9 (2) July 2023 122-133 ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
Register: *Jurnal Ilmiah Teknologi Sistem Informasi*

Contents lists available at www.journal.unipdu.ac.id

Register

Journal Page is available to www.journal.unipdu.ac.id/index.php/register

Research article

Detecting Objects Using Haar Cascade for Human Counting Implemented in OpenMV

Mustika Mentari [a,*], *Rosa Andrie Asmara* [b], *Kohei Arai* [c], *Haidar Sakti Oktafiansyah* [d]

[a,b,d] *Department of Information Technology, State Polytechnic of Malang, Malang, Indonesia*
[c] *Graduate School of Science and Engineering, Saga University, Saga, Japan*
email: [a,*] *must.mentari@polinema.ac.id*, [b] *rosa.andrie@polinema.ac.id*, [c] *arai@cc.saga-u.ac.jp*, [d] *haidaroktafiansyah@gmail.com*,
* *Correspondence*

ARTICLE INFO

Article history:
Received 12 December 2022
Revised 7 May 2023
Accepted 4 September 2023
Available online 30 September 2023

Keywords:
Object Detection;
Object Counting;
Haar Cascade Classifier;
OpenMV;
Human Object;

Please cite this article in IEEE style as:
M. Mentari, R. A. Asmara, K. Arai, and H. S. Oktafiansyah, "Detecting Objects Using Haar Cascade for Human Counting Implemented in OpenMV," *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 9, no. 2, pp. 122-133, 2023.

ABSTRACT

Sight is a fundamental sense for humans, and individuals with visual impairments often rely on assistance from others or tools that promote independence in performing various tasks. One crucial aspect of aiding visually impaired individuals involves the detection and counting of objects. This paper aims to develop a simulation tool designed to assist visually impaired individuals in detecting and counting human objects. The tool's implementation necessitates a synergy of both hardware and software components, with OpenMV serving as a central hardware device in this study. The research software was developed using the Haar Cascade Classifier algorithm. The research process commences with the acquisition of image data through the OpenMV camera. Subsequently, the image data undergoes several stages of processing, including the utilization of the Haar Cascade classifier method within the OpenMV framework. The resulting output consists of bounding boxes delineating the detection areas and the tally of identified human objects. The results of human object detection and counting using OpenMV exhibit an accuracy rate of 71%. Moreover, when applied to video footage, the OpenMV system yields a correct detection rate of 73% for counting human objects. In summary, this study presents a valuable tool that aids visually impaired individuals in the detection and counting of human objects, achieving commendable accuracy rates through the implementation of OpenMV and the Haar Cascade Classifier algorithm.

1. Introduction

Vision is one of the fundamental human senses that fosters survival instincts. It enables humans to effectively engage with their environment. When someone loses their sense of sight, it profoundly influences their social life. Individuals with visual impairments often transition from an independent lifestyle to a pattern of dependency on others for making decisions related to various aspects of life. These include determining directions for navigation, interpreting signs and informational markings in their vicinity that could pose risks, and addressing privacy concerns that they cannot resolve independently [1] [2].

Impaired vision leading to blindness can be categorized into several distinct situations. While some individuals are blind from birth, others experience vision loss due to diseases or other factors. Those who are blind from birth typically perceive nothing but darkness in their visual field. People coping with blindness often use canes as aids for navigation. They navigate by relying on their sense of smell, attentively listening to the sounds in their surroundings, tactile sensations from objects they encounter or touch, keeping track of the number of steps taken, or recollecting familiar landmarks and previously traversed routes [3].

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 122-133

In daily life, individuals who are blind often rely on assistive devices or assistance from others. One such tool that assists them in understanding their surroundings, particularly in unfamiliar environments, is a cane. However, this tool has limitations, such as not being able to discern objects that are beyond its immediate reach [4].

As a result, various navigation aids for individuals with visual impairments have been developed. One of these aids is the Electronic Travel Aid (ETA), designed to detect objects at ground-level heights within a home environment. Additionally, a Wearable Navigation System has been developed to provide location information and identify the quickest routes to desired destinations for individuals with visual impairments [5].

When it comes to counting and conveying information to blind individuals, object counting plays an integral role. This information, including crowd sizes, the number of objects present, and other supplementary details, can significantly enhance the understanding of the environment for blind individuals [6].

The field of object counting represents a challenging aspect of computer vision. Researchers have devoted considerable attention to this subject due to its broad practical applications. These applications encompass diverse domains, ranging from calculating mass and counting living cells to animal population assessments for ecological research, environmental surveys, and determining the quantity of palm oil during the oil production process [7].

When considering alternative means of navigation, aside from object counting, object detection stands as a fundamental pillar in addressing this challenge. It even offers the possibility for individuals not to actively participate in the navigation process [8].

Developing tools to assist individuals with visual impairments, based on prior research, necessitates a synergy of both hardware and software components. The hardware encompasses various elements, including the initial input of information data, the processing of information, and the delivery of information required by individuals with visual impairments. These three hardware capabilities are designed to be portable and user-friendly. Among these, a crucial tool resides in the information processing segment. One such compact, portable tool that can be seamlessly integrated into other devices and accepts input data in the form of images from individuals with visual impairments, subsequently processing this input into valuable information, is the OpenMV system.

