesia. The following are the results and discussion of this research:

A. Testing Results against Real Data

Here are some results of testing the SSD model on real data sourced from outside the dataset. Researchers searched for three complex traffic sign image data sourced from internet searches.

From the results, it was found that the SSD model has an average detection time of 1.546 seconds. The detection results of the 3 complex real data found by the researcher can be seen in Fig. 5, Fig. 6, and Fig. 7.

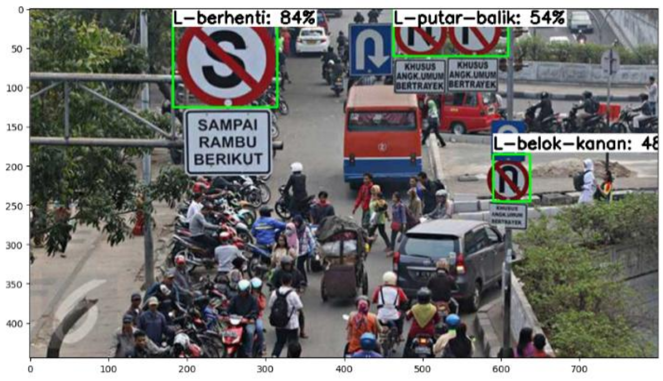


Fig. 5 *Testing Result of Data Real 1*

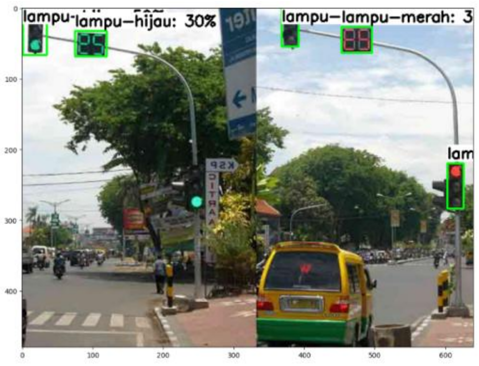


Fig. 6 *Testing Result of Data Real 2*

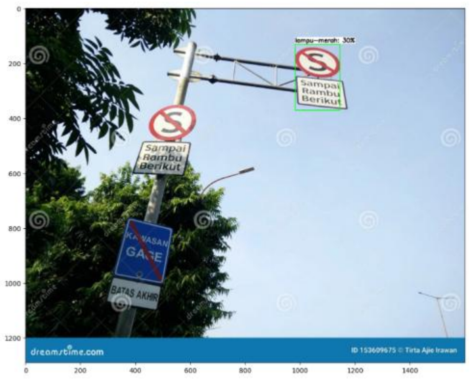


Fig. 7 *Testing Result of Data Real 3*

B. Evaluation Results

From the test results, an evaluation of the model that has been built is carried out. Fig. 8 shows the confusion matrix of the SSD model evaluation with the class labels as shown in Table II.

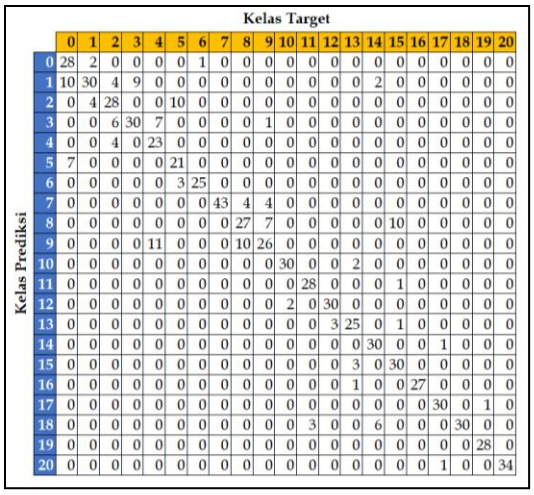
In this study, the evaluation metric used is Mean Average Precision (mAP). Table III and Table IV show the mAP generated from the SSD model using a step count of 4,000 and 10,000, respectively.

