Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 9, No. 4, December 2023, pp. 895-903
ISSN: 2338-3070, DOI: 10.26555/jiteki.v9i4.27009

Fast Human Recognition System on Real-Time Camera

Yuliza[1], Rachmat Muwardi[1], Mustain Rhozaly[1], Lenni[2], Mirna Yunita[3], Galatia Erica Yehezkiel[3]
[1]Department of Electrical Engineering, Universitas Mercu Buana, Jakarta, Indonesia
[2]Department of Electrical Engineering, Universitas Muhammadiyah Tangerang, Tangerang, Indonesia
[3]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China

ARTICLE INFO

Article history:

Received August 23, 2023
Revised September 22, 2023
Published September 30, 2023

Keywords:

YOLO;
Optimization;
Image Processing;
Computer Vision;
Object Detection

ABSTRACT

The rapid development of technology requires several technological fields to keep pace to increase work effectiveness and efficiency. One focus that is widely used is image processing technology. Many areas have implemented image processing systems due to various benefits, including security, health, and education. One of the current obstacles is safety, focusing on searching for missing people, which is still done manually. The vast search area, low light conditions, and complicated search fields made finding the person challenging for the search team. Therefore, it is necessary to have a tool that can detect people to assist in the search process. This paper proposes an object detector on a simple device capable of detecting human objects. The detection device was made using the You Only Look One (YOLO) method with the YoloV4-Tiny type, where this algorithm has high detection speed and accuracy. Using the YOLOV4-Tiny simulation method for human object recognition obtained satisfactory results with a detection rate of 100% with an FPS value of 5.

Corresponding Author:

Yuliza, Department of Electrical Engineering, Universitas Mercu Buana, Jakarta, Indonesia
Email: yuliza@mercubuana.ac.id

1. INTRODUCTION

Currently, the rapid development of technology is marked by the application of various modern technologies in everyday life to improve quality in multiple fields. Technological developments enhance the quality of security systems we often encounter daily, including object detection technology. One of the promising applications of object detection technology is unlocking cell phones using face detection. Apart from that, currently, a robot has developed those functions as a human assistant, with one of its primary functions being to follow humans, use object detection, and carry out tasks according to the instructions given [1]-[4]. One area considered necessary for modernization is the search for missing people in the forest. This is due to the difficulty of the terrain to traverse and the limited number of search teams, which makes the search process often experience problems. One possible solution is to apply object detection technology to help search missions for missing people in the forest using drones [5]-[7]. To carry out the object detection process, a modern technological device is needed to support the system's functions. The most vital component of this modern technology is the microprocessor. A high-speed and high-technology microprocessor is required to carry out the functions of heavy programming algorithms [8]-[10]. The part of the microprocessor itself is the primary control. The microprocessor will process the input data according to the executed program and then produce output as instructions. Several programming languages, such as C, C ++, Assembly, Verilog, VHDL, Python, etc., are used to do programming in a microprocessor [11]-[13]. These various programming languages have advantages and disadvantages and are used according to the programmer's needs [14]-[16].

The algorithm often used in object detection is YOLO (You Only Look Once), developed to detect objects in real-time. The system works by using a single neural network for the entire image. This network will divide the image into regions and predict the bounding boxes and the possibility of being detected as an object. YOLO can recognize objects based on the classification data that has been entered and can see the location of things.

This method can be called one of artificial intelligence because it can interpret external conditions based on the data entered as the YOLO training algorithm [17]-[20].

This research aims to count the number of people in one room. Using the YoloV3 and SSD model detection methods, an average precision of 99% was obtained for the YoloV3 model and 87% for the SSD model [21]-[23]. However, using YoloV3 requires a longer detection time, 0.2 seconds or only around 5 FPS. Another problem encountered was the need for a clear explanation regarding the hardware devices used. Then, comparisons using other hardware were also not carried out, which resulted in the unknown cause of the decrease in FPS during the image identification process [24]-[27].

Based on this, the authors will install an object detection system on the drone to make finding people in the forest easier. Then, to make an effective small object detection device, the YOLOV4-Tiny method is used. The latest development algorithm from YOLOV4 Version 5 has a higher Precision function and is 2x faster than version 4. A webcam with 1080p resolution supports high-quality video, and a laptop equipped with a GPU is used for the processor.

