Journal of
Robotics, Automation, and Electronics Engineering
Volume 2, Nomor 1, March 2024

JRAEE EDITORS

Journal of
Robotics, Automation, and Electronics Engineering
Volume 2, Nomor 1, March 2024

JRAEE REVIEWERS

Journal of

Robotics, Automation, and Electronics Engineering

Volume 2, Nomor 1, March 204

Table of Contents

JRAEE

Journal of Robotics, Automation, and Electronics Engineering
Vol. 2, No. 1, 2024, pp. 1-10
e-ISSN 3025-3780
https://journal.uny.ac.id/publications/jraee

Implementation of Kalman Filter With Pi-Controller for Temperature Sensor in Fish Pond Monitoring System

Wardah Alfalah[a,1,*], Varlenda Sarma Quraini[a,2], Nabilah Qurrotuayun[a,3]

[a] Universitas Negeri Yogyakarta
[1] wardahalfalah.2020@student.uny.ac.id; [2] varlendasarma.2020@student.uny.ac.id;
[3] nabilah0707ft.2020@student.uny.ac.id
[*] Corresponding Author

ARTICLE INFO

Article History
Received 04 Nov. 2023
Revised 03 Dec. 2023
Accepted 10 Feb 2024

Keywords
Kalman Filter;
PI Controller;
Firebase Realtime Database;
Temperature Sensor;
Fish Pond.

ABSTRACT

The research aims to determine monitoring the temperature in fish ponds is crucial for successful cultivation, especially in tropical climates that often experience hot weather. This study proposes an approach using the Kalman Filter method and PI (Proportional-Integral-Derivative). Aside from that, to controller to improve the accuracy of monitoring the temperature in fish ponds. Integration with the Firebase Realtime Database allows for real-time data monitoring. Testing was conducted by comparing the DS18B20 temperature sensor without a filter with three variations of the Kalman Filter and PI controller. The results show that using Kalman Filter 3 with the PI controller resulted in a significant reduction in error and noise compared to using Kalman Filter alone. In conclusion, the integration of the Kalman Filter and PI controller with the Firebase Realtime Database can improve the accuracy of monitoring the temperature in fish ponds and has positive implications for increasing efficiency and fish welfare.

Tujuan penelitian adalah mengetahui pemantauan suhu dalam kolam ikan menjadi krusial untuk keberhasilan budidaya, terutama dalam iklim tropis yang seringkali mengalami cuaca panas. Penelitian ini mengusulkan pendekatan menggunakan metode Kalman Filter dan pengontrol PI (Proportional-Integral-Derivative). Selain itu juga untuk meningkatkan akurasi pemantauan suhu kolam ikan. Integrasi dengan Firebase Realtime Database memu- ngkinkan pemantauan data secara real-time. Pengujian dilakukan dengan membandingkan sensor suhu DS18B20 tanpa filter dengan tiga variasi Kalman Filter dan pengontrol PI. Hasilnya menunjukkan bahwa penggunaan Kalman Filter 3 dengan pengontrol PI menghasilkan penurunan error dan noise yang signifikan dibandingkan dengan penggunaan Kalman Filter saja. Kesimpulannya, integrasi Kalman Filter dan pengontrol PID dengan Firebase Realtime Database dapat meningkatkan akurasi pemantauan suhu kolam ikan dan memiliki implikasi positif dalam meningkatkan efisiensi dan kesejahteraan ikan.

1. Introduction

Nowadays, the growth of technology creates an increasing demand for devices that can support various human activities. Technological advancements are expected to provide many positive benefits in daily life. The presence of technology not only opens up opportunities for more innovations but can also enhance better performance in the fish farming sector. Especially in very large industrial-scale fish farms. From the graph above, based on data from KKP Statistics, aquaculture production from 2019-2021 has decreased. This has resulted in fish farmers experiencing big enough losses [1].