OpenMV is a small microcontroller with a function like Arduino [9], [10]. It exhibits lower memory usage, reduced latency, and greater accuracy when compared to Arduino [9]. Additionally, OpenMV is equipped with a camera to capture input data in the form of images, which are subsequently processed using the microPython programming language [10]. OpenMV offers an array of modules designed to support object detection and counting, including the Haar Cascade classifier method. This method relies on Haar features extracted from images, with testing data matched against reference information stored in .xml format [11].

The primary objective of this study is to develop a simulation tool for the detection and counting of human objects utilizing the Haar Cascade Classifier and OpenMV methods. The Haar Cascade algorithm is a commonly employed technique in object detection systems [12] [13], renowned for its rapid object detection capabilities in digital images. For this particular system, human object detection and counting involve classifying the human body into one of three categories: the entire body, upper body, or lower body, with a focus on the upper body due to its prevalence in digital imagery. The computational process within this algorithm centers on quantifying the differences between dark and light pixels.

The complete package tool is not the concern of our study. This focus if this paper concerns one of the pioneering concepts within our research in this field. Consequently, our investigation has centered on identifying the primary hardware and software requirements. We intend to showcase and assess the utility of object detection and counting systems utilizing the Haar Cascade classifier method on OpenMV devices based on the results generated by the system. The image data used in this system is sourced from the camera module of the OpenMV device. We process these images and videos to evaluate the implementation of the object detection and counting system using the Haar Cascade method on OpenMV devices. Furthermore, we compare the test results obtained from OpenMV with those from OpenCV, both utilizing the same Haar Cascade classifier.

http://doi.org/10.26594/register.v9i2.3175

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 122-133

For evaluation purposes, we have assembled a collection of image data from the OpenMV camera module. Our testing regimen comprises four distinct types of tests: assessing the impact of object distance from the background, evaluating the effect of the contrast between objects and the background, analyzing the effect of object condition, and examining the influence of lighting conditions. We perform three types of processing: image processing with OpenCV, image processing with OpenMV, and video processing with OpenMV. These tests are designed to gauge the success rate of object detection and counting using the OpenMV tool.

This paper is structured into four main sections: Part 1 provides an introduction to the research background, Section 2 presents a review of the current state of the field, Section 3 outlines the research methods employed, and Section 4 offers a comprehensive presentation of the results and corresponding discussions. Finally, in Section 5, we draw conclusions based on the findings of this study.

2. Materials and Methods

This section provides an overview of various aspects related to the title. Subsection 2.1 discusses object detection, Subsection 2.2 explores object counting, and Subsection 2.3 explores the Haar Cascade method.

2.1. Object Detection

In recent developments within the field of object detection algorithms, significant emphasis has been placed on the precise identification of faces, skin colors, and surveillance cameras [14]. A scientific article highlights how object detection accuracy can reach up to 95% through MATLAB 2017b simulations, making it feasible for implementation on high-spec computers. This literature underscores the possibility of implementing object detection with a reasonably high level of accuracy, even on systems with moderate specifications, thereby optimizing computational efficiency.

Within the field of object detection, person detection constitutes a crucial aspect, focused on identifying human-like objects within images. The process involves defining the coordinates, tagging, and labeling of these human objects within an image. A study [15] provides insights into the implementation of person detection using YOLO 3, which has been evolved into PDnet. This development revolves around the optimization of the clustering algorithm within YOLO 3, aimed at reducing distortion and enhancing accuracy in the detection of humans.

Additionally, Lodhi et al. [8] employ YOLO3 to detect objects for navigation on Autonomous Ground Vehicles (AGV). The outcomes presented in this paper indicate 3.7 detection errors within a viewing distance ranging from 2.5 to 5.5 meters, based on images captured by the camera.

2.2. Object Counting

Counting the number of objects has become increasingly important in the present context. The utilization of HFNet for object counting, as demonstrated by Zhang et al. [16] has yielded competitive results in tallying crowd and vehicle data. In a separate study, Börold, et al. [17] developed an automatic calculation system tailored for the automotive industry's product objects, leveraging deep learning. Their findings suggest that accurate detection of car components is contingent on the correct packaging of these components.

Kakehi, et al [18] introduced a deep learning-based object detection technique for identifying and counting Pacific oyster larvae, achieving an accuracy rate of 82.4%. Meanwhile, Liu, et al [19] performed object calculations with a focus on addressing variations in detected objects, posing challenges in multi-class object detection and routing. In a related context, Yang [20] explored multi-class objects using a tracking-by-detection strategy, stemming from automatic counting of cotton seedlings, with a Root Mean Square Error (RSME) value reaching 0.394.

Rodriguez-Vazquez, et al. [21] performed object calculations using the CNN and Laplacian of Gaussian object detection methods. The results showed that the position of the detected object could be mapped well to support the object calculation process. Li He at al [22] designed a deep-scale aggregation network dedicated to object counting, yielding commendable performance for the automatic calculation system they developed. Lastly, Zhang Yanchao, et al. [23] performed fruit fruit counting using panorama and deep learning object detection methods. Their study highlights the high accuracy achieved when working with high-quality input images, showcasing the method's superior efficacy in object counting compared to previous approaches.

http://doi.org/10.26594/register.v9i2.3175

2.3. Haar Cascade

A signature serves as a critical means of identifying individuals and plays a pivotal role in document legalization. In this study, the Haar Cascade method is introduced as a tool to differentiate between signatures and Persian writing. The results demonstrate that Haar Cascade can yield satisfactory outcomes, achieving an actual accuracy rate of 92.42% in distinguishing Persian Latin text and correctly identifying signatures composed in various styles [24].