C. Analysis of Model Evaluation Results

From the results of the model evaluation, an analysis was conducted to determine the best performance produced by the model in detecting traffic sign objects in Indonesia. A comparison is made of the time required to build the model and the resulting mAP. In this research, a comparison of hyperparameter settings is carried out, namely the number of steps and epochs in building the model so that it can be seen whether hyperparameters affect the performance and time required to build the model. Performance comparison on the SSD model can be seen in Table V.

TABLE II
CLASS LABEL CAPTION CONFUSION MATRIX

Class	Class Label Confusion Matrix
L-belok-kanan	0
L-belok-kiri	1
L-berhenti	2
L-berjalan-terus	3
L-masuk	4
L-parkir	5
L-putar-balik	6
lampu-hijau	7
lampu-kuning	8
lampu-merah	9
p-area-parkir	10
p-isyarat	11
p-masuk-jalur	12
p-masuk-kiri	13
p-pemberhentian-bus	14
p-penegasan	15
p-penyeberangan	16
p-perlintasan-kereta	17
p-putar-balik	18
p-simpang-tiga	19
p-*zebra-cross*	20



Kelas Target

Kelas Prediksi	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	28	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	10	30	4	9	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
2	0	4	28	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	6	30	7	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	0	0	4	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	7	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	3	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	43	4	4	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	27	7	0	0	0	0	0	10	0	0	0	0	0
9	0	0	0	0	11	0	0	0	10	26	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	30	0	0	2	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	28	0	0	0	1	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	2	0	30	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	3	25	0	1	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	1	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	30	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	27	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	1	0
18	0	0	0	0	0	0	0	0	0	0	0	3	0	0	6	0	0	0	30	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	34

Fig. 8 *Confusion Matrix* SSD

TABLE III
mAP RESULT ON SSD MODEL (STEP = 4.000)

Class	mAP 50 (%)	mAP 50-95 (%)
L-belok-kanan	96,55	78,56
L-belok-kiri	86,81	66,40
L-berhenti	92,76	76,81
L-berjalan-terus	87,30	52,16
L-masuk	76,28	65,36
L-parkir	62,11	45,47
L-putar-balik	83,21	60,09
lampu-hijau	89,67	59,45
lampu-kuning	87,89	51,46
lampu-merah	68,12	37,79
p-area-parkir	100,00	67,88
p-isyarat	93,22	68,51
p-masuk-jalur	100,00	80,02
p-masuk-kiri	82,69	68,84
p-pemberhentian-bus	99,89	76,83
p-penegasan	99,58	77,65
p-penyeberangan	89,76	62,42
p-perlintasan-kereta	100,00	69,26
p-putar-balik	99,44	73,90
p-simpang-tiga	93,33	69,81
p-*zebra-cross*	94,29	66,53
Overall	**89,66**	**65,49**

TABLE IV
mAP RESULT ON SSD MODEL (STEP = 10.000)

Class	mAP 50 (%)	mAP 50-95 (%)
L-belok-kanan	1,25	0,50
L-belok-kiri	0,76	0,16
L-berhenti	0,00	0,00
L-berjalan-terus	0,00	0,00
L-masuk	0,00	0,00
L-parkir	0,00	0,00
L-putar-balik	0,00	0,00
lampu-hijau	4,17	0,78
lampu-kuning	0,00	0,00
lampu-merah	0,00	0,00
p-area-parkir	0,00	0,00
p-isyarat	0,00	0,00
p-masuk-jalur	0,00	0,00
p-masuk-kiri	0,00	0,00
p-pemberhentian-bus	0,00	0,00
p-penegasan	0,00	0,00
p-penyeberangan	0,00	0,00
p-perlintasan-kereta	0,00	0,00
p-putar-balik	0,00	0,00
p-simpang-tiga	0,00	0,00
p-*zebra-cross*	3,22	0,89
Overall	**0,45**	**0,11**

From Table V, it can be seen that the SSD model of scenario 1 has a faster training time and a much better mAP than the SSD model built with scenario 2. This shows that if the SSD model is built using too high a number of steps, it can cause overfitting. It is said to be overfitting because the machine learning model built is too "fit" to the training dataset, so when testing the model using data outside the training dataset, it cannot predict accurately. The SSD model obtained mAP 50 and mAP 50-95 values of 99.50% and 99.01% respectively.

