Prediction of Traffic Density Using YOLO Object Detection and Implemented in Raspberry Pi 3b + and Intel NCS 2

Rosa Andrie Asmara
Information Technology Department
State Polytechnic of Malang
Malang, Indonesia
rosa.andrie@polinema.ac.id

Bimo Syahputro
Information Technology Department
State Polytechnic of Malang
Malang, Indonesia
bimosyahputro88@gmail.com

Dodit Supriyanto
Information Technology Department
State Polytechnic of Malang
Malang, Indonesia
dodit.suprianto@polinema.ac.id

Anik Nur Handayani
Electrical Engineering Department
State University of Malang
Malang, Indonesia
aniknur.ft@um.ac.id

Abstract — **Increasing the number of vehicles as the population increases causes traffic congestion, thus indirectly hampering community activities. One way to unravel traffic jams is to do activities outside of peak hours. This study discusses how to determine vehicle traffic density using the Double Exponential Smoothing forecasting model, which is to estimate in quantity, what will happen in the future based on relevant data in the past. While testing the forecasting model using the MAPE (Mean Absolute Percentage Error) method, it aims to measure how the error rate obtained from the results of forecasting. Vehicle identification as the object causing the traffic jam uses the Convolutional Neural Network (CNN) object detection method, especially the You Only Look Once (YOLO) method. The system design involves a Raspberry Pi 3 device, an Intel NCS 2 device, and a website application. Based on the test results obtained, accurate forecasting accuracy of more than 86%.**

Keywords— Information Systems, Object Detection, Forecasting, YOLO, MAPE, CNN

I. INTRODUCTION

Traffic is a medium to move from one place to another. The role of traffic is crucial to support human mobility. If congestion occurs, mobility will be hampered, so that it can harm other sectors. The cause of congestion is caused by 1) Increasing population and vehicles (Wibowo, Soesanti, & Widyawan, 2017) and 2) Increasing the number of vehicles but not offset by road widening, causing traffic density (Esesiawati, 2017). Previous research related with traffic conduct in our lab experiments mostly used features for classification or recognition. This could results in high computation for feature extraction and longer time for dataset labelling (C. Rahmad, 2018), (R. A. Asmara et al.,2019) .

One way to find out traffic density that will occur is by forecasting. Forecasting is an attempt to predict the situation in the future by testing the situation in the past. The purpose of predicting data is to reduce uncertainty and make better estimates of what will happen in the future (Setyarini & Bulan, 2014). In this study, forecasting is applied to predict traffic density, which aims to predict vehicle density in the coming days.

The forecasting process uses multiple object data, including cars, motorbikes, bicycles, and people captured by the camera. At the same time, the camera is connected to the Raspberry Pi B3 + and Intel NCS 2 devices. The location of the data collection was conducted in front of the Malang State Polytechnic campus at 8:10 am, 8:20 am, and 8:30 am. In real-time, the camera will detect the vehicle through the YOLO Object Detection implementation that is installed on the Raspberry Pi B3 +, then counts and sends the number of cars, motorcycles, bicycles, and people to the server computer periodically. The data that has been obtained is processed by the Double Exponential Smoothing method. It is hoped that this research will be able to assist related parties to estimate the flow of traffic that will occur in the future.

II. RELATED WORKS

A. YOLO Object Detection

There are several object detection algorithms with different capabilities. These algorithms are mostly split into two groups according to how they perform their tasks. The first group is composed of algorithms based on classification and work in two stages. First, they select the exciting parts of the image, and then they classify objects within those regions using Convolutional Neural Networks (CNN). This group, which includes solutions such as R-CNN, is usually too slow to be applied in real-time situations[1]. The algorithms in the second group are based on regression - they scan the whole image and make predictions to localize, identify and classify objects within the image. Algorithms in this group, such as You Only Look Once (YOLO), are faster and can be used for real-time object detection[1], [2].

Therefore, in this study we use the YOLO object detection model to detect objects in a rel-time manner because the processing speed and accuracy in detecting objects are very high[1]–[4].

B. Double Exponential Smoothing

After object detection, the system then calculates the objects captured by the camera as a data source, as a basis for predicting the level of traffic density in the future. Forecasting technique uses the Double Exponential Smoothing model which is one type of Exponential Smoothing forecasting. This procedure continuously improves forecasting by averaging the past value of a time series data in an exponential manner. Exponential smoothing analysis is a time series analysis, and is a forecasting method by giving a weighting value to a series of previous observations to predict future values [5].