In the literature review conducted within the respective subsections, the Haar Cascade method is applied to detect objects within other objects, specifically focusing on motorcycles within zebra crossings, which is a violation of regulations in Indonesia. The specified criterion considers a motorcycle user's object label as non-compliant when it occupies 50% or more of the zebra cross square object. The findings reveal that detection accuracy can reach 83% when using a 720P resolution camera in crowded conditions. This study underscores the pivotal role of image quality in influencing the accuracy of the Haar Cascade classifier [13].

In the subsections explaining the literature study, one of the prevalent urban challenges is traffic congestion. To address this issue, the author suggests implementing traffic control systems to enhance traffic management efficiency. Haar Cascade emerges as a rapid implementation method, with timing calculations tailored to specific locations. Its primary function lies in object counting to monitor changes in traffic duration.

Dawn et al [25] employ the Haar Cascade Classifier method to index images and subsequently compile similar images based on queries generated from the original image. The research yields valid results for identical images while producing false values for images with similar pixel values.

Recently, Aarthi and Harini [26] designed an innovative and efficient traffic light system utilizing the Haar Cascade Classifier. This system detects object clusters, adjusting traffic light timers accordingly. As a result, traffic congestion is reduced by up to 60%. employ the Haar Cascade Classifier to filter facial images from Instagram searches based on hashtags. In a real-time context, Ulfa et al. [28] perform frame detection on motorcycle users using the Haar Cascade Classifier, achieving frame processing speeds as quick as 40 milliseconds.

2.4. Observing the Object

Data utilized in this research is sourced from the camera module of the OpenMV device, consisting of four distinct types of tests: assessing the impact of object distance from the background, evaluating disparities between objects and the background, considering object conditions, and accounting for lighting conditions. The processing phase is segmented into three categories: image processing with OpenCV, image processing with OpenMV, and video processing with OpenMV. All three scenarios utilize the same classifier method, namely the Haar Cascade. It is worth noting that different tools are employed for image capture in the OpenMV and OpenCV processes. OpenMV captures images at a significantly lower resolution compared to OpenCV scenarios.

Images captured on the OpenMV device have a maximum resolution of 5MP, while the device can only process images at a maximum resolution of 240P. If an image exceeding 240P resolution is supplied to the OpenMV device, it may crash and fail to continue the processing. Conversely, the image input for the OpenCV test scenario utilizes a cellphone camera with a 48MP resolution. In terms of video handling on OpenMV, the simplest algorithm operates at a frame rate ranging from 25 to 50 FPS. The system employs a dataset focused on human upper bodies for human object detection. Upon detecting the upper portion of a human body within an image, the system delineates a square around the object and furnishes information regarding the number of objects detected.

2.5. Proposed Tools

The simulation tool introduced in this paper utilizes a compact tool known as OpenMV and processes input image data using the primary Haar Cascade classifier method. The testing location selected for this study is the State Polytechnic of Malang campus, encompassing both indoor and outdoor lighting conditions.

The way this system works is shown in Figure 1. Notably, the second-to-last stage takes place within OpenMV. The process is described as follows: collecting data using an OpenMV camera, processing the data by converting it to grayscale, performing an integral image calculation, combining weak classifiers into robust classifiers using the AdaBoost cascade classifier.

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 122-133

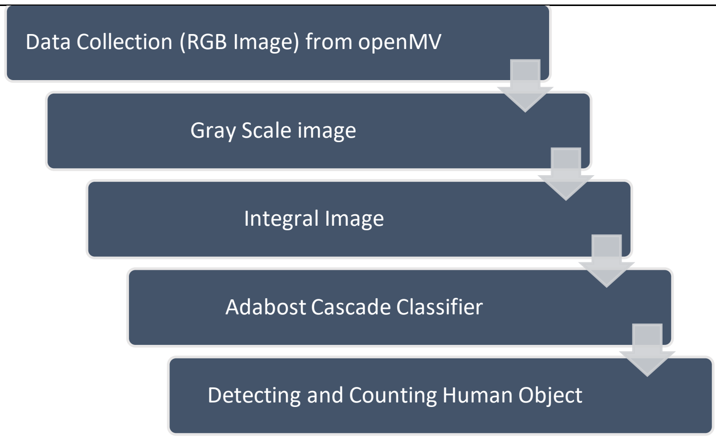

Fig. 1. Method Stages

2.6. Determining Haar Feature in Grayscale Image

The grayscale image is subsequently processed using Haar-like features, employing training data to construct a decision tree within the cascade classifier. The objective is to ascertain the system's capability and identify the presence of human objects within each processed frame.

