BIS INFORMATION TECHNOLOGY AND COMPUTER SCIENCE

Volume 1 2024 | DOI: 10.31603/bistycs.123
5th Borobudur International Symposium on Science and Technology (BIS-STE) 2023

UNIMMA PRESS
Universitas Muhammadiyah Magelang

Improved vehicle detection accuracy using CLAHE

A Widiyanto[1], S Nugroho[2*] and M R A Yudianto[2]

[1] Department of Information Technology, Universitas Muhammadiyah Magelang, Magelang, Indonesia
[2] Department of Informatic Engineering, Universitas Muhammadiyah Magelang, Magelang, Indonesia
[*]Corresponding author email: setiya@ummgl.ac.id

Abstract

The large number of vehicles can cause new problems in various fields. Vehicle detection errors can occur in the vehicle detection system when several vehicles are side by side so that they are not detected or are detected as larger vehicles. This research produces a vehicle type detection system to improve vehicle detection accuracy by applying image processing on Convolutional Neural Network (CNN). In this study, experiments were conducted with 20 image processing scenarios in the pre-processing image before the training process to produce an object detection testing model. The simulation test results show that not all image processing scenarios can improve the accuracy of the detection process. The combined image processing scenario of Blue Channel + CLAHE + gaussian filter + thresholding produces an accuracy of 97%.

Keywords

Vehicle detection accuracy, CLAHE, Image processing

Introduction

In 2018 the number of motorized vehicles in Indonesia according to data from the Biro Pusat Statistik (BPS) 146,858,759 experienced an average growth of 7.03% from the previous year [1]. The large number of vehicles can cause new problems in various fields such as environmental pollution problems [2] traffic [3], fuel wastage [4], parking management [5] and other problems.

Research on the detection of the number of vehicles has been carried out using ultrasonic sensors [6], digital image processing in the form of video [7–11], with vehicle speed detection [12] and with Internet of Things technology (IoT) [3][5][13].

In general, the process of detecting a moving object is the first step by obtaining a frame from a running video from either a live camera or a video recording camera to detect motion. The next step is to match the object with the vehicle type data based on the method used, for example with the size of a rectangle as shown in Figure 1 [14] or counting the number of pixels as shown in Figure 2 [12].

Previous research on vehicle detection even uses various detection technologies and methods when road conditions are congested, in addition to experiencing vehicle type

Published:
October 20, 2024

Selection and Peer-review under the responsibility of the 5th BIS-STE 2023 Committee

errors [11], or experiencing a decrease in accuracy when road conditions are increasingly congested [14]. This study focuses on the process of detecting vehicles that move hand in hand with the aim of producing a more accurate vehicle detection process.

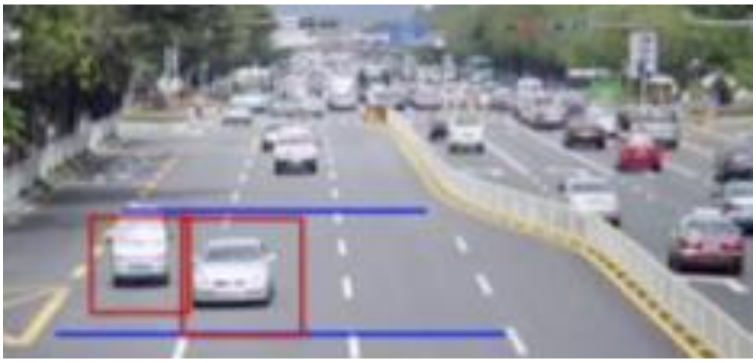


Figure 1. Object Detection Using Object Marker

Figure 2. Object Detection Using BLOB Tracking

Related Work

This study focuses on the pre-processing process, namely modifying the training data in such a way that it becomes a model that will be used in the object detection process.

Convolutional Neural Network (CNN)

CNN is specifically designed for handling multiple arrays as the input. This initial input process of CNN is analogous to imitating the eyes to identify images, followed by a training process for further recognition of the scene [15] The CNN workflow for image classification is to group the images based on their similarities, then identify objects in several samples.