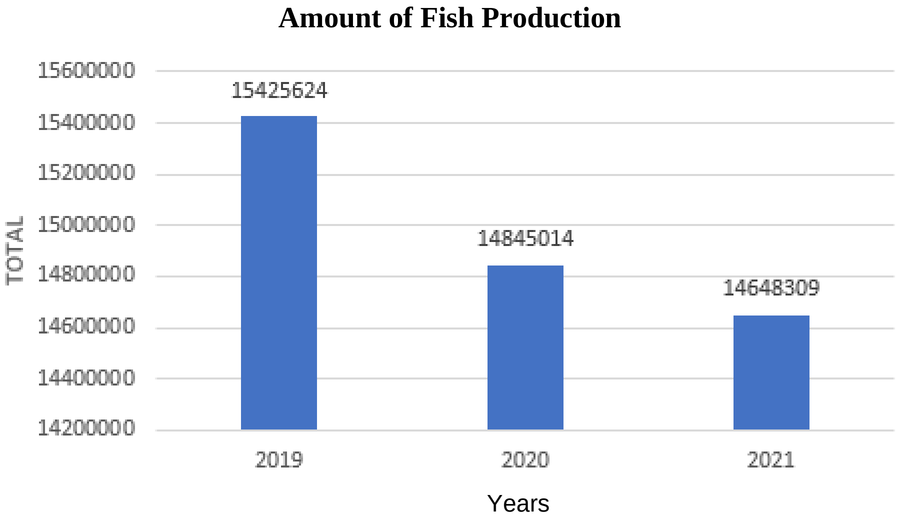


Fig. 1. Data on the amount of fish production

It was reported by liputan6.com that extreme weather caused Sawohan's ponds to fail in their harvest. In the life of the fish, this interaction will have a significant impact on the conditions for success in the fish cultivation sector, becoming a determining factor in achieving the goals of the Ministry of Education and Culture.

According to information obtained from the Central Bureau of Statistics in 2021 regarding the instability of temperature weather in Indonesia, this is a serious challenge for the fish farming sector. Therefore, to overcome this problem, water measurement parameters, including water quality parameters, that can be monitored in fish farming are regulated following the provisions contained in Government Regulation No. 82 of 2001. This regulation aims to establish class two and three water quality standards applicable to fish farming.

In facing this challenge, the importance of continuously monitoring the temperature in the fish pond becomes very important [2][3]. One effective method for carrying out this monitoring is to apply the Kalman filter method to the temperature sensor. Kalman filter is a signal processing technique that can improve estimates of measured data, overcome interference, and increase the accuracy of temperature measurements [4][5]. By applying the Kalman filter, data obtained from temperature sensors can be analyzed and refined to produce more precise temperature estimates related to the conditions of the fish pond cultivation industry sector.

The main advantage of the Kalman filter method lies in its capacity to estimate the state of the system and make predictions of the system state, even in conditions where data is limited or there is interference with sensor readings [6]. In the context of monitoring temperature in fish ponds during long periods of dry season, the Kalman filter plays a role in reducing the effects of unwanted temperature fluctuations, ensuring more accurate temperature estimation, and providing more effective control over possible temperature changes. Thus, the application of the Kalman filter is essential in maintaining environmental conditions in fish ponds to meet the needs of fish and support their well-being in the face of extreme weather changes, such as those that occur during prolonged dry seasons.

Therefore, this project emphasizes the important aspect of maintaining fish welfare in fish ponds through efficient monitoring and regulation of water temperature using the Firebase Realtime Database enabling temperature and humidity data from sensors to be stored directly. Firebase provides fast and structured data access through mobile or web applications, enabling the user to monitor fish pond conditions in real time and respond quickly to changes [7]. Thus, the integration of Firebase in a fish pond monitoring system not only improves monitoring efficiency, but also allows users to take appropriate actions based on available information, to support fish welfare, and to improve the overall optimization of the fish pond environment.

This method can be achieved by utilizing high technology, such as applying the Kalman filter method to the DS18B20 temperature sensor and integrating it with the ESP32 microcontroller. The main advantage of using the Kalman filter method is its ability to predict system status by estimating the state of the system, which has the effect of increasing the precision and accuracy of the temperature sensor measurements. Even in situations with limited data, the use of these filters allows the sensor to achieve balance more quickly than conventional methods that only take sensor measurements. Through the implementation of the Kalman filter and PI control, our efforts are aimed at increasing the accuracy of temperature estimation, achieving more stable temperature control, and maintaining the water temperature to always suit the needs of the fish within the framework of the fish pond monitoring system [8].

2. Research Method

This research utilizes an experimental approach that involves direct testing of the Kalman Filter method and PI controller in the context of fish pond temperature monitoring. The experimental approach was chosen because it allows the researcher to control certain variables that affect the results of the experiment, thereby allowing a more accurate analysis of the effectiveness of the method being tested.

In the initial stage of the research, an experimental design was carried out which included selecting a representative type of fish pond, selecting an appropriate temperature sensor, and setting the hardware configuration. The fish ponds were selected concerning adequate size for accurate temperature monitoring as well as environmental conditions suitable for the needs of the fish [9][10].

The Kalman Filter method was chosen because of its ability to process sensor data to produce more accurate temperature estimation, especially in the face of noise or interference in sensor readings [11]. Meanwhile, the PI controller was chosen to optimize the system response to temperature changes that occur in the fish pond [12][13].