2. METHODS

After considering all the background discussed, we can solve the problems using multi-thread. Then, there will be some issues and questions regarding this multithread, such as what image will be processed, what is the purpose of image processing, how to process images, how to make a detailed and correct image processing framework, how to make the core work regularly and directed, and utilize core tools with high efficiency when using the image processing framework real-time. The following section will address and explain all the problems mentioned above. In this work, the image processing optimization methodology will be described. Every Biometric system has four main features: Human Detection, Preprocessing, Feature Extraction, and Human Recognition, which are shown in Fig. 1. Capturing an image is the first task of a human recognition system. Images are captured by video, camera, or from a database. This image is provided for the next step of the human recognition system. The computer is supported by the software, the photo by the device sampled, and the output results or data of the recognized human image. Finally, the output data can be displayed on the screen.

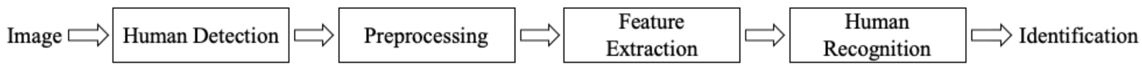

Fig. 1. The architecture of human recognition system.

2.1. YOLOV4 TINY

Yolo, which stands for You Only Look Once, performs an image detection process with a different algorithmic approach than other methods. Yolo cannot only classify objects but can also determine the exact position of objects in real time. Joseph Redmo first developed Yolo and has developed several versions, such as VIZ, YoloV2, YoloV3, YoloV4, and YoloV5. There is a tiny, compressed Yolo version with a more straightforward and lighter algorithm to run so it doesn't burden the PC [28]-[31]. Based on various studies, the YoloV4 Tiny is the most effective and lightweight version of Yolo. It reflected on the development and use of the previous version of Yolo, which has been successfully implemented as an object detection algorithm with a high confidence level for object detection. YoloV4 Tiny was to be introduced to the public and released on April 24, 2020, by Alexey Bobrovsky, Chien yao wang, Hong yuan mark.

YoloV4 Tiny has a different architecture, as shown in Fig. 2 which shows the architecture of YoloV4. Tiny consists of a backbone resulting from the training from an artificial neural network on the image net and a head (neck), which is used to calculate classification and determine boxes on objects [32]-[34].

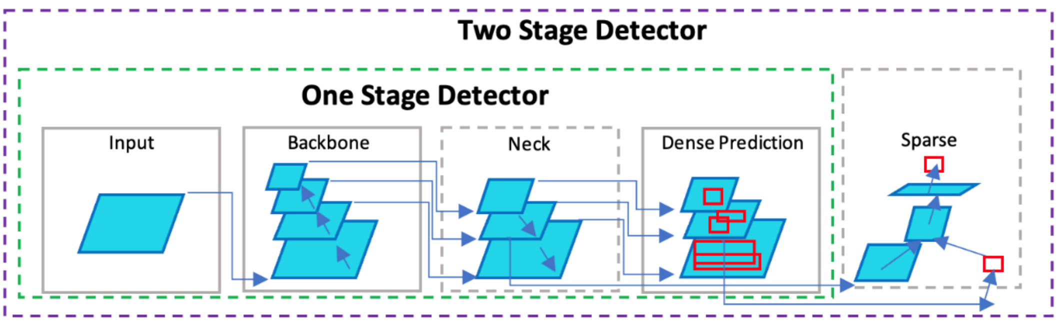

Fig. 2. YOLOV4 Tiny Architecture

To be able to run on a Graphic Processing unit (GPU), the backbones that can be used include VGG, ResNet, and ResNext, while Squize net, mobile net, or shuffle net can run on a Central processing unit (CPU). For the head, which is a sub of the backbone, there are two parts: 1) Stage detector/Dense prediction (Yolo, SSD, Retina Net) and 2) two-stage detector/Spars prediction (Fast R-CNN, Faster R-CNN).

2.2. Real-Time Computing

Real-time computing (RTC), or reactive computing, is the computer science term for hardware and software systems subject to a "real-time constraint" from event to system response. Real-time programs must guarantee a response within specified time constraints, often called "deadlines". Real-time responses are usually understood to be in the order of milliseconds and sometimes microseconds. A system not established as operating in real time can only sometimes guarantee a response within any timeframe, although typical or expected response times may be given. Real-time processing will only succeed if completed within a specified deadline relative to an event, and deadlines must always be met, regardless of system load. A real-time system is one that "controls an environment by receiving data, processing them, and returning the results sufficiently quickly to affect the environment at that time". The term "real-time" is also used in simulation to mean that the simulation's clock runs at the same speed as an accurate clock and in process control and enterprise systems to tell "without significant delay"[35]-[37].