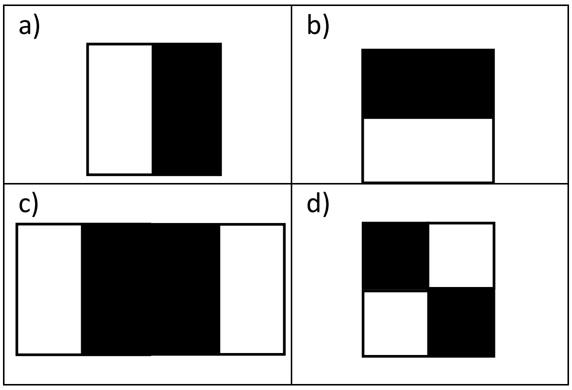

Fig. 2. Haar Feature

The presence of the Haar feature shown in Figure 2 is determined by subtracting the average pixel in the dark area from the average pixel in the bright area with equation (1). It is essential that all images are transformed into an average value normalized from previous variations to ensure consistent brightness levels. This normalization process results in images with reduced variation compared to others, minimizing information loss during assessment.

$$F(Haar) = \sum FWhite - \sum FBlack \quad (1)$$

with $\sum FBlack$ = the number of black pixels, $\sum FWhite$ = the number of white pixels, and

$F(Haar) = Haar - like\ features$

When the Haar-like feature surpasses a specific threshold, it indicates the detection of an object within the examined area. Consequently, program code blocks that efficiently compute Haar-like features enable code to utilize substantial memory and processing resources [29].

2.7. Integral Image Process

The essential image processing step involves simplifying feature calculations using Haar-like features to expedite the detection process. Calculating features without employing an integral image results in a higher computational load compared to when using an integral image. An integral image is where each pixel is the sum of the pixels from the top left to the bottom right.

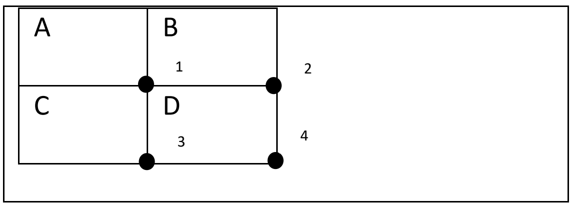

Fig. 3. Integral Image

http://doi.org/10.26594/register.v9i2.3175

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 122-133

The utilization of integral images, as depicted in Figure 3, simplifies the process of calculating the number of pixels within a specific area of the image. Subsequently, a machine learning technique known as AdaBoost is employed to choose the particular Haar feature for use and establish the threshold value.

2.8. Integral Image Process

The Adaboost cascade classifier process, as illustrated in Figure 4, aims to accelerate the object classification procedure. Adaboost, short for "adaptive boost," is a learning algorithm employed for object detection. This algorithm operates by combining numerous weak classifications into a more robust one. Within Adaboost, a specific feature with a dynamically adjusted threshold value is automatically determined for human object detection. This threshold is subsequently utilized to establish the minimum value required for a feature to pass through a filter.

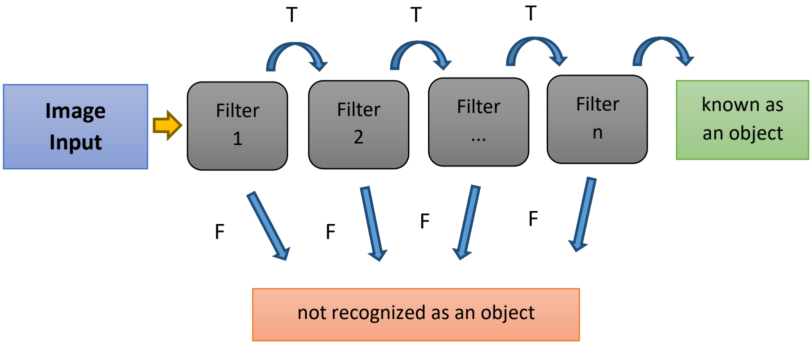

Fig. 4. Cascade Classifier Diagram

The filter contains criteria for determining whether the input image contains human elements. The highest weight is assigned to the first filter, primarily designed to swiftly identify whether a feature contains an object. If a feature fails to be detected by the first filter, it is categorized as "not an object." Conversely, when a feature successfully passes through all filters, the corresponding area within the image is classified as a human object.

2.9. The OpenMV Camera

The OpenMV camera shown in Figure 5 is a compact, energy-efficient, and cost-effective camera. This circuit board is designed to facilitate rapid machine vision capabilities. Notably, programming can be carried out using Python, specifically MicroPython, rather than relying on C/C++. It is worth mentioning that Python can effectively handle various intricate outputs in machine vision algorithms. Furthermore, an additional advantage lies in the camera's capacity to generate code for conveniently controlling I/O pins. MicroPython simplifies the complexity of OpenMV, enabling it to trigger image capture and execute algorithms concurrently while processing I/O pins. The simulation tool featured in this study is presented in Figure 6. It encompasses results that include object captures and their corresponding environmental conditions, bounding boxes indicating the detection of human objects, and a numerical display of the total count of detected human objects.


Fig. 5. OpenMV Camera

Fig. 6. Display of Detection and Counting of Human Objects

3. Results and Discussion

As previously outlined, the tests conducted in this paper are categorized into several sections. These tests encompass an examination of the impact of object-to-background distance, disparities between the object and the background, object conditions, and the influence of lighting conditions. The testing environment for these experiments was at the State Polytechnic of Malang. It is noteworthy that the

http://doi.org/10.26594/register.v9i2.3175

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 122-133

image resolution used was 240P for OpenMV, 48MP resolution for OpenCV input, and a video frame rate ranging between 25 and 50 FPS.