During the testing process, the data collected from the temperature sensor is processed using the Kalman Filter method and a PI controller. The processed data is then compared with the actual temperature data measured directly to evaluate its accuracy. The tests are conducted under controlled conditions to ensure the validity of the results obtained. Using this experimental approach, it is expected that this research can make a significant contribution in improving fish pond temperature monitoring and ultimately, supporting the success of fish farming.

2.1 Materials and Tools

- **ESP-32**, The ESP-32 microcontroller is used to read and process data from the sensors used in this project. In addition, it is also used to send data to the Firebase real-time database.
- **DS18B20 Sensor**, The DS18B20 sensor is used to read the water temperature in fish ponds. This sensor can be accessed using the ESP-32 microcontroller.
- **DHT22 Sensor**, The DHT22 sensor is used to read the air temperature around the fish pond. This sensor can also be accessed using the ESP-32 microcontroller.
- **Resistors**, Resistors are used to limit current and control voltage levels in a circuit, providing resistance to electric current.
- **18650 battery**, An 18650 battery is used as the power source in this project, with a nominal voltage of about 3.7 volts and a capacity measured in milliampere-hours (mAh), ranging from 2000mAh to over 3500mAh.

- **TP4056 Module**, The TP4056 module is used to charge the battery to keep it safe and efficient in this project.
- **Jumper Cables**, Jumper cables are used as a link between two points in this project circuit.
- **Solder**, Solder is used to connect two or more components in this project.
- **Tin**, Tin is used as a process for connecting tool components in this project.

2.2 Data Collection Procedure

- **Installation of DS18B20 Temperature Sensor**: The DS18B20 temperature sensor is installed outside the fish pond keeping in mind the optimal distance to ensure accurate temperature monitoring.
- **Integration with ESP32 Microcontroller**: The temperature sensor is interfaced with the ESP32 microcontroller. A WiFi connection is configured to transmit the temperature data detected by the sensor to the Firebase Realtime Database. This allows direct and real-time access to data remotely over the internet.
- **Initial Testing**: Before the actual data collection begins, initial testing is conducted to ensure that the sensors are functioning properly and the data can be sent appropriately to the Firebase Realtime database. This testing aims to identify and fix potential technical issues before the actual data collection process begins.
- **Testing the Kalman Filter Method and PI Controller**: After the initial setup is complete, testing of the Kalman Filter method and PI controller on the data received from the temperature sensor is carried out. This test aims to evaluate the effectiveness of these methods in improving the accuracy of fish pond temperature monitoring. These steps are carefully performed to ensure the validity and reliability of the results obtained.

2.3 Data Analysis

- **Data Processing**: The temperature data received from the sensors will be processed using the Kalman Filter method and a PI controller. The Kalman Filter is used to remove noise and produce a more accurate temperature estimation, while the PI controller is used to correct the desired temperature set point.
- **Comparison of Results**: After the data has been processed with and without the filter, a comparison is made between the results. This comparison aims to evaluate the accuracy improvement provided by the Kalman Filter and PI controller in improving the temperature estimation.
- **Change Analysis**: In addition, analyses were also conducted to understand the change in temperature estimation before and after the use of the filter. This helps in determining the effectiveness of the filter in improving the accuracy of fish pond temperature monitoring.
- **Interpretation of Results**: The analyzed results will be interpreted to conclude the effectiveness of the Kalman Filter method and PI controller in improving the accuracy of fish pond temperature monitoring. These findings will form the basis for recommendations regarding the use of these methods in practical applications.

3. Sensor Calibration

DS18B20 Temperature Sensor Calibration: The DS18B20 temperature sensor is calibrated through a series of tests involving three different conditions: plain water, ice cubes, and hot water. Each test is carried out sequentially by inserting a temperature sensor probe into the media, while the measurement value is displayed on the Arduino IDE serial monitor. The purpose of this calibration is to ensure that the DS18B20 sensor can provide accurate measurements according to actual conditions. DHT22 Temperature Sensor Calibration: Calibration of the DHT22 sensor is carried out using an accurate comparison tool, namely the HTC-2 Temperature Clock. Temperature measurements are carried out in environments with different temperature variations, ensuring that both devices (Temperature Clock HTC-2 and DHT22 sensor) are in stable condition. Measurement data from both devices is displayed on the Arduino IDE serial monitor for comparison. This process aims to verify the measurement accuracy of the DHT22 sensor and ensure that the sensor can provide accurate values according to actual environmental conditions. With this approach, the accuracy of the DS18B20 and DHT22 temperature sensors is verified through a systematic and controlled calibration process. The results of this calibration will be an important basis for ensuring the accuracy of sensor measurements before being implemented in fish pond temperature monitoring applications.