2.3. Open CV

Open CV is one of the libraries commonly used in programming languages like Python, C, C++, Java, Matlab, etc. This library was launched by Intel in 1999, and until now, there have been various versions for its use. The Open CV library is suitable for use in different OS such as Windows, Mac, and Linux, so multiple groups widely use it. Open CV is closely related to image processing, which generally has functions in Fig. 3. Image and video reading in an application so that the Open CV library can easily input videos and images to be processed in an application being developed. Digital image processing, besides being used as input and output of an image, Open CV can also be used to process the picture, such as pixel processing, color, display, zoom in, zoom out, blur, etc. Combined with the Artificial Intelligence program, with the advantage of processing images, it has a high intelligence digital image processing function such as recognizing objects, identifying similar things, counting the number of objects, as in out, and displaying objects. Of the various advantages of the Open CV library, it is suitable for object processing [38]-[41].

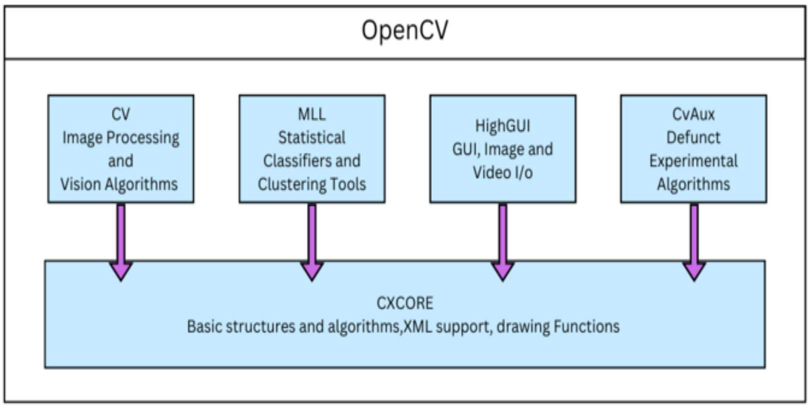

Fig. 3. The structure of OpenCV

2.4. Open CV

A hardware design is made to run the system, while the hardware design is shown in Fig. 4. The input consists of the Logitech C922 Pro webcam as a video capture device with a predetermined resolution of 1080P and a frame rate of 30 Fps. Then, the process and output sections consist of a device in the form of a laptop composed of various components in the processor section, such as an I/O USB port that functions as a connector between the input section and the processor section of the system, GPU acts to optimize the graphics processing process on the computer system, SSD functions as a memory that stores data to be processed on the processor section, RAM is a memory that stores programs that are being executed at that time. The CPU functions to process all incoming algorithms so that a decision is obtained in the form of output data, which is then forwarded to the display monitor section. It is necessary to make a block diagram that clearly describes the input part of the device, the processor part, and the output part—the block diagram, as shown in Fig. 5.

Based on the Fig. 5, it can be seen that the tool consists of 3 parts, namely: 1) Input: In this section, there is an input device in the form of a camera which functions to take pictures in real time, besides that there is also a video that is entered into the process section to be analyzed, and finally in the form of a datasheet, in the datasheet section contains the results of calculations from object detection which are analyzed and used as the Real value of the video which will be compared with the reading results from the process section. 2) Process: This section uses a laptop with software to process Python, OpenCV, and YoloV4-Tiny programs. The program analyzes input in video or real-time from the camera. 3) Output: In this section, a monitor displays the object detection results with an algorithm embedded in the laptop to process object detection. The reading results are displayed as an image on the monitor and data on a data sheet. The output datasheet contains the values obtained from the processing results in the form of Recall, Precision, Accuracy, the number of detected objects, and the FPS of the processed video.