TABLE V
PERFORMANCE COMPARISON ON SSD MODEL

Scenario	Num of Step	Training Time	mAP 50 (%)	mAP 50-95 (%)
1	4.000 [11]	3 hours 25 minutes	89,66	65,49
2	10.000 [11]	12 hours 45 minutes	0,45	0,11

V. CONCLUSIONS

From the results of this study, it is concluded that:

1) The detector-based deep learning method (in this research, the SSD model is used) is able to recognize traffic signs in Indonesia which are divided into 21 classes through the learning or training stage, then the results of the learning that have the best performance are stored in a model that can detect traffic signs objects in image data.
2) Traffic sign recognition using a detector-based deep learning method which in this research uses the SSD (Single Shot MultiBox Detector) model with a total dataset of 2,096 traffic sign images divided into 21 classes produces mAP 50 and mAP 50-95 values of 89.66% and 65.49% respectively for the SSD model.

REFERENCES

[1] T. Andari, R. Rosidah, P. Purwadi, H. Y. Harefa, and B. Hertasning, "Pengenalan Rambu Lalu Lintas pada Anak Usia Dini: Pendekatan Metode Vosviewer dalam Kajian Literatur Sistematis," *J. Obs. J. Pendidik. Anak Usia Dini*, vol. 7, no. 6, pp. 7743–7754, 2023, doi: 10.31004/obsesi.v7i6.5325.

[2] R. K. Paavani, V. Indraja, V. Neelimajyothi, S. Sai, and M. M. Sriramulu, "Traffic Sign Board Detection Using Single Shot Detection (SSD)," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 5, pp. 4095–4097, 2022,

[3] Z. Wang *et al.*, "Detecting Everything in the Open World: Towards Universal Object Detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2023-June, pp. 11433–11443, 2023, doi: 10.1109/CVPR52729.2023.01100.

[4] C. Feng and M. R. Scott, "TOOD : Task-aligned One-stage Object Detection," in *2021 IEEE/CVF International Conference on Computer Vision(ICCV)*, 2021, pp. 3510–3519.

[5] K. Fujii and K. Kawamoto, "Generative and self-supervised domain adaptation for one-stage object detection," *Array*, vol. 11, p. 100071, 2021, doi: 10.1016/j.array.2021.100071.

[6] T. Wang and D. Lin, "FCOS3D : Fully Convolutional One-Stage Monocular 3D Object Detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 913–922.

[7] H. Zhang and R. S. Cloutier, "Review on One-Stage Object Detection Based on Deep Learning," *EAI Endorsed Trans. e-Learning*, vol. 7, no. 23, p. 174181, 2022, doi: 10.4108/eai.9-6-2022.174181.

[8] I. P. Sary, E. U. Armin, and S. Andromeda, "Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection Using Aerial Images," *J. Sist. Komput.*, vol. 15, no. 1, pp. 1–6, 2023.

[9] I. Pacal, D. Karaboga, A. Basturk, B. Akay, and U. Nalbantoglu, "A comprehensive review of deep learning in colon cancer," *Comput. Biol. Med.*, vol. 126, no. April, p. 104003, 2020, doi: 10.1016/j.compbiomed.2020.104003.

[10] UMY, "Traffic sign in indonesia Dataset," *Roboflow Universe*. Roboflow, 2022. [Online]. Available: https://universe.roboflow.com/umy-35d0e/traffic-sign-in-indonesia-1j13y

[11] J. Howard and S. Gugger, *Practical Deep Learning for Coders*. Sebastopol: O'Reilly Media, 2020.