4. Result and Discussion

4.1 DS18B20 Temperature Sensor Calibration

The calibration process is carried out to ensure that the DS18B20 temperature sensor provides accurate measurement results. The test results show variations in temperature results in some test samples. In 30 tests, the average error of the DS18B20 sensor was 0.33°C. Although this level of accuracy is slightly below the datasheet specification which mentions an accuracy level of 0.33°C, the average error is 0.33°C. ±0.5°C within the temperature range of -10°C to 85°C, but the sensor is still reliable for temperature measurements within that range.

Table 1. Calibration of DS18B20 temperature sensor

No	DHT Temp. Sensor	Temp. Clock HTC-2	Error
1	20.31	20.8	0.11
2	20.61	20.9	0.29
:	:	:	:
30	38.75	38.8	0.05
average = 0.33			

4.2 DHT22 Sensor Calibration

The calibration process of the DHT22xsensor involves comparing the measurement results with a comparison tool that has high accuracy. The calibration results show that the DHT22 sensor provides fairly accurate results, with an average error of about 0.49°C. This value is below the 0.5°C limit which is an acceptable level of accuracy for the temperature sensor in this application.

Table 2. DHT22 Sensor Calibration

No	DHT Temp. Sensor	Temp. Clock HTC-2	Error
1	23.9	24.0	0.1
2	23.8	24.1	0.3
:	:	:	:
30	29.7	30.9	1.2
average = 0.49			

In conclusion, the calibration results show that both temperature sensors, DS18B20 and DHT22, provide fairly accurate and reliable measurement results for use in monitoring fish pond temperatures. Although there is a slight variation in the measurement results, the accuracy of the sensors is still within the acceptable range for this application. Thus, the calibration results ensure that the data generated by both sensors is reliable and accurate for use in the fish pond temperature monitoring system [14][15].

4.3 Determination of The Best Kalman Filter

This test aims to evaluate the effectiveness of the Kalman filter in reducing noise in DS18B20 temperature sensor data. The testing process is done by taking data for 5 minutes with an estimation per second. The test results show that Kalman filter 3 with Q=10 and R=10 values gives the best results in reducing noise in temperature sensor data. The use of Kalman filter 3+PI can improve the accuracy of temperature estimation and reduce noise significantly. The integration of Kalman filter 3 with the PI controller can be an effective solution to improve the performance of the fish pond temperature monitoring system.

Table 3. Absolute Error of The Best Kalman Filter

KF1	**KF2**	**KF3**
0.28	0.19	0.1

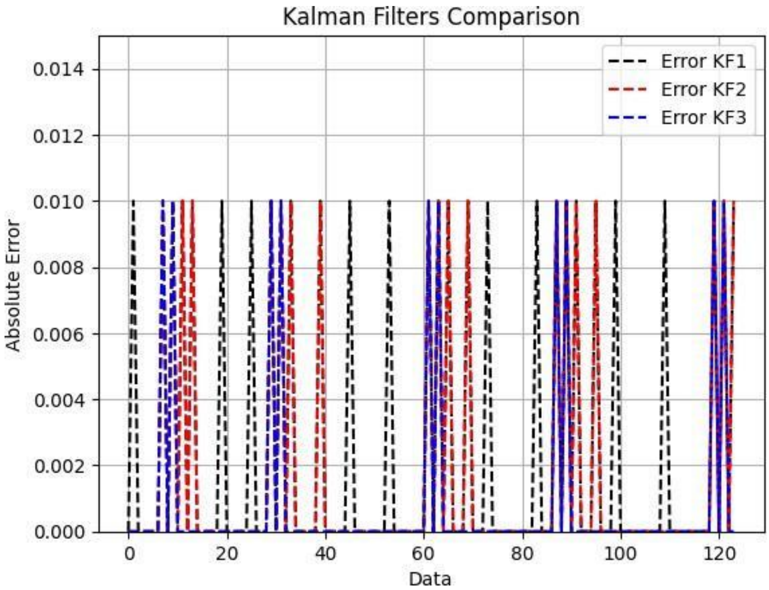


Fig. 2. Best Kalman filter

4.4 Testing Kalman Filter with PI Kalman Filter

The integration of Kalman filter 3 with the PI controller shows better performance in overcoming noise and improving the accuracy of temperature estimation. This test has the potential to provide great benefits in fish pond temperature monitoring, where accurate estimation is critical for fish welfare and overall production performance.