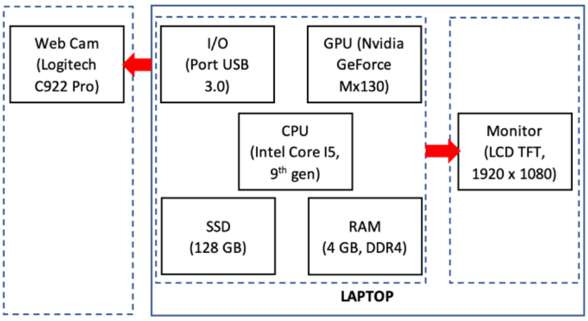

Fig. 4. Hardware design architecture

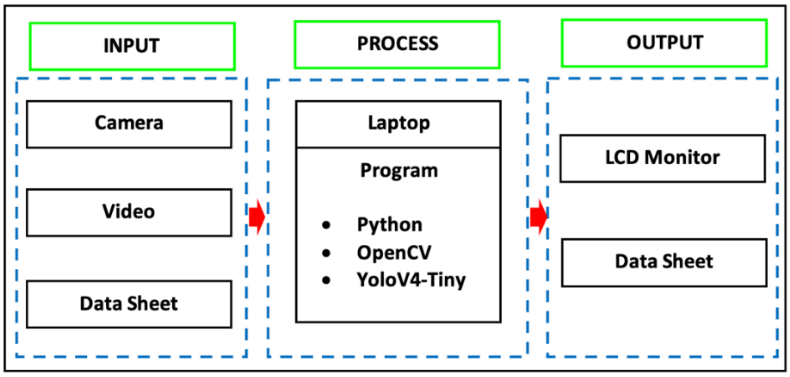

Fig. 5. Diagram Block

2.5. Algorithm Programming

The YOLO algorithm applies the function to train images through 4 main stages: (1) determining YOLO features, (2) making integral images, (3) Ad boost training, and (4) linking with human classifiers. This article will share a short tutorial using the YOLO algorithm to discuss which areas contain human recognition in images using OpenCV.

The neural networks give a nonlinear solution to the human recognition problem. The basic idea is to consider a net with a neuron for every pixel in the image. The advantage of neural networks in classification over linear ones is that they can reduce misclassifications among the neighborhood classes. The Neural Network contains neurons and can carry the data from the input to the output. The data is processed and calculated as the weighted sum of the information applied as a non-linear function.

(1) Initial setting handler;
(2) Processor core and hardware identification;
(3) Image retrieval by camera;
(4) Core utilization analysis;
(5) Workload division mapping and assigner for each core;
(6) Execute image processing task;
(7) Retrieve and arrange the raw result and data of the task;
(8) Further result processing;
(9) Result analysis.

3. RESULTS AND DISCUSSION

This section will compare three existing algorithms: YOLOV4, YOLOV4 TINY, and YOLOV5. The experiment will be tested using the Nvidia MX130 GPU. The parameters used for each algorithm use an image scale of 640x640 with a total of 480 frames using a video of the same duration and the same image.

Testing this level of accuracy is used to determine the level of accuracy of the tool that has been made so that it can be said that the device can be used properly with a perfect level of accuracy. From the existing calculations, the results will be obtained: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). If you get it, you can consider the existing Recall (R) and Precision (P) to determine an algorithm's accuracy level. The definition is as follows.

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Mean Average Precision (mAP) is used to evaluate the model. The purpose of mAP is as follows:

$$mAP = \frac{1}{N}\sum_{t=1}^{N} APi \tag{3}$$

In Fig. 6 and Fig. 7 discusses the video run for 350 Frame. It can detect a human image object using a video. This can be done using a real-time camera without reducing the accuracy level.

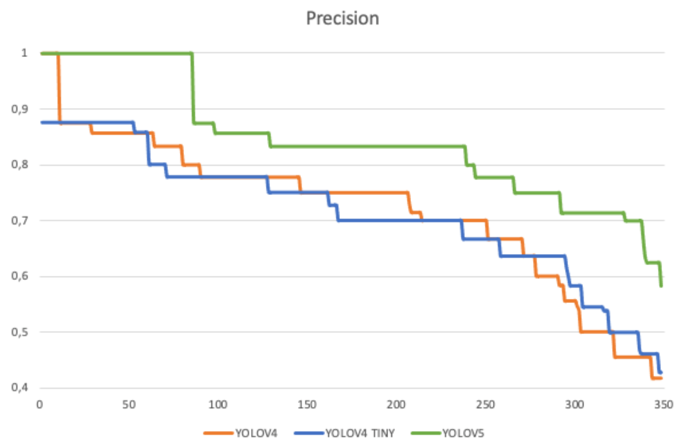

Fig. 6. Precision of several algorithms

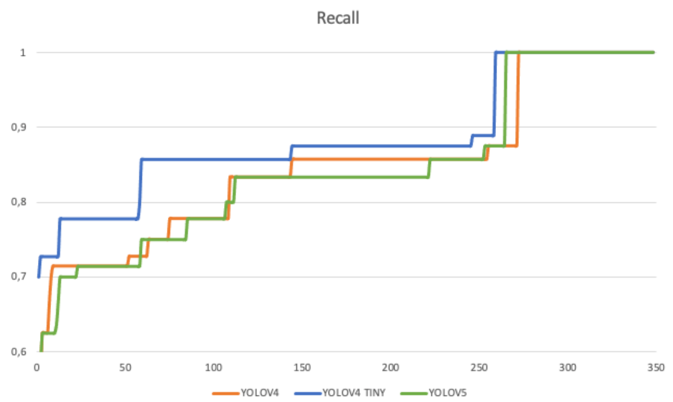

Fig. 7. Recall several algorithms

Fig. 8 discusses a video run for 350 Frame. Human image objects can be detected using video. This can be done using a real-time camera without reducing accuracy and 5 FPS speed. The results show that the YOLOV4 Tiny algorithm has obtained entirely satisfactory results because it has improvised large and small cores so that processing is more focused and improves processor performance much better to compete with YOLOV5.