978-1-7281-8131-8/20/$31.00 ©2020 IEEE

Some researchers have proven that forecasting with the Double Exponential Smoothing method produces more satisfying performance than Single Exponential Smoothing [6][7]. This method is a linear model invented by Brown. It is a smoothing process twice. The advantage of this method is to model trends and levels of a time series that is more efficient than other methods because it requires fewer data and uses one parameter so that it becomes more straightforward. But the drawback of this method requires time in finding the most optimal α (alpha) value.

Determine the first Smoothing value

$$S'_t = (\,.\,X_t + (1-\alpha)\,S'_{t-1}$$

Determine the second Smoothing value

$$S''_t = \alpha . S'_t + (1-\alpha)\,S''_{t-1}$$

Determine the constant value α (alpha)

$$a_t = 2S'_t - S''_t$$

Determine the value of slope/trend

$$b_t = \frac{\alpha}{1-\alpha}\,(S'_t - S''_t)$$

Determine the forecast value

$$S_{t+m} = a_t + b_t\,m$$

Explanation of equation:

- S_{t+m} = Prediction value for the next period *m*
- m = distance of the period to be predicted
- X_t = *t*-actual period value
- S'_t = *t*-period smoothing value
- α = Smoothing Constant (1 / *n*)

C. Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) shows how much the forecast error rate compared to the actual value. MAPE is useful for measuring the accuracy of the model's alleged value expressed in the average absolute percentage of error (Robial, 2018).

The interpretation of the MAPE value is as follows:

- <10% = very accurate forecasting
- 10% - 20% = accurate forecasting
- 20% - 50% = forecasting is quite accurate
- \> 50% = forecasting is not accurate

The MAPE calculation shows the accuracy of forecasting percentage by determining PE (Percentage Error) or Percentage Error. PE itself to determine the rate of error in forecasting.

The following formula for calculating PE and MAPE:

- PE = $\frac{Xt-Ft}{Xt}\ x\ 100$
- MAPE = $\sum_{t=1}^{n} \frac{PEt}{n}$

Explanation of equation:

n = value of a certain period

X_t = true value in the t-period

F_t = forecast value in the t-period

Based on the MAPE calculation results, the lower the MAPE value, the better the performance of the forecasting model, conversely, the higher the MAPE value, the worse the performance of the forecasting model.

D. Hardware

In this study, we use a single-board Raspberry Pi circuit to detect vehicle objects from the camera. This aims to make it easy for us to place the camera sensor, as well as the placement of the computer as the processor even in an open and narrow area. The advantage of Raspberry Pi is its small size which is the size of a credit card. But on the other hand, Raspberry pi has low resources (processor, memory, etc.) so that if it is charged with heavy tasks such as YOLO Deep Learning processing it won't be able to do it [8], [9]. Therefore researchers combine it with Intel NCS 2 devices. Intel NCS 2 uses the latest generation Intel VPU, namely Intel Movidius Myriad X, which in this version has been supported by Intel distribution of the OpenVINO toolkit. Intel itself claims that the NCS 2 Intel is more easily tested, configured, and used for the development of deep neural network prototypes, with 8x better performance so as to provide improved performance compared to previous generation neural compute sticks and only for USB. Intel also said that the vision accelerator on Intel NCS 2 could make it easier for developers to develop smart cameras, industrial robots, and the Internet of Things (IoT).

III. METHODOLOGY

A. System Description

The system architecture built shown in Figure 1. Raspberry Pi is the main system to control the acquire camera image. Object detection will be done by Yolo Model run in Intel NCS2. Vehicle amount value will be send to the Web application and the information can be seen from PC

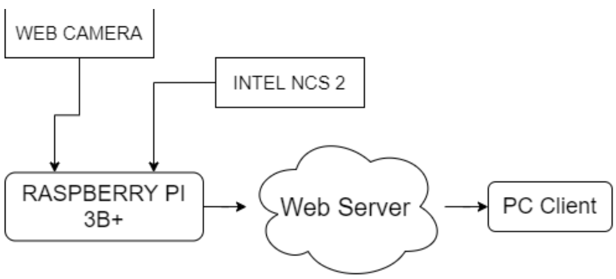

Fig 1. Architecture System

user.