Contrast Limited Adaptive Histogram Equalization (CLAHE)

It is a technique of improving image contrast by increasing the local contrast of the image giving a limit value to the histogram. This limit value is called the clip limit which states the maximum limit for the height of a histogram [16]. To calculate the clip limit, use the following formula 1.

$$\beta = \frac{M}{N}\left(1 + \frac{\alpha}{100}(s_{max} - 1)\right) \quad (1)$$

where: M = region size; N = grayscale value (256), and α = clip factor (0-100).

The histogram above the cliplimit value is considered to be excess (excess) pixels which will be distributed to the surrounding area below the cliplimit so that the histogram is evenly distributed [17].

Gaussian Filter

The Gaussian Filter used to filter the image before classification. linear filter with a weighted value for each member is elected based on the shape of the Gaussian function. This filter is very effective for removing noise that is normally distributed. To calculate the values of each element in the Gaussian filter that will be formed, it's following equation (2):

$$\frac{h(x,y)}{c} = \partial^{\frac{x^2+y^2}{2a^2}} \tag{2}$$

Where: ∂ = The width of the Gaussian function; c = Normalization constant; and h(x,y)= Convoluted image.

Thresholding

Thresholding is one of the popular techniques to segmenting images. Segmentation that uses the intensity parameter of each area to separate high-intensity (bright) areas from low-intensity (dark) areas. The Thresholding technique successfully detects on the images taken in uncontrol environment [18].

Method

The process flow of the object detection system in this study is described in the form of a flowchart as shown in Figure 3.

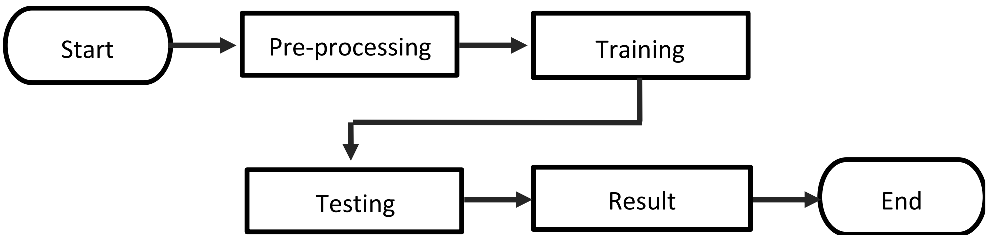


Figure 3. Flowchart of Object Detection Process

Convolutional Neural Network (CNN) architecture is the most important parts in this research. CNN has the advantage over conventional approaches to extract features individually, decrease dimensional of the data as well as the category in one network structure [19].

This section is a layer that functions to extract the features contained in an object. This layer will perform a convolutional operation, namely the process of multiplying the image matrix with the filter matrix in the feature extraction process. CNN is specifically designed for handling multiple arrays as the input. This initial input process of CNN is analogous to imitating the eyes to identify images, followed by a training process [15].

The input image dataset that has been conditioned with various possibilities according to the object of the problem. when 2 vehicles go hand in hand (Figure 4), the error that appears is not detected or detected as a larger vehicle, so that 3 types of input images are prepared, that's a car, 2 cars and bus.

Figure 4. Vehicles Go Hand in Hand

Pre-processing image consisting of channel extraction (red, green, blue channel), CLAHE, gaussian filter and thresholding. The Result of apply complement to channel extraction is presented in Figure 5 and Figure 6 shows the results of image processing.

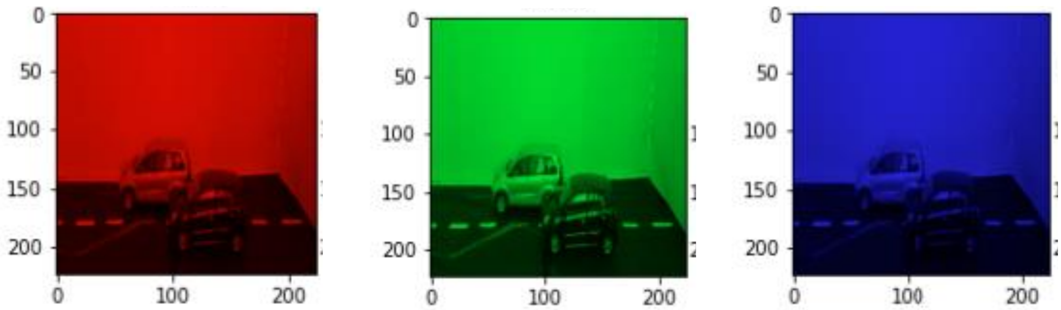


Figure 5. The Result of Apply Complement to Channel Extraction (a) Red Channel (b) Green Channel (c) Blue Channel