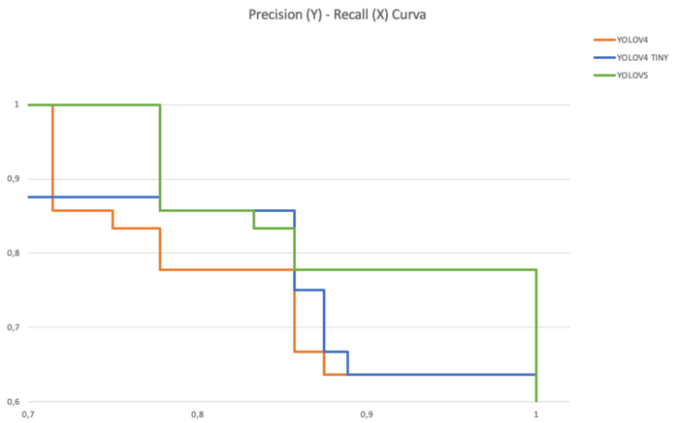

Fig. 8. PR Curve of several algorithms

Evaluation of the human object model that was run with several algorithms is precise that YOLOV4-Tiny is still relatively superior to the others because there is an improvised processor, so the image process is much better and superior to YOLOV5.

When processing images in Fig. 9 the program will then process the video and display the processing results with video values, accuracy levels, and detected object names. Table 1 describes the map of the algorithm used and the average time needed to read images with a total of 350 frames.

(A) (B)

Fig. 9. (A) Human Detection 100% correct (B) Human detection error

Table 1. Evaluation Result

Model	YOLOV4	YOLOV4 TINY	YOLOV5
mAP	0.9519	0.9721	0.9805
Average Time	60 ms/frame	58 ms/frame	55 ms/frame

4. CONCLUSION

Based on the discussion of the results and analysis of this research, it can be concluded that a small object detection system is done by making hardware wiring, then making programs and libraries, and finally by running the program on the hardware. The number of objects detected using the YOLOV4-TINY method compared to YOLOV4 and YOLOV5 shows that YOLOV4-TINY is quite good. This indicates that the YoloV4-Tiny method is the most effective for detecting small objects. The effectiveness of small object detection systems has the same capabilities, which can be seen from the number of detected objects, accuracy, recall, and precision. In the detection system using the YOLOV4-TINY, YOLOV4, and YOLOV5 methods, the average MAP is obtained on the YOLOV4-TINY method so that the process is feasible to be implemented in a small object detection system.

Acknowledgments

The first special thanks go to Universitas Mercu Buana for supporting foreign collaborative research, the second to the Beijing Institute of Technology, Mirna Yunita, and Galatia Erica Yehezkiel for her help and cooperation during this research. Third, to the Universitas Muhammadiyah Tangerang, Lenni, who has provided continuous support and encouragement. Hopefully, there will always be a paper collaboration with Beijing Institute of Technology and Universitas Muhammadiyah Tangerang in further research.

REFERENCES

[1] R. Muwardi and R. R. Adisaputro, "Design Sistem Keamanan Pintu Menggunakan Face Detection," *Jurnal Teknologi Elektro,* vol. 12, no. 3, pp. 120-128, 2021, https://doi.org/10.22441/jte.2021.v12i3.004.

[2] F. Becattini, F. Palai and A. D. Bimbo, "Understanding Human Reactions Looking at Facial Microexpressions With an Event Camera," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 9112 - 9121, 2022, https://doi.org/10.1109/TII.2022.3195063.

[3] R. Muwardi, H. Gao, H. U. Ghifarsyam, M. Yunita, A. Arrizki and J. Andika, "Network Security Monitoring System Via Notification Alert," *Journal of Integrated and Advanced Engineering (JIAE),* vol. 1, no. 2, 2021, https://doi.org/10.51662/jiae.v1i2.22.

[4] R. Muwardi, H. Qin, H. Gao, H. U. Ghifarsyam, M. H. I. Hajar and M. Yunita, "Research and Design of Fast Special Human Face Recognition System," *in 2nd International Conference on Broadband Communications, Wireless Sensors and Powering (BCWSP)*, pp. 68-73, 2020, https://doi.org/10.1109/BCWSP50066.2020.9249452.