4.5 Kalman Filter Testing with The Best PI Kalman Filter Against Noise and Without Noise

This test aims to evaluate the performance of the Kalman Filter with the best Kalman Filter PI in overcoming noise interference and improving the accuracy of fish pond temperature estimation. The testing process is carried out by taking DS18B20 temperature sensor data for 5 minutes with an estimation per second. This aims to simulate the condition of temperature sensor measurements in an actual fish pond environment, where temperature fluctuations and noise interference can occur unexpectedly [17].

The test results show that the integration of Kalman Filter 3 with the PI controller can reduce the measurement error due to noise. Kalman Filter 3+PI produces an error value of 13.88, while Kalman Filter 3 without PI controller produces a value of 14.18. This shows that the integration of PI in Kalman Filter 3 has a positive impact on reducing noise and increasing the accuracy of temperature estimation [18].

Analyses were also carried out in conditions without noise. Kalman Filter 3+PI without noise produces an error value of 3.68, while Kalman Filter 3 without noise produces a value of 7.91. This result shows that Kalman Filter 3+PI is also better at reducing measurement errors on the sensor when conditions are without noise.

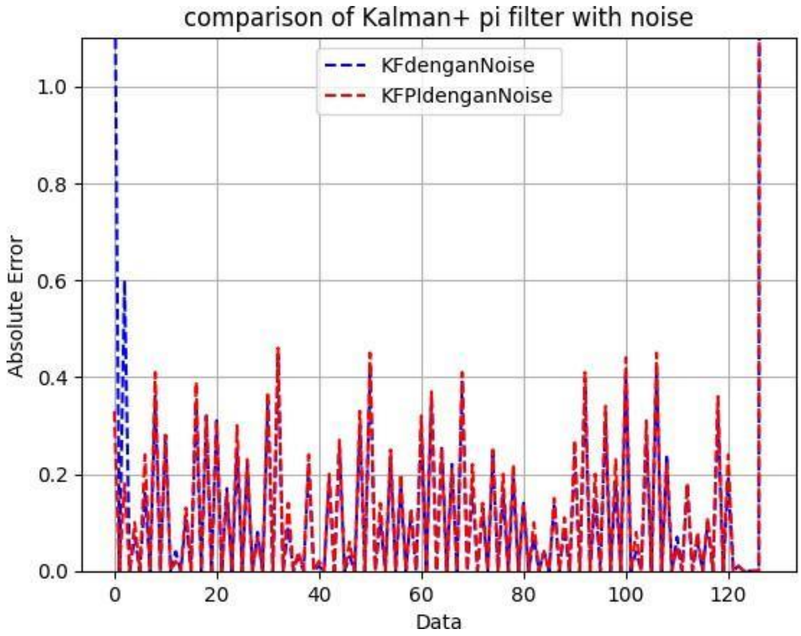


Fig. 3. KFPI graph with noise

Table 4. Absolute Error of Kalman Filter With Noise

KF1	KF2
14.18	13.88

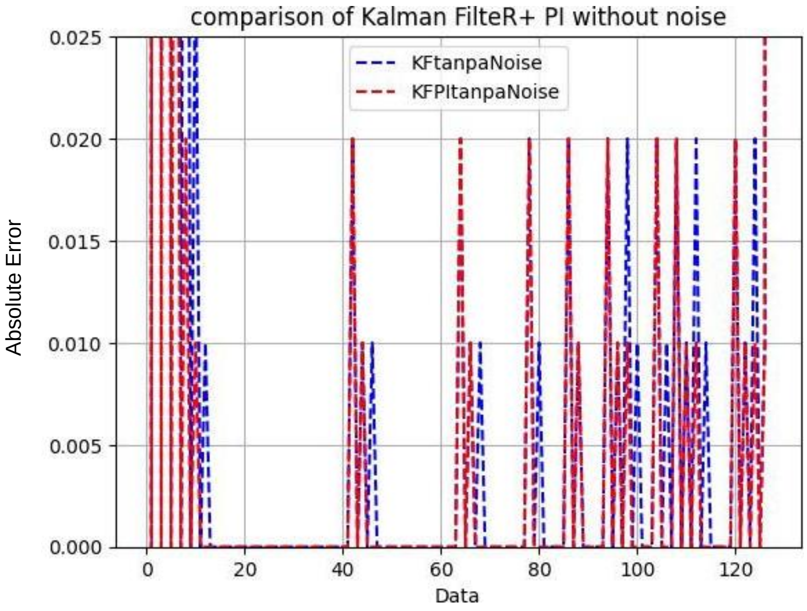


Fig. 4. KFPI graph without noise

From these test results, it can be concluded that the integration between Kalman Filter 3 and PI controller has better performance in overcoming noise and improving temperature estimation accuracy. This has great potential in monitoring the temperature of fish ponds, where accurate estimation is critical to fish welfare and overall production performance [16].