We build forecasting applications with a web platform, where data collection is done at 8:10, 8:20, and 8:30 in the morning. The data will be used to predict future events by using past data collected using the YOLO algorithm. This algorithm is useful for detecting vehicles passing in front of the main door of Malang State Polytechnic. YOLO is installed on a Raspberry Pi 3B + device that is connected to the webcam as a camera to detect objects. At the same time, Intel NCS 2 is useful as a device to speed up the calculation of the Deep Learning model (YOLO) on computers that have low resources such as Raspberry Pi.

B. System Analysis

The stage of developing a vehicle density forecasting system is shown in Figure 2 and the development of a traffic density forecasting system based on data obtained from Raspberry Pi 3B + will follow the following stages:

- The initial phase is the initial process for obtaining forecasting data. Data will be collected for 13 days at the same place and hour, between 8:00 to 8:30 in the morning.
- Forecasting is the second stage after data collection. The collected data will be processed using the Double Exponential Smoothing method.
- Testing is the system testing stage produced by calculating the level of forecasting accuracy and the level of detection accuracy. Forecasting accuracy will be calculated using MAPE by finding the smallest value on a particular alpha. In contrast, the level of detection accuracy will be compared to the number of vehicles detected by calculating the number of cars manually.

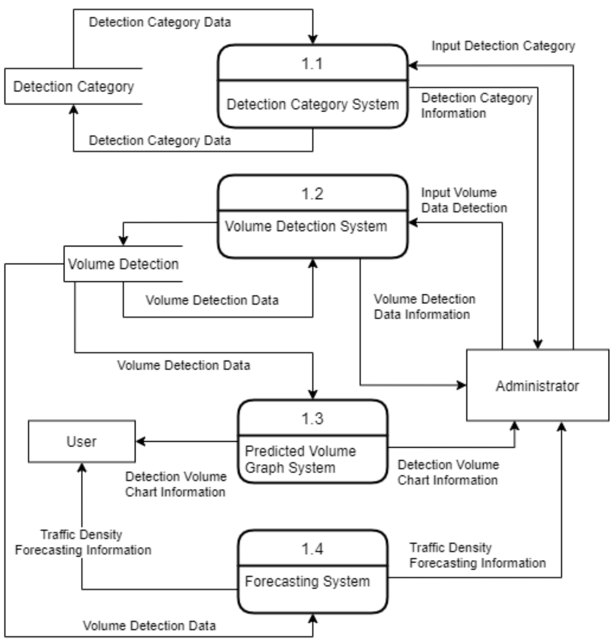

Fig 2. Development Phase Application Forecasting

C. Application Interface

The forecasting application interface design is as follows:

- Dashboard, dashboard page is a display of vehicle volume data in the form of a graph, as shown in Figure 3.
- Forecasting, the forecasting page is the result page of the forecasting of each vehicle. Next is the