[5] K. Bhujbal and S. Barahate, "Custom Object detection Based on Regional Convolutional Neural Network & YOLOv3 With DJI Tello Programmable Drone," *in 7th International Conference on Innovation & Research in Technology & Engineering (ICIRTE)*, 2022, https://doi.org/10.2139/ssrn.4101029.

[6] D. Lee, W. G. La and H. Kim, "Drone Detection and Identification System using Artificial Intelligence," *in International Conference on Information and Communication Technology Convergence (ICTC),* pp. 1131-1133 ,2018, https://doi.org/10.1109/ICTC.2018.8539442.

[7] S. Singha and B. Aydin, "Automated drone detection using YOLOv4," *Drones*, vol. 5, no. 4, p. 95, 2021, https://doi.org/10.3390/drones5030095.

[8] P. Bombelli, A. Savanth, A. Scarampi, S. J. L. Rowden, D. H. Green, A. Erbe, E. Årstøl, I. Jevremovic, M. F. Hohmann-Marriott, S. P. Trasatti, E. Ozer and C. J. Howe, "Powering a microprocessor by photosynthesis," *Energy & Environmental Science*, vol. 15, no. 6, pp. 2529-2536, 2022, https://doi.org/10.1039/D2EE00233G.

[9] A. Y. Montoya, M. Jimenez-Aparicio, J. Hernandez-Alvidrez and M. J. Reno, "A Fast Microprocessor-Based Traveling Wave Fault Detection System for Electrical Power Networks," *in IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT),* pp. 1-5, 2023, https://doi.org/10.1109/ISGT51731.2023.10066370.

[10] D. Naraharisetti, R. Karne, J. Weymouth and A. Wijesinha, "Obsolescence in Operating Systems and Microprocessors," in *IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA)*, pp. 110-115, 2023, https://doi.org/10.1109/SERA57763.2023.10197809.

[11] A. Ravikumar and H. Sriraman, "Acceleration of Image Processing and Computer Vision Algorithms," in *Handbook of Research on Computer Vision and Image Processing in the Deep Learning Era*, pp. 1-18, 2023, https://doi.org/10.4018/978-1-7998-8892-5.ch001.

[12] D. G. Bailey, *Design for embedded image processing on FPGAs*, John Wiley & Sons, 2023. https://doi.org/10.1002/9781119819820.

[13] V. K. Tiwari and M. Meribout, "A Hybrid FPGA-GPU-based Hardware Accelerator for High Throughput 2D Electrical Impedance Tomography," *TechRxiv,* 2023, https://doi.org/10.36227/techrxiv.22262332.v1.

[14] S. Aftab, C. Lal, S. K. Beejal and A. Fatima, "Raspberry Pi (Python AI) for Plant Disease Detection," *International Journal of Current Research and Review*, vol. 14, no. 3, pp. 36-42, 2022, https://doi.org/10.31782/IJCRR.2022.14307.

[15] S. Pattanayak, *Pro Deep Learning with TensorFlow 2.0: A Mathematical Approach to Advanced Artificial Intelligence in Python,* Apress, 2023, https://doi.org/10.1007/978-1-4842-8931-0.

[16] A. Kale, Z. Sun and X. Ma, "Utility of the Python package Geoweaver_cwl for improving workflow reusability: an illustration with multidisciplinary use cases," *Earth Science Informatics,* vol. 16, p. 2955–2961, 2023, https://doi.org/10.1007/s12145-023-01045-0.

[17] T. Diwan, G. Anirudh and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, pp. 9243–9275, 2023, https://doi.org/10.1007/s11042-022-13644-y.

[18] P. Jiang, D. Ergu, F. Liu, Y. Cai and B. Ma, "A Review of Yolo Algorithm Developments," *Procedia Computer Science*, vol. 199, pp. 1066-1073, 2022, https://doi.org/10.1016/j.procs.2022.01.135.

[19] M. Majumder and C. Wilmot, "Automated Vehicle Counting from Pre-Recorded Video Using You Only Look Once (YOLO) Object Detection Model," *Journal of Imaging,* vol. 9, no. 7, p. 131, 2023, https://doi.org/10.3390/jimaging9070131.

[20] Y. Chen, X. Yuan, R. Wu, J. Wang, Q. Hou and M.-M. Cheng, "YOLO-MS: Rethinking Multi-Scale Representation Learning for Real-time Object Detection," *arXiv preprint arXiv:2308.05480,* 2023, https://doi.org/10.48550/arXiv.2308.05480.