3.1. All Human Counting Results

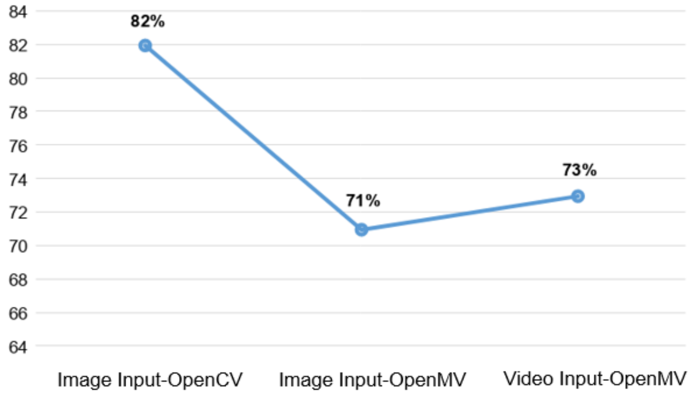

Fig. 7. Counting Results

Figure 7 presents the comprehensive results of human counting. The system achieved its highest object detection and counting accuracy during testing when using OpenCV for image processing, reaching an 82% accuracy rate. It is important to note that this test employed a system designed for detecting and counting human upper bodies, as these are commonly present in images featuring human objects. While OpenCV yielded impressive results, it is essential to recognize that it may not be accessible on smaller hardware platforms like OpenMV. Despite its slightly lower percentage, OpenMV achieved a commendable object detection accuracy rate of 71%. To attain even better performance than OpenMV, video input, as opposed to images, was utilized, resulting in an accuracy rate of 73%. OpenMV, which employs the MicroPython programming language, offers distinct advantages by being compatible with microcontrollers, as indicated by Plauska et al.'s research [30]. These advantages include lower complexity, greater developer-friendliness, suitability for beginners, and faster coding capabilities. This flexibility sets it apart from OpenCV, which utilizes the Python language, although both OpenCV and OpenMV boast numerous libraries [31]. In specific hardware-related scenarios, they serve different purposes.

The bar chart below employs a rating scale of 10 for total objects, considering that the number of objects tested in each scenario exhibits varying differences. Figure 8 reveals that the highest detection and accurate counting percentage for object detection and counting was achieved in image testing conducted with OpenCV processing. Scoring 9 out of 10 in detection accuracy is noteworthy, given that OpenCV offers greater flexibility in image resolution compared to the lower resolution images captured by the OpenMV camera.

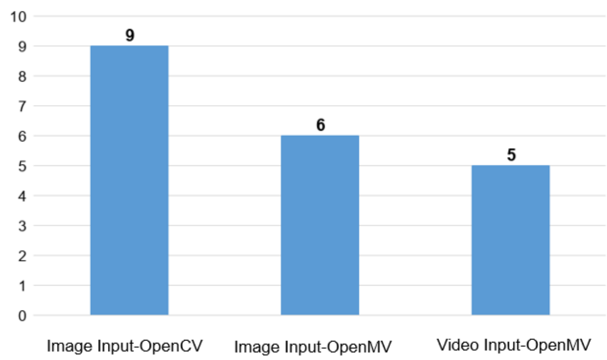

Fig. 8. Object Detection Results - Testing the Effect of Distance on the Image

 http://doi.org/10.26594/register.v9i2.3175

M. Mentari et al.
regist. j. ilm. teknol. sist. inf.
ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 122-133

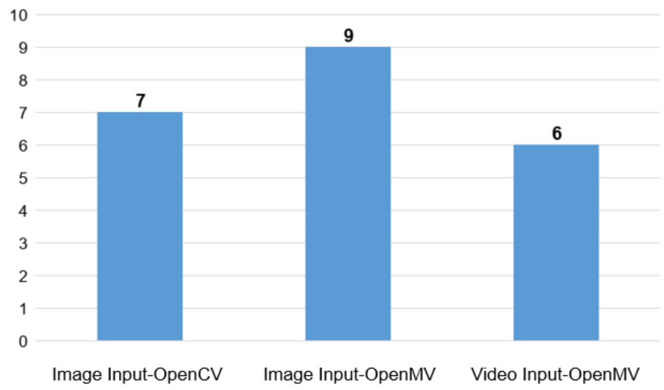

Fig. 9. Object Counting Results - Testing the Effect of Object Color Differences on the Image

Analyzing the impact of color differences, as depicted in Figure 9, reveals that the most precise object counting and detection results are obtained in images processed with OpenMV, particularly when evaluating higher color differences. The test outcomes indicate that the system tends to struggle in detecting and counting objects with low color contrast against the background color, especially in the case of OpenCV, which can only detect approximately 7 out of 10 objects in such scenarios. In contrast, OpenMV demonstrates the ability to detect objects even when they exhibit minimal color contrast.

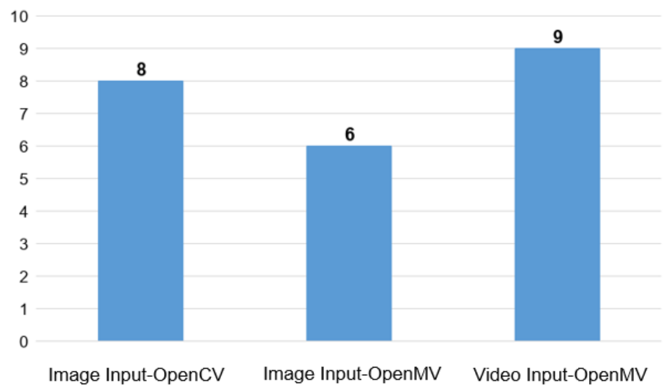

Fig. 10. Object Counting Results - Testing the Effect of Object Conditions

After conducting tests to assess the impact of color differences, the most precise counting and detection results were observed in scenarios examining object conditions, as depicted in Figure 10, and object conditions with lighting effects, shown in Figure 11. The detection results using video in OpenMV demonstrated high accuracy similar to that of OpenCV. These tests revealed that the system is capable of detecting and counting objects under normal conditions but faces challenges when certain objects obstruct the view of human objects, causing them to extend beyond the chest area or when human objects are oriented sideways. In such situations, the system may fail to recognize these objects as human objects.

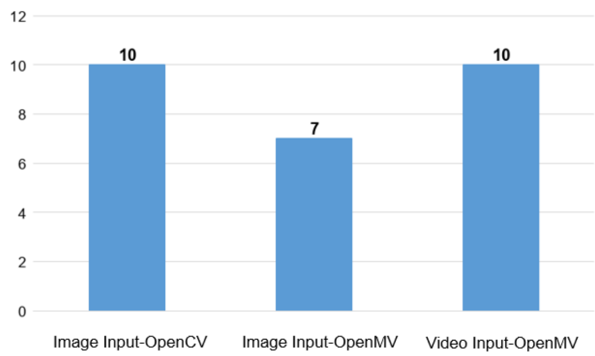

Fig. 11. Object Counting Results - Testing the Effect of Light

The majority of successful detection and accurate counting results were obtained in image testing with OpenCV and video images with OpenMV. This testing indicated that the system generally succeeded in detecting and counting objects across various locations, including both indoor and outdoor

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 122-133

settings when objects were well-illuminated. However, detection failures with OpenMV primarily occurred in bright outdoor lighting conditions. The high contrast in outdoor settings led to elevated grayscale values in OpenMV, diminishing the quality of object detection in these conditions.

3.2. Human counting results using image data on OpenMV

The evaluation of the human object detection and counting program involves direct processing with the OpenMV device under various environmental conditions and object positions. This testing is conducted to gauge the success rate of human object detection under different image conditions when processed directly with the OpenMV tool.

Notably, the maximum distance at which objects can be detected using OpenMV processing is identical to that achieved with OpenCV, which is 4 meters, as depicted in Figure 12. It is important to observe that the distance between the human object and the background does not significantly affect the detection results. However, the detection results are influenced by the spacing between human objects, as illustrated in Figure 13. In scenarios where there should be two distinct human objects, the system tends to detect them as a single human object when using OpenMV processing.

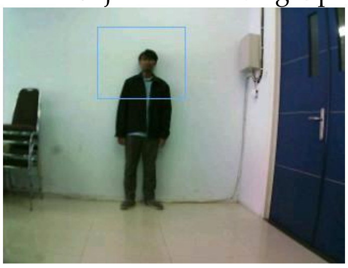

Fig. 12. Distance Testing of One Object

Fig. 13. Distance Testing of Two Objects

Objects with colors nearly identical to the background color may not be detected by the system due to the inherent limitations of the Haar Cascade algorithm. This algorithm can only detect objects with distinctive color differences compared to their surroundings when using OpenMV for processing, as illustrated in Figure 14. Interestingly, it was observed that error detection results were more prevalent when using OpenMV processing compared to OpenCV.

Figure 14. Contrast Color Testing of Objects

The system's capability lies in detecting and counting human objects by focusing on the upper body. Consequently, it can still detect and count human objects even when the lower body is partially obscured by other objects. Notably, there are disparities in the test results between processing with OpenMV and OpenCV. In OpenMV processing, the system may fail to detect and count human objects when the upper abdomen is obstructed by a barrier. Conversely, when using OpenCV processing, these objects can be successfully detected and counted by the system. Furthermore, an additional difference is evident in Figure 15. When processed with OpenMV, the system may detect and count objects facing away from the camera as two human objects, whereas OpenCV processing typically recognizes and counts such objects as a single human object.

http://doi.org/10.26594/register.v9i2.3175

Fig. 15. The human object is facing away from the camera.

A system utilizing OpenMV processing for image analysis demonstrated the capability to detect human objects under three different lighting conditions. However, under bright outdoor lighting conditions, the system encountered difficulties in detecting and enumerating human objects. Out of the four object detection and counting tests conducted, three yielded accurate results.

4. Conclusion

This paper has successfully developed a simulation tool designed to assist individuals with visual impairments in detecting and counting the number of human objects using OpenMV and the Haar Cascade Classifier. Notably, OpenMV exhibits a more lightweight profile compared to other microcontrollers. The evaluation results reveal that utilizing the Haar Cascade algorithm on OpenMV devices for object detection and counting yields a lower detection rate when compared to using OpenCV on PC devices. Specifically, the detection rates are 82% for OpenCV and 71% for OpenMV. This disparity in accuracy can be attributed to the lower resolution of the OpenMV camera compared to the high-resolution camera used with OpenCV. Further research stemming from this study suggests the need for a tool supporting higher resolution than OpenMV to enhance data accuracy. Nevertheless, the employment of the Haar Cascade method as a classifier in OpenMV demonstrates a commendable level of accuracy, which is not too distant from the performance of OpenCV, despite the substantial difference in image resolution.

Acknowledgment

We would like to extend our gratitude to all members of the Computer Vision Lab and the Information Technology Department at the State Polytechnic of Malang for their invaluable support in providing tools for this research. We also extend our appreciation to Saga University and other parties who have contributed to the completion of this research, even though we cannot mention each one individually.

Author Contributions

M. Mentari: Conceptualization, data curation, methodology, resources, validation, visualization, writing – original draft, and writing - review & editing. R. A. Asmara: Conceptualization, data curation, methodology, resources, and writing – review & editing. K. Arai: Investigation, methodology, resources, supervision, writing – original draft, and writing - review & editing. H. S. Oktafiansyah: Formal analysis, funding acquisition, methodology, project administration, software, validation, writing – original draft.

References

[1] V. Kunta, C. Tuniki, and U. Sairam, "Multi-Functional Blind Stick for Visually Impaired People," in 2020 5th International Conference on Communication and Electronics Systems (ICCES), IEEE, Jun. 2020, pp. 895–899. doi: 10.1109/ICCES48766.2020.9137870.

[2] S. Zaib, S. Khusro, S. Ali, and F. Alam, "Smartphone Based Indoor Navigation for Blind Persons using User Profile and Simplified Building Information Model," in 2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), IEEE, Jul. 2019, pp. 1–6. doi: 10.1109/ICECCE47252.2019.8940799.

[3] S. Kumar KN, R. Sathish, S. Vinayak, and T. Parasad Pandit, "Braille Assistance System for Visually Impaired, Blind & Deaf-Mute people in Indoor & Outdoor Application," in 2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), IEEE, May 2019, pp. 1505–1509. doi: 10.1109/RTEICT46194.2019.9016765.

[4] J. Zhu et al., "An Edge Computing Platform of Guide-dog Robot for Visually Impaired," in 2019 IEEE 14th International Symposium on Autonomous Decentralized System (ISADS), IEEE, Apr.

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 122-133

2019, pp. 1–7. doi: 10.1109/ISADS45777.2019.9155620.

[5] J. Bai, S. Lian, Z. Liu, K. Wang, and D. Liu, "Virtual-Blind-Road Following-Based Wearable Navigation Device for Blind People," IEEE Transactions on Consumer Electronics, vol. 64, no. 1, pp. 136–143, Feb. 2018, doi: 10.1109/TCE.2018.2812498.

[6] T. Stahl, S. L. Pintea, and J. C. van Gemert, "Divide and Count: Generic Object Counting by Image Divisions," IEEE Transactions on Image Processing, vol. 28, no. 2, pp. 1035–1044, Feb. 2019, doi: 10.1109/TIP.2018.2875353.

[7] G. Gao, Q. Liu, and Y. Wang, "Counting From Sky: A Large-Scale Data Set for Remote Sensing Object Counting and a Benchmark Method," IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 5, pp. 3642–3655, May 2021, doi: 10.1109/TGRS.2020.3020555.

[8] H. Jha, V. Lodhi, and D. Chakravarty, "Object Detection and Identification Using Vision and Radar Data Fusion System for Ground-Based Navigation," in 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, Mar. 2019, pp. 590–593. doi: 10.1109/SPIN.2019.8711717.

[9] W. Raza, A. Osman, F. Ferrini, and F. De Natale, "Energy-Efficient Inference on the Edge Exploiting TinyML Capabilities for UAVs," Drones, vol. 5, no. 4, p. 127, Oct. 2021, doi: 10.3390/drones5040127.

[10] I. Abdelkader, Y. El-Sonbaty, and M. El-Habrouk, "OPENMV: A Python Powered, Extensible Machine Vision Camera," 2017.

[11] P. Sridhar, P. Chithaluru, S. Kumar, O. Cheikhrouhou, and H. Hamam, "An Enhanced Haar Cascade Face Detection Schema for Gender Recognition," in 2023 International Conference on Smart Computing and Application (ICSCA), IEEE, Feb. 2023, pp. 1–5. doi: 10.1109/ICSCA57840.2023.10087742.

[12] R. Y. Adhitya, A. Khumaidi, S. T. Sarena, S. Kautsar, B. Widiawan, and F. L. Afriansyah, "Applied Haar Cascade and Convolution Neural Network for Detecting Defects in The PCB Pathway," in 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), IEEE, Nov. 2020, pp. 408–411. doi: 10.1109/CENIM51130.2020.9297996.

[13] M. P. Anggadhita and Y. Widiastiwi, "Breaches Detection in Zebra Cross Traffic Light Using Haar Cascade Classifier," in 2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS), IEEE, Nov. 2020, pp. 272–277. doi: 10.1109/ICIMCIS51567.2020.9354275.

[14] A. Raghunandan, Mohana, P. Raghav, and H. V. R. Aradhya, "Object Detection Algorithms for Video Surveillance Applications," in 2018 International Conference on Communication and Signal Processing (ICCSP), IEEE, Apr. 2018, pp. 0563–0568. doi: 10.1109/ICCSP.2018.8524461.

[15] L. Zhao and Y. Wan, "A New Deep Learning Architecture for Person Detection," in 2019 IEEE 5th International Conference on Computer and Communications (ICCC), IEEE, Dec. 2019, pp. 2118–2122. doi: 10.1109/ICCC47050.2019.9064172.

[16] S. Zhang, H. Li, W. Kong, L. Wang, and X. Niu, "An object counting network based on hierarchical context and feature fusion," J Vis Commun Image Represent, vol. 62, pp. 166–173, Jul. 2019, doi: 10.1016/j.jvcir.2019.05.003.

[17] A. Börold, M. Teucke, A. Rust, and M. Freitag, "Deep Learning-based Object Recognition for Counting Car Components to Support Handling and Packing Processes in Automotive Supply Chains," IFAC-PapersOnLine, vol. 53, no. 2, pp. 10645–10650, 2020, doi: 10.1016/j.ifacol.2020.12.2828.

[18] S. Kakehi et al., "Identification and counting of Pacific oyster Crassostrea gigas larvae by object detection using deep learning," Aquac Eng, vol. 95, p. 102197, Nov. 2021, doi: 10.1016/j.aquaeng.2021.102197.

[19] Z. Liu, Q. Wang, and F. Meng, "A benchmark for multi-class object counting and size estimation using deep convolutional neural networks," Eng Appl Artif Intell, vol. 116, p. 105449, Nov. 2022, doi: 10.1016/j.engappai.2022.105449.

[20] H. Yang et al., "Multi-object tracking using Deep SORT and modified CenterNet in cotton seedling counting," Comput Electron Agric, vol. 202, p. 107339, Nov. 2022, doi: 10.1016/j.compag.2022.107339.

 http://doi.org/10.26594/register.v9i2.3175

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 122-133

[21] J. Rodriguez-Vazquez, A. Alvarez-Fernandez, M. Molina, and P. Campoy, "Zenithal isotropic object counting by localization using adversarial training," Neural Networks, vol. 145, pp. 155–163, Jan. 2022, doi: 10.1016/j.neunet.2021.10.010.

[22] H. Li, W. Kong, and S. Zhang, "Deeply scale aggregation network for object counting," Knowl Based Syst, vol. 210, p. 106485, Dec. 2020, doi: 10.1016/j.knosys.2020.106485.

[23] Y. Zhang, W. Zhang, J. Yu, L. He, J. Chen, and Y. He, "Complete and accurate holly fruits counting using YOLOX object detection," Comput Electron Agric, vol. 198, p. 107062, Jul. 2022, doi: 10.1016/j.compag.2022.107062.

[24] A. AbdelRaouf and D. Salama, "Handwritten Signature Verification using Haar Cascade Classifier Approach," in 2018 13th International Conference on Computer Engineering and Systems (ICCES), IEEE, Dec. 2018, pp. 319–326. doi: 10.1109/ICCES.2018.8639437.

[25] S. Dawn, S. Tulsyan, S. Bhattarai, S. Gopal, and V. Saxena, "An Efficient Approach to Image Indexing and Retrieval Using Haar Cascade and Perceptual Similarity Index," in 2020 6th International Conference on Signal Processing and Communication (ICSC), IEEE, Mar. 2020, pp. 108–113. doi: 10.1109/ICSC48311.2020.9182741.

[26] V. Aarthi and B. Harini, "Prototype Design of Intelligent Traffic Signal Control using Haar Cascade Classifier," in 2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 260-264, 2021. DOI: 10.1109/WiSPNET51692.2021.9419366.

[27] A. Priadana and M. Habibi, "Face Detection using Haar Cascades to Filter Selfie Face Image on Instagram," in 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), IEEE, Mar. 2019, pp. 6–9. doi: 10.1109/ICAIIT.2019.8834526.

[28] D. K. Ulfa and D. H. Widyantoro, "Implementation of haar cascade classifier for motorcycle detection," in 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), IEEE, Nov. 2017, pp. 39–44. doi: 10.1109/CYBERNETICSCOM.2017.8311712.

[29] T. Mantoro, M. A. Ayu, and Suhendi, "Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods," in 2018 6th International Conference on Multimedia Computing and Systems (ICMCS), IEEE, May 2018, pp. 1–5. doi: 10.1109/ICMCS.2018.8525935.

[30] I. Plauska, A. Liutkevičius, and A. Janavičiūtė, "Performance Evaluation of C/C++, MicroPython, Rust and TinyGo Programming Languages on ESP32 Microcontroller," Electronics (Basel), vol. 12, no. 1, p. 143, Dec. 2022, doi: 10.3390/electronics12010143.

[31] M. El-Habrouk, I. Abdelkader, and Y. El-Sonbaty, "Openmv: A Python powered, extensible machine vision camera," 2017. [Online]. Available: https://www.researchgate.net/publication/321347754