Table 5. Absolute Error of Kalman Filter Without Noise

KF1	KF2
7.91	3.68

4.6 Firebase Realtime Data Testing

Realtime Firebase data testing is an important stage in the development of applications that use the Firebase platform. The main goal is to ensure that the connection between the application and the database runs smoothly and that the data displayed in the application is accurate and matches what is stored in the database. This test ensures that the Firebase Realtime Database function operates as expected.

The first step in this test is to populate data into the database and ensure that the data is immediately reflected in the application interface using the Real-time Database. This was done to verify that the data synchronization process was automatic and that data changes were immediately visible to the application user.

In addition, these tests emphasize security and performance aspects. Developers need to ensure that data retrieval from the database follows predefined access rules and that the system remains responsive and optimized, especially when there is an increase in the number of users or data volume.

Conducting thorough Firebase Realtime data testing is critical to improving application quality, ensuring system reliability, and providing a seamless user experience. By conducting rigorous testing, developers can identify and fix potential issues before the app is released to end users, thereby increasing user confidence in the developed app.

4.7 Overall System Testing

The overall system test is carried out to validate operational consistency and effective integration by previously prepared design plans. This testing process involves verifying the function of each component involved in the system, from hardware to servers, as well as evaluating the availability of sensor readings and real-time data transmission that can be accessed via the user's cellphone.

The main purpose of system testing is to ensure that each element in the system can operate according to the desired standards and can interact with each other effectively. This is very important to ensure that the system can function properly in a real operational environment.

The system testing process was carried out thoroughly, starting from testing individual components to testing the integration between components. Each stage of testing was meticulously carried out to ensure that every aspect of the system was working properly and in line with expectations.

5. Conclusion

Implementation of Kalman filter in water temperature monitoring system in a fish pond using DS18B20 and DHT22 temperature sensors has been successfully carried out. This system is designed to monitor water temperature quality in the fishing industry. Data from both sensors can be displayed in real time, enabling automatic and instant data synchronization between connected devices. Communication between devices uses the HTTP protocol. Component testing has verified that the DS18B20 and DHT22 sensors function according to datasheet specifications. The DS18B20 sensor has an accuracy level of around ±0.33°C, while the DHT22 sensor has an accuracy of around 29.7°C with an average error of around 0.49°C. Both sensors meet established accuracy standards, with error rates lower than the 0.5°C limit. The use of Kalman filters, especially Kalman filter 3, has been proven to increase the accuracy of temperature estimates on the DS18B20 sensor. With the Kalman filter, fluctuations in temperature sensor values can be suppressed, producing more consistent and accurate estimates. The use of the PI Controller together with the Kalman filter also improves the system's response to changes in temperature estimates, by adjusting the KP and KI values accordingly. The system has functioned according to the planned design. Sensors can take readings and send data directly to a real-time database using the HTTP protocol. Data can be monitored directly via the Firebase web platform, which can be accessed in real-time via the user's mobile device.

The results of this study show that the implementation of the Kalman filter in a fish pond temperature monitoring system can improve the accuracy of temperature estimation and system response to temperature changes. This system can be an effective solution to monitor the quality of water temperature in fish ponds in real time.

References

[1] P. A. Rosyady and M. A. Agustian, “Sistem Monitoring dan Kontrol Keasaman Larutan dan Suhu Air pada Kolam Ikan Mas Koki dengan Smartphone Berbasis IoT,” Techne´: Jurnal Ilmiah Elektroteknika, vol. 21, no. 2, pp. 169-188, 2022, doi: 10.31358/techne.v21i2.317.

[2] S. Indriyanto, F. T. Syifa, and H. A. Permana, “Sistem Monitoring Suhu Air pada Kolam Benih Ikan Koi Berbasis Internet of Things,” TELKA-Jurnal Telekomunikasi, Elektronika, Komputasi dan Kontrol, vol. 6, no. 1, pp. 10–19, 2020, doi: 10.15575/telka.v6n1.10-19.

[3] R. Pramana, “Perancangan Sistem Kontrol Dan Monitoring Kualitas Air Dan Suhu Air Pada Kolam Budidaya Ikan,” Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan, vol. 7, no. 1, pp. 13–23, doi: 10.31629/sustainable.v7i1.435.

[4] G. F. Welch, Kalman filter, Computer Vision: A Reference Guide, pp. 1–3, Springer, 2020.

[5] S. Mohinder, Kalman Filtering Theory and Practice Using Matlab, 2001

[6] A. Winursito, I. Masngut, and G. N. P. Pratama, “Development and Implementation of Kalman Filter for Iot Sensors: Towards a Better Precision Agriculture,” In 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI) IEEE, pp. 360-364, doi: 10.1109/ISRITI51436.2020.9315464.