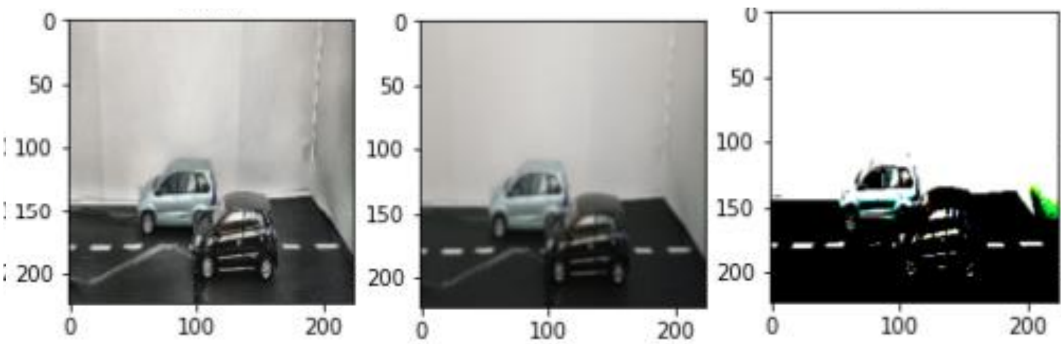


Figure 6. The Result of Image Processing (a) CLAHE (b) Gaussian Filter (c) Thresholding

Measurement of the level of object detection accuracy is based on the parameters used in measuring the level of object detection accuracy [20]:

1. True Positive (TP) is the actual object detected by the system
2. False Positive (FP) is the noise detected by the system as an object
3. False Negative (FN) is an actual object that is not detected by the system
4. Accuracy is the result of calculating the level of accuracy of object detection against vehicle objects in the whole frame that is carried out by the detection process. Formula 2 of the accuracy calculation can be seen in

$$Accuracy = \frac{true\ object\ detection}{total\ object\ detection} \tag{3}$$

5. Precision, is the number of correct predictions compared to the overall results predicted by the system. In this case, precision will answer how many objects are detected correctly from the total number of objects detected by the system. The formula for calculating precision can be seen in

$$Precision = \frac{TP}{(TP+FP)} \tag{4}$$

6. Recall is the number of correct predictions compared to the overall actual results. In this case Recall will answer how many objects are detected correctly from the total number of actual objects. The formula for the recall calculation can be seen in.

$$Recall = \frac{TP}{(TP+FN)} \tag{5}$$

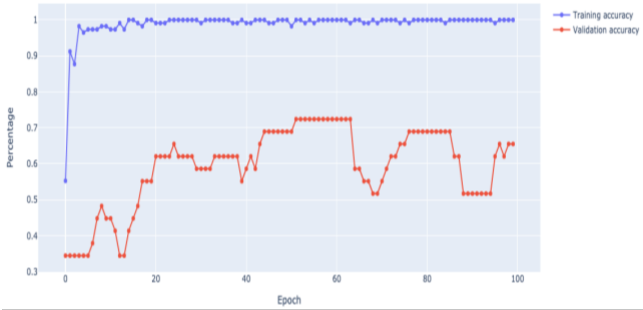
Finally, by testing and comparing the vehicle detection accuracy of all image processing scenarios, better results are obtained (Table 1).

Table 1. Experiment scenario

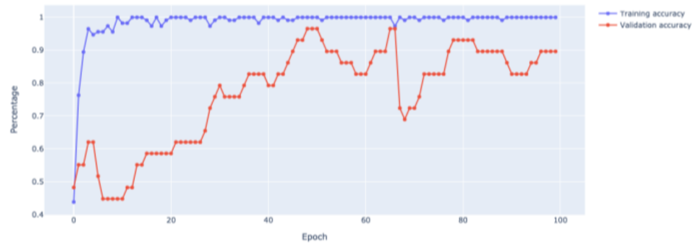
No	Scenario	Information
1	S1	Red Channel + CLAHE
2	S2	Red Channel + CLAHE + gaussian
3	S3	Red Channel + CLAHE + thresholding
4	S4	Red Channel + CLAHE + gaussian + thresholding
5	S5	Green Channel + CLAHE
6	S6	Green Channel + CLAHE + gaussian
7	S7	Green Channel + CLAHE + thresholding
8	S8	Green Channel + CLAHE + gaussian + thresholding
9	S9	Blue Channel + CLAHE
10	S10	Blue Channel + CLAHE + gaussian
11	S11	Blue Channel + CLAHE + thresholding
12	S12	Blue Channel + CLAHE + gaussian + thresholding
13	S13	RGB + CLAHE
14	S14	RGB + CLAHE + gaussian
15	S15	RGB + CLAHE + thresholding
16	S16	RGB + CLAHE + gaussian + thresholding
17	S17	Red Channel
18	S18	Green Channel
19	S19	Blue Channel
20	S20	RGB

Results and Discussion

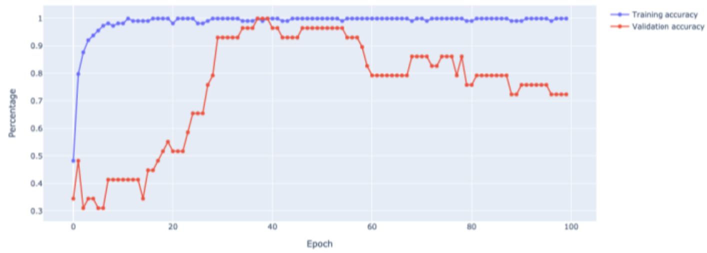
Simulation of the experimental scenario using google colab (link) with 100 epochs. The results of 20 Experiment scenario presented in Table 2. Figure 7 shows that the training accuracy for all scenarios is relatively the same. validation accuracy begins to increase at epoch 20 with a different level of accuracy in each scenario and at epoch 80 it begins to decrease. The comparison of the result experimental scenarios is shown in the form of a graph in Figure 8.



(a)



(b)



(c)

Figure 7. Graph of training & validation accuracy (a) RGB (S20) (b) Green Channel + CLAHE + thresholding (S7) (c) Blue Channel + CLAHE + gaussian + thresholding (S12).

Table 2. The result of experiment scenario

Code	Time (s)	Accuracy	Precision	Recall	F1-Score
S1	38.06 + 235.38	00.38	00.14	00.33	00.19
S2	33.88 + 203.13	00.31	00.43	00.37	00.21
S3	38.21 + 201.93	0.05486111	0.05972222	0.05694444	0.05416667
S4	35.5 + 231.38	0.04305556	0.05625	0.04583333	0.04166667
S5	38.06 + 185.34	00.38	00.45	00.36	00.23
S6	33.88 + 194.36	00.34	0.04166667	00.39	00.30
S7	38.21 + 192.81	0,0625	0.06319444	0.06319444	0.06180556
S8	35.5 + 160	00.45	00.44	00.48	00.38
S9	44.12 + 189.2	00.34	00.11	00.33	00.17
S10	39.35 + 187	00.24	00.28	00.23	00.23
S11	73.27 + 237.93	0,05	0.05972222	0.04722222	0.04652778
S12	35.5 + 155.31	0.06736111	0.06736111	0.06666667	0.06666667
S13	44.12 + 186.73	0.04583333	0,05	0.04375	0.04375
S14	39.35 + 154.38	00.28	00.09	00.33	00.14
S15	73.27 + 156.88	0.05486111	0.05625	0.05416667	0.05486111
S16	35.5 + 160.69	0.05763889	0.05972222	0.05902778	0.05694444
S17	44.12 + 159.23	00.41	00.22	00.38	00.26
S18	39.35 + 159.90	00.45	00.47	00.40	00.30
S19	73.27 + 158.8	00.55	0.04305556	00.59	00.53
S20	35.5 + 160.25	0.04583333	00.50	0.04236111	00.52

The S20 scenario is the standard scenario as a comparison to other scenarios. Image processing scenarios that use one of the techniques such as red channel, green channel, blue channel, CLAHE, gaussian filter or thresholding alone do not always improve object detection accuracy. The accuracy is quite good in S7 (Green Channel + CLAHE + Thresholding) and S12 (Blue Channel + CLAHE + gaussian filter + thresholding) scenarios above 90%. The combination of image processing scenarios at the pre-processing process results in a significant increase in accuracy at the object detection process. The test results show that the scenario of S12 Accuracy and Precision is the highest.

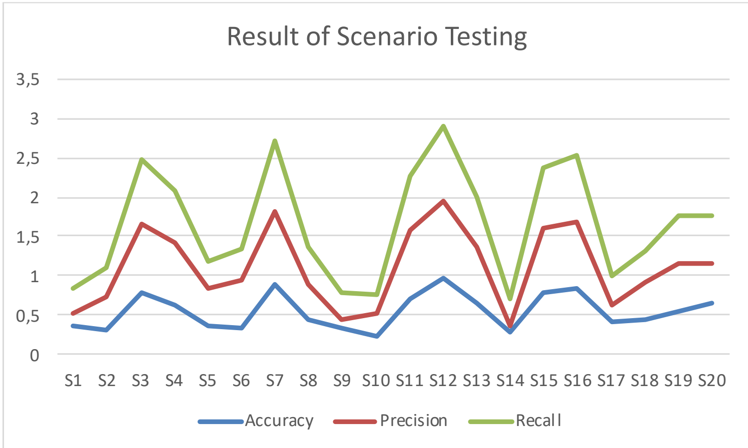


Figure 8. Graph of the result experimental scenarios

Conclusion

According to the simulation experimental results show that not all image processing scenarios can improve the accuracy of the detection process. The result of the best scenario is that the combined image processing of Blue Channel + CLAHE + gaussian filter + thresholding produces an accuracy of 97%.

Acknowledgement

This research received funding from LPPM Universitas Muhammadiyah Magelang. Our gratitude goes to the laboratory assistants in the Informatics Engineering laboratory of the Universitas Muhammadiyah Magelang.

References

[1] BPS Jumlah Kendaraan Bermotor (Unit), 2016-2018 Available online: https://www.bps.go.id/indicator/17/57/1/jumlah-kendaraan-bermotor.html (accessed on Oct 25, 2020).

[2] Roshintha, R.R.; Mangkoedihardjo, S. Analisis Kecukupan Ruang Terbuka Hijau Sebagai Penyerap Emisi Gas Karbon Dioksida (CO2) pada Kawasan Kampus ITS Sukolilo, Surabaya. J. Tek. ITS 2016, 5, doi:10.12962/j23373539.v5i2.17510.

[3] Limantara, A.D.; Candra, A..; Mudjanarko, S.W. Manajemen Data Lalu Lintas Kendaraan Berbasis Sistem Internet Cerdas Ujicoba Implementasi di Laboratorium Universitas Kadiri. In Proceedings of the Pros. Semnastek; 2017; Vol. 4, pp. 1–11.

[4] Prasetyo, B.A.; Rizani, D.A.; Setiyo, M.; Widodo, N.; Saifudin; Purnomo, B.C. Estimasi Pemborosan Bahan Bakar Akibat Kemacetan Menggunakan Analisis Citra Google Map (Studi Kasus pada Simpang Armada Town Square Mall Magelang). Automot. Exp. 2018, 1, 36–42.

[5] Hariyanto, M.S.; Sofwan, A.; Hidayatno, A. Perancangan Sistem Penghitung Jumlah Kendaraan Pada Area Parkir Dengan Metode Background Subtraction Berbasis Internet of Things. *Transient* **2019**, 7, 775, doi:10.14710/transient.7.3.775-781.

[6] Hardiyanto, R.D.; Rochim, A.F.; Windasari, I.P. Pembuatan Penghitung Jumlah Mobil Otomatis Berbasis Mikrokontroler ATMega 8535 Menggunakan Sensor Ultrasonik. *J. Teknol. dan Sist. Komput.* **2015**, 3, 185, doi:10.14710/jtsiskom.3.2.2015.185-191.
[7] Aynurrohmah, M.; Sunyoto, A. Penghitung Jumlah Mobil Menggunakan Pengolahan Citra Digital Dengan Input Video Digital. Data Manaj. dan Teknol. Inf. 2011, 12, 2–6.
[8] Mu'arifah, S.; Pratomo, A.H.; Kaswidjanti, W. Pengolahan Citra Untuk Klasifikasi dan Perhitungan Jumlah Kendaraan. In Proceedings of the SEMNASTIK; STMIK Binadarma: Palembang, 2016; pp. 771–782.
[9] Budiarto, J.; Qudsi, J. Deteksi Citra Kendaraan Berbasis Web Menggunakan Javascript Framework Library. MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput. 2018, 18, 125–133, doi:10.30812/matrik.v18i1.325.
[10] Adistya, R.; Muslim, M.A. Deteksi dan Klasifikasi Kendaraan menggunakan Algoritma Backpropagation dan Sobel. J. Mech. Eng. Mechatronics 2016, 1, 65–73.
[11] Purwiyanti, S.; Herlinawati; Murdika, U. Rancang Bangun Penghitung Jumlah Kendaraan Menggunakan Metode Tracking Feature Point Berbasis Raspberry. Fakultas Teknik Universitas Lampung, 2017.
[12] Setyawan, G.E.; Adiwijaya, B.; Fitriyah, H. Sistem Deteksi Jumlah, Jenis dan Kecepatan Kendaraan Menggunakan Analisa Blob Berbasis Raspberry Pi. J. Teknol. Inf. dan Ilmu Komput. 2019, 6, 211, doi:10.25126/jtiik.2019621405.
[13] Gozali, F.; Iskandardinata, R.; Subrata, R.H. Sistem Pemantauan Dan Perekaman Gerak Kendaraan Secara Nirkabel Dengan Menggunakan Raspberry Pi. J. Teknol. Elektro 2017, 8, 9, doi:10.22441/jte.v8i1.1365.
[14] Lazaro, A.; Buliali, J.L.; Amaliah, B. Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV. J. Tek. ITS 2017, 6, doi:10.12962/j23373539.v6i2.23175.
[15] Caesarendra, W.; Triwiyanto, T.; Pandiyan, V.; Glowacz, A.; Permana, S.D.H.; Tjahjowidodo, T. A CNN Prediction Method for Belt Grinding Tool Wear in a Polishing Process Utilizing 3-Axes Force and Vibration Data. Electronics 2021, 10, 1429, doi:10.3390/electronics10121429.
[16] Miranda, N.D.; Novamizanti, L.; Rizal, S.; Elektro, F.T.; Telkom, U. Convolutional Neural Network Pada Klasifikasi Sidik Jari Menggunakan Resnet-50 Classification of Fingerprint Pattern Using Convolutional Neural Network in Clahe Image. J. Tek. Inform. 2020, 1, 61–68.
[17] Nurzaenab; Hadis, M.S.; Angriawan, R. Nilai Optimal Clip Limit Metode Clahe Untuk Meningkatkan Akurasi Pengenalan Wajah Pada Video Cctv. J. INSTEK 2020, 5, 178–187.
[18] Abu Bakar, M.N.; Abdullah, A.H.; Abdul Rahim, N.; Yazid, H.; Misman, S.N.; Masnan, M.J. Rice leaf blast disease detection using multi-level colour image thresholdin. J. Telecommun. Electron. Comput. Eng. 2018, 10, 1–6.
[19] Wong, Y.C.; Lai, J.A.; Ranjit, S.S.S.; Syafeeza, A.R.; Hamid, N.A. Convolutional Neural Network for Object Detection System for Blind People. J. Telecommun. Electron. Comput. Eng. 2019, 11, 1–6.
[20] Manajang, D.; Dompie, S.; Jacobus, A. Implementasi Framework Tensorflow Object Detection Dalam Mengklasifikasi Jenis Kendaraan Bermotor. J. Tek. Inform. 2020, 15, 171–178.