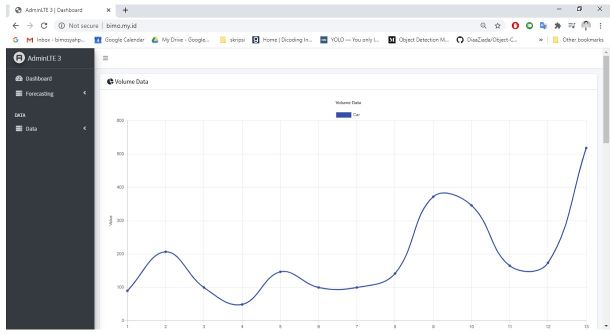

Fig 3. Dashboard Page

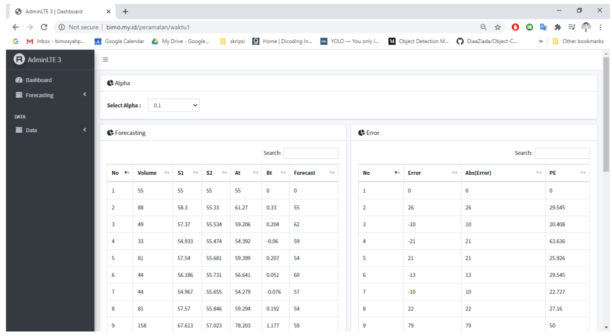

Fig 4. Forecasting Page

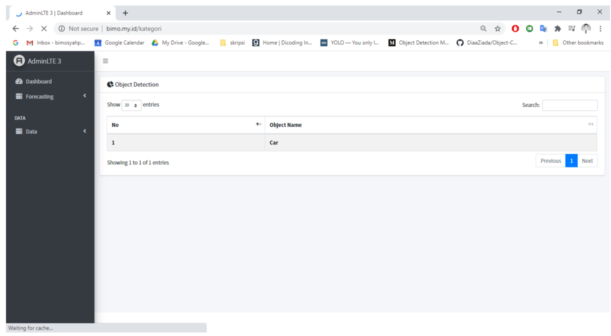

Fig 5. Category Data Page

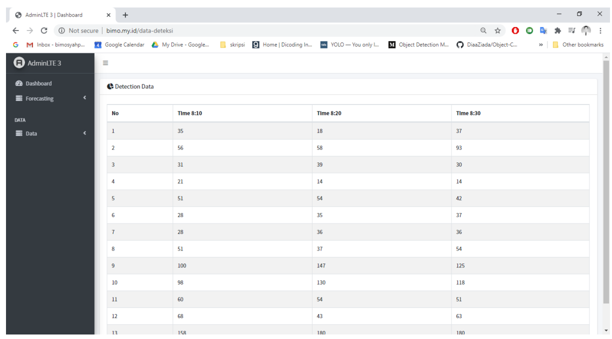

Fig 6. Vehicle Volume Data Page

implementation of the vehicle volume forecasting results, as shown in Figure 4.

- Category Data, Category data page is an object detection category page that will be used as a forecasting process, as shown in Figure 5.
- Vehicle Volume Data, the volume data page is the vehicle volume data sent by Raspberry Pi 3B + and shows the total amount per day, as shown in Figure 6.
- Figure 7 is an object detection application display on the Raspberry Pi side, where YOLO object detection is detecting a car that is passing and is captured by the camera.

IV. RESULT ANALYSIS

In this section, an analysis of the results of traffic density forecasting is done by testing the vehicle detection and then proceeding with checking the prediction itself.

- Detection Testing

  YOLO object detection test by comparing application calculations with manual calculations. Data retrieval time is 13 days, and each session takes 3x10 minutes (total of 30 minutes per day). The following Table I is an analysis of detection testing.

TABLE I. TABLE TYPE STYLES

No	Accuracy								
	YOLO			*Real Results*			*Accuracy*		
	8.10	8.20	8.30	8.10	8.20	8.30	8.10	8.20	8.30
1	35	18	37	46	51	44	76%	35%	84%
2	56	58	93	31	34	25	181%	171%	372%
3	31	39	30	83	85	68	37%	46%	44%
4	21	14	14	40	52	54	53%	27%	26%
5	51	54	42	60	64	59	85%	84%	71%
6	28	35	37	68	54	64	41%	65%	58%
7	28	36	36	70	76	77	40%	47%	47%
8	51	37	54	100	76	77	51%	49%	70%
9	100	147	125	120	131	80	83%	112%	156%
10	98	130	118	104	136	128	94%	96%	92%
11	60	54	51	79	100	99	76%	54%	52%
12	68	43	63	96	93	110	71%	46%	57%
13	158	180	180	167	187	181	95%	96%	99%
Average every time							76%	71%	95%
Average accuracy							81%		

Fig 7. Object Detection Application

TABLE II. TABLE TYPE STYLES

Alpha	Application			Manual		
	08:10	*08:20*	*08:30*	*08:10*	*08:20*	*08:30*
0.1	133	167	147	133.13	167.18	147.65
0.2	177	218	194	177.19	218.56	194.35
0.3	205	245	224	205.18	245.56	224.54
0.4	227	269	250	227.25	270.06	250.39
0.5	249	300	278	249.49	301.28	278.91
0.6	275	341	312	274.77	342.62	313.1
0.7	304	392	352	303.41	393.58	352.58
0.8	334	449	394	334.03	450.91	394.72
0.9	365	508	435	364.41	509.88	436.11

TABLE III. TABLE TYPE STYLES

Time	*Alpha*	*Value*
8:10	0.5	7.727%
8:20	0.6	11.971%
8:30	0.5	12.159%

- Forcasting Testing

  In Table II, it appears that the results of forecasting comparison using the Double Exponential Smoothing method in the application and the manual are appropriate. Best Alpha Recommendations

  The best Alpha recommendation result is to find the smallest MAPE value of each alpha value. A list of the best alpha results for 8:10, 8:20 and 8:30 appears in Table III

V. CONCLUSION

Based on the prediction results of traffic density using YOLO object detection applied to Raspberry pi 3b + and Intel NCS 2, it can be concluded as follows:

- After testing and testing comparing the calculation results of the system with manual calculations, the Double Exponential Smoothing method has been successfully applied to this system and can be used as predictive information.