[7] S. Suriana, A. P. Lubis, and E. Rahayu, “Sistem Monitoring Jarak Jauh Pada Suhu Kolam Ikan Nila Bangkok Memanfaatkan Internet of Things (IOT) Berbasis NODEMCUESP8266,” Jurnal Teknologi dan Sistem Informasi (JUTSI), vol. 1, no. 1, pp. 1–8, 2021, doi: 10.33330/jutsi.v1i1.1004.

[8] K. M. Ilham, “Rancang Bangun Sistem Kendali Suhu Air Kolam Benih Lobster Dengan Pengendali Pid Berbasis Mikrokontroler ESP32”, Fakultas Teknik, 2023.

[9] A. S. Wicaksana, and B. Suprianto, “Rancang Bangun Sistem Pengendalian PH Air Pada Tambak Ikan Bandeng Menggunakan Kontroller PID Berbasis LabView,” Jurnal Teknik Elektro, vol. 9, no. 2, 2020, doi: 10.26740/jte.v9n2.p%25p.

[10] S. Mulyani, Idris, *et al.*, “Analisis Perencanaan Pembangkit Listrik Tenaga Surya Sebagai Catu Daya Aerator dan Alat Pemberi Pakan Ikan,” In Seminar Nasional Teknik Elektro dan Informatika (SNTEI) 2023, vol. 9, no. 1, pp. 59–66.

[11] A. Ma’arif, I. Iswanto, A. A. Nuryono, and R. I. Alfian, “Kalman Filter for Noise Reducer on Sensor Readings,” Signal and Image Processing Letters, vol. 1, no. 2, pp. 50–61, 2019, doi: 10.31763/simple.v1i2.2.

[12] T. A. Rahmawati, G. N. P. Pratama, A. I. Cahyadi, S. Herdjunanto, and others, “A Remedy Design of Pi Controller for Liquid Level Control,” In 2018 International Conference on Information and Communications Technology (ICOIACT) IEEE, pp. 661-666, doi: 10.1109/ICOIACT.2018.8350821.

[13] A. Maarif, R. D. Puriyanto, and F. R. T. Hasan, “Robot Keseimbangan Dengan Kendali Proporsional-Integral-Derivatif (PID) dan Kalman Filter,” IT Journal Research and Development, vol. 4, no. 2, 2020, doi: 10.25299/itjrd.2020.vol4(2).3900, doi: 10.25299/itjrd.2020.vol4(2).3900.

[14] L. Lasmadi, A. I. Cahyadi, R. Hidayat, and S. Herdjunanto, “Inertial Navigation for Quadrotor Using Kalman Filter With Drift Compensation,” International Journal of Electrical and Computer Engineering, Publisher: IAES Institute of Advanced Engineering and Science, vol. 7, no. 5, p. 2596, 2017, doi: 10.11591/ijece.v7i5.pp2596-2604.

[15] M. Han, K. Kim, J. Lee, *et al.*, “Implementation of Unicycle Segway Using Unscented Kalman Filter in LQR Control,” In 2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI) IEEE, pp. 695–698, doi: 10.1109/URAI.2013.6677427.

[16] T. Herlambang, “Estimasi Posisi Magnetic Levitation Ball Menggunakan Metode Akar Kuadrat En-Semble Kalman Filter (Ak-Enkf)”, REM (Rekayasa Energi Manufaktur) Jurnal, vol. 2, no. 1, pp. 45–49, 2017, doi: 10.21070/r.e.m.v2i1.768.

[17] M. Khairudin, S. Yatmono, I. M. Nashir, F. Arifin, W. Aulia, *et al.*, “Exhaust Fan Speed Controller Using

Fuzzy Logic Controller," Journal of Physics: Conference Series, vol. 1737, no. 1, p. 012046, 2021, doi: 10.1088/1742-6596/1737/1/012046.

[18] S. Nurmaini, and S. Pangidoan, "Localization of Leader-Follower Robot Using Extended Kalman Filter", Computer Engineering and Applications," vol. 7, no. 2, 2018, doi: 10.18495/COMENGAPP.V7I2.253.

JRAEE

Journal of Robotics, Automation, and Electronics Engineering
Vol. 2, No. 1, 2024, pp. 11-19
e-ISSN 3025-3780
https://journal.uny.ac.id/publications/jraee

Adaptive Bounding Box Coordinate Adjustment on License Plate Character Detection Using Machine Learning

Ahmad Taufiq Musaddid[a,1,*]

[a] Universitas Negeri Yogyakarta
[1] ahmadtaufiqmusaddid@uny.ac.id
[*] Corresponding Author

ARTICLE INFO

Article History
Received 23 Nov. 2023
Revised 28 Jan. 2023
Accepted 25 Feb. 2024

Keywords
ANPR;
Machine Learning;
CNN;
Object Detection;
License Plate

ABSTRACT

Effective law enforcement, including the use of the ANPR (Automatic Number Plate Recognition) system, is essential for reducing the number of road traffic accidents. ANPR involves plate localization, character segmentation, and recognition to build a minimum system. This study aims to improve a character segmentation method using a detection approach to address issues like noisy or modified plates. We propose an adaptive improvement on an established sliding window technique, by integrating a CNN (Convolutional Neural Network) for bounding box coordinate adjustment to handle various plate conditions. The proposed method was tested on 280 license plate images and improved the average IoU (Intersection over Union) from 0.4811 to 0.8980. Hence, the recall and precision of the model could be improved to increase any character recognition performance.

Penegakan hukum yang efektif, termasuk penggunaan sistem ANPR (*Automatic Number Plate Recognition*) sangat penting untuk mengurangi jumlah kecelakaan lalu lintas. ANPR melibatkan lokalisasi pelat nomor, segmentasi karakter, dan pengenalan untuk membangun sistem minimum. Penelitian ini bertujuan untuk meningkatkan metode segmentasi karakter yang menggunakan pendekatan deteksi untuk mengatasi masalah seperti pelat nomor yang berderau atau dimodifikasi. Kami mengusulkan peningkatan adaptif pada teknik *sliding wondow* yang sudah ada, dengan mengintegrasikan CNN (*Convolutional Neural Network*) untuk penyesuaian koordinat *bounding box* guna menangani berbagai kondisi pelat nomor. Metode yang diusulkan diuji dengan 280 gambar pelat nomor dan mampu meningkatkan rata-rata IoU (*Intersection over Union*) dari 0,4811 menjadi 0,8980. Sehingga performa model seperti *recall* dan *precision* dapat ditingkatkan untuk mendukung performa pengenalan karakter di masa mendatang.

1. Introduction

The rise in accident rates in Indonesia suggests a lack of public awareness regarding traffic safety. According to BPS (Badan Pusat Statistika), there were 109 thousand vehicle accidents in 2018, increasing to 139 thousand cases in 2022 [1]. To curb this trend, consistent monitoring and enforcement of traffic regulations are imperative. Rezapour et al. [2] highlighted that such measures can effectively lower accident rates. Hence, the deployment of automatic surveillance technology is crucial to support law enforcement efforts in this regard, such as ANPR (Automatic Number Plate Recognition).

To implement ANPR, there are 3 general steps to be considered. The first step is to localize the plate that is being analyzed on a vehicle. The second is to segment the characters on the plate. And last is to recognize the characters that are extracted. The whole step should yield all information in the form of a short list of characters or even a string. Hence, the latter step's performance is decided by the performance of the previous steps. In this study, the main focus is the character segmentation step but with a detection approach.

Several studies have conducted character segmentation on license plates using machine learning, with a segmentation or detection approach. In [3], a combination of sliding windows and CNN (Convolutional Neural Network) was employed for character detection and recognition on Taiwanese vehicle plates. The CNN used was a modified version of YOLO (You Only Look Once), with a reduction in the number of CNN layers. The concept utilized was SWSCD (Sliding Window Single Class Detection), where each extracted area is fed into a single class classifier, thus employing multiple classifiers for character detection and recognition. In [4], Mask-RCNN was utilized for character detection and recognition on vehicle plates. The detection implemented in Mask-RCNN operates at the pixel level, using Resnet-50 architecture as the backbone. However, even though this object detection approach could handle plate noise, it wasn't sufficient to handle the overlapping characters. Researchers [5] conducted character segmentation on Chinese vehicle plates with 7 characters, using a concept of detecting areas between characters. They utilized a CNN inspired by Darknet-19 architecture, reducing it to 18 convolutional layers and adding an output layer with 6 neurons to determine 6-character boundaries. The limitation of this study arises if the plate has fewer or more than 7 characters. In [6], sliding windows and CNN were employed for character recognition on Indonesian plates, using a mechanism that detects characters among 36 actual character classes. Variations in sliding window sizes and positions were considered based on average character sizes obtained from measuring character contours on a binary plate image. Additionally, the CR-NET framework, inspired by Fast-YOLO, was adapted for character detection on Brazilian vehicle plates in [7], while a modified version of YOLOv3 was used for Bangladeshi vehicle