[21] Z. Zhang, X. Lu, G. Cao, Y. Yang, L. Jiao and F. Liu, "ViT-YOLO:Transformer-Based YOLO for Object Detection," *in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 2799-2808, 2021, https://doi.org/10.1109/ICCVW54120.2021.00314.
[22] Q. Lin, R. Wang, Y. Wang, F. Zhou and N. Guo, "Target Detection Algorithm Incorporating Visual Expansion Mechanism and Path Syndication," *IEEE Access,* vol. 11, 2023, https://doi.org/10.1109/ACCESS.2023.3280197.
[23] R. Yang, W. Li, X. Shang, D. Zhu and X. Man, "KPE-YOLOv5: An Improved Small Target Detection Algorithm Based on YOLOv5," *Electronics,* vol. 12, no. 4, p. 817, 2023, https://doi.org/10.3390/electronics12040817.
[24] S. Zhai, D. Shang, S. Wang and S. Dong, "DF-SSD: An Improved SSD Object Detection Algorithm Based on DenseNet and Feature Fusion," IEEE Access , vol. 8, pp. 24344 - 24357, 2020, https://doi.org/10.1109/ACCESS.2020.2971026.
[25] Y. Li, H. Dong, H. Li, X. Zhang, B. Zhang and Z. Xiao, "Multi-block SSD based on small object detection for UAV railway scene surveillance," *Chinese Journal of Aeronautics*, vol. 33, no. 6, pp. 1747-1755, 2020, https://doi.org/10.1016/j.cja.2020.02.024.
[26] D. Wan, R. Lu, S. Wang, S. Shen, T. Xu and X. Lang, "YOLO-HR: Improved YOLOv5 for Object Detection in High-Resolution Optical Remote Sensing Images," *Remote Sens,* vol. 15, no. 3, p. 614, 2023, https://doi.org/10.3390/rs15030614.
[27] . Jiang, Z. Li, M. Tian, J. Liu, S. Yi and D. Miao, "Few-Shot Object Detection via Improved Classification Features," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5386-5395 , 2023, https://doi.org/10.1109/WACV56688.2023.00535.
[28] Z. Jiang, L. Zhao, S. Li and Y. Jia, "Real-time object detection method based on improved YOLOv4-tiny," *Computer Vision and Pattern Recognition*, vol. 7, no. 1, 2020, https://doi.org/10.48550/arXiv.2011.04244.
[29] C. Guo, X.-l. Lv, Y. Zhang and M.-l. Zhang, "Improved YOLOv4-tiny network for real-time electronic component detection," *Scientific Reports,* vol. 11, p. 22744, 2021, https://doi.org/10.1038/s41598-021-02225-y.
[30] S.-J. Ji, Q.-H. Ling and F. Han, "An improved algorithm for small object detection based on YOLO v4 and multi-scale contextual information," *Computers and Electrical Engineering,* vol. 105, p. 108490, 2023, https://doi.org/10.1016/j.compeleceng.2022.108490.
[31] H. Zhu, W. Xie, J. Li, J. Shi, M. Fu, X. Qian, H. Zhang, K. Wang and G. Chen, "Advanced Computer Vision-Based Subsea Gas Leaks Monitoring: A Comparison of Two Approaches," *Sensors,* vol. 23, no. 5, p. 2566, 2023, https://doi.org/10.3390/s23052566.
[32] H. Li, C. Li, G. Li and L. Chen, "A real-time table grape detection method based on improved YOLOv4-tiny network in complex background," *Biosystems Engineering*, vol. 212, pp. 347-359, 2021, https://doi.org/10.1016/j.biosystemseng.2021.11.011.
[33] Y. Tang, H. Zhou, H. Wang and Y. Zhang, "Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision," *Expert Systems with Applications,* vol. 211, p. 118573, 2023, https://doi.org/10.1016/j.eswa.2022.118573.
[34] M.-L. Huang and Y.-S. Wu, "GCS-YOLOV4-Tiny: A lightweight group convolution network for multi-stage fruit detection," *Mathematical Biosciences and Engineering ,* vol. 20, no. 1, p. 241–268, 2022, https://doi.org/10.3934/mbe.2023011.
[35] Nehme, E. Coronel, D. A. Barringer, L. G. Romero, M. A. Shafi, W. A. Ross and P. S. Ge, "Performance and attitudes toward real-time computer-aided polyp detection during colonoscopy in a large tertiary referral center in the United States," *Gastrointestinal Endoscopy,* vol. 98, no. 2, pp. 100-109.e6, 2023, https://doi.org/10.1016/j.gie.2023.02.016.
[36] J. Guo and H. Guo, "Real-Time Risk Detection Method and Protection Strategy for Intelligent Ship Network Security Based on Cloud Computing," *Symmetry,* vol. 15, no. 5, p. 988, 2023, https://doi.org/10.3390/sym15050988.
[37] S. T, S. S. Rajest, R. Regin and S. R, "A Review on Using Machine Learning to Conduct Facial Analysis in Real Time for Real-Time Profiling," *International Journal of Human Computing Studies,* vol. 5, no. 2, pp. 18-37, 2023, https://doi.org/10.1109/ICCCIS48478.2019.8974493.
[38] M. Khan, S. Chakraborty, R. Astya and S. Khepra, "Face Detection and Recognition Using OpenCV*," in International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 116-119, 2019, https://doi.org/10.1007/978-1-4842-4261-2_2.
[39] S. Gollapudi, "OpenCV with Python," *Learn Computer Vision Using OpenCV*, pp. 31–50, 2019, https://doi.org/10.1016/j.jclepro.2022.135748.
[40] H. Zhang, C. Li, L. Li, S. Cheng, P. Wang, L. Sun, J. Huang and X. Zhang, "Uncovering the optimal structural characteristics of flocs for microalgae flotation using Python-OpenCV," *Journal of Cleaner Production,* vol. 385, p. 135748, 2023, https://doi.org/10.1016/j.jclepro.2022.135748.
[41] D. D. K. A. A. Levin, A. A. Nechunaev, L. S. Prokhorenko, D. S. Mishchenkov, A. G. Nosova, D. A. Astakhov, Y. V. Poduraev and D. N. Panchenkov, "Assessment of experimental OpenCV tracking algorithms for ultrasound videos," *Scientific Reports volume,* vol. 13, no. 1, p. 6765, 2023, https://doi.org/10.1038/s41598-023-30930-3.

BIOGRAPHY OF AUTHORS

Yuliza is currently an Assistant Professor in Department of Electrical Engineering, Universitas Mercu Buana, Jakarta, Indonesia. She completed Master from Universitas Mercu Buana, Jakarta, Indonesia. E-mail: yuliza@mercubuana.ac.id.

Rachmat Muwardi is currently a Lecturer in the Department of Electrical Engineering, Universitas Mercu Buana, Jakarta, Indonesia. He graduated from the Beijing Institute of Technology in 2020 with a Master's in Electronic Science and Technology. Currently, He declared as a recipient of a China Scholarship Council (CSC) to continue his doctoral program at the Beijing Institute of Technology in September 2022, majoring in Optical Engineering. During his undergraduate, he received a double degree scholarship from Universitas Mercu Buana and Beijing Institute of Technology in Electrical Engineering and Computer Science. His research interest is Object Detection, Target Detection, and Embedded System. He can be contacted at email: rachmat.muwardi@mercubuana.ac.id.

Mustain Rhozaly currently working as a Civil Servant in the Provincial Government of DKI Jakarta, Indonesia. He is the best D-III graduate of Electronic Engineering, Semarang State Polytechnic and has currently completed his Bachelor of Electrical Engineering at Mercu Buana University, Jakarta Indonesia. Have robot programming skills and experience in programming industrial robots such as: handling robots, spot welding robots, and arc welding robots. Won a bronze medal in the Quality Control Circle 2021 industrial contest using a handling robot and an automatic rotary holder. currently studying Artificial Intelligence systems related to small object detection systems using the Yolo V4-Tiny method. He can be contacted by e-mail: rhozaly1@gmail.com.

Lenni currently an Assistant Professor in the Department of Electrical Engineering, Muhammadiyah Tangerang University, Tangerang, Indonesia. She completed Master from Trisakti University. She completed doctoral education at Institute Pertanian Bogor, Indonesia. She can be contacted at email: lenni@umt.ac.id.

Mirna Yunita, received a Master's in Computer Science and Technology from Beijing Institute of Technology, Beijing, China. Currently, as a Ph.D. student at the School of Computer Science and Technology, Beijing Institute of Technology, China. She is interested in related topics in Machine Learning, Web Development, Data Mining, and Bioinformatics. She was a Frontend and Mobile Application Developer in a Logistics & Supply Chain company in Jakarta, Indonesia. She can be contacted at email: mirnayunita@bit.edu.cn.

Galatia Erica Yehezkiel, currently a Master's student in Beijing Institute of Technology, Beijing, China, courtesy of the China Scholarship Council (CSC). She received her undergraduate degree in the same university, majoring in Computer Science. Her areas of interest include Image Processing, Machine Learning, and Internet of Things (IoT). She can be reached from the email: galatia.erica@bit.edu.cn.