- Forecasting results are entirely accurate because MAPE values are in the range of 7% to 14%. In other words, the level of detection accuracy reaches an average of 81%.

REFERENCES

[1] B. Hardjono, A. E. Widjaja, R. Kondorura, and A. M. Halim, "Deep Learning Methods," *2018 IEEE 9th Annu. Inf. Technol. Electron. Mob. Commun. Conf.*, pp. 556–562, 2018.

[2] L. Lou, Q. Zhang, C. Liu, M. Sheng, Y. Zheng, and X. Liu, "Vehicles detection of traffic flow video using deep learning," *Proc. 2019 IEEE 8th Data Driven Control Learn. Syst. Conf. DDCLS 2019*, pp. 1012–1017, 2019, doi: 10.1109/DDCLS.2019.8908873.

[3] J. P. Lin and M. Te Sun, "A YOLO-Based Traffic Counting System," *Proc. - 2018 Conf. Technol. Appl. Artif. Intell. TAAI 2018*, pp. 82–85, 2018, doi: 10.1109/TAAI.2018.00027.

[4] A. Forero and F. Calderon, "Vehicle and pedestrian video-tracking with classification based on deep convolutional neural networks," *2019 22nd Symp. Image, Signal Process. Artif. Vision, STSIVA 2019 - Conf. Proc.*, pp. 1–5, 2019, doi: 10.1109/STSIVA.2019.8730234.

[5] A. C. Adamuthe, R. A. Gage, and G. T. Thampi, "Forecasting cloud computing using double exponential smoothing methods," *ICACCS 2015 - Proc. 2nd Int. Conf. Adv. Comput. Commun. Syst.*, pp. 3–7, 2015, doi: 10.1109/ICACCS.2015.7324108.

[6] Y. Hu, H. Zhang, C. Li, S. Liu, and Y. Zhang, "Exponential smoothing model for condition monitoring: A case study," *QR2MSE 2013 - Proc. 2013 Int. Conf. Qual. Reliab. Risk, Maintenance, Saf. Eng.*, pp. 1742–1746, 2013, doi: 10.1109/QR2MSE.2013.6625913.

[7] C. Wang, "Quantitative analysis on the bullwhip effect in a supply chain using double moving average and double exponential smoothing forecasts," *Proc. - Int. Symp. Inf. Process. ISIP 2008 Int. Pacific Work. Web Min. Web-Based Appl. WMWA 2008*, pp. 114–118, 2008, doi: 10.1109/ISIP.2008.32.

[8] E. S. Lage, R. L. Santos, S. M. T. Junior, and F. Andreotti, "Low-cost iot surveillance system using hardware-Acceleration and convolutional neural networks," *IEEE 5th World Forum Internet Things, WF-IoT 2019 - Conf. Proc.*, pp. 931–936, 2019, doi: 10.1109/WF-IoT.2019.8767325.

[9] X. Xu, J. Amaro, S. Caulfield, A. Forembski, G. Falcao, and D. Moloney, "Convolutional neural network on neural compute stick for voxelized point-clouds classification," *Proc. - 2017 10th Int. Congr. Image Signal Process. Biomed. Eng. Informatics, CISP-BMEI 2017*, vol. 2018-January, pp. 1–7, 2018, doi: 10.1109/CISP-BMEI.2017.8302078.

[10] C. Rahmad, I. F. Rahmah, R. A. Asmara and S. Adhisuwignjo, "Indonesian traffic sign detection and recognition using color and texture feature extraction and SVM classifier," 2018 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, 2018, pp. 50-55

[11] R. A. Asmara et al., "Comparison of Vehicle Detection Using Haar-like Feature, LBP and HOG Technique for Feature Extraction in Cascade Classifier", IJAST, vol. 28, no. 8s, pp. 834 - 838, Oct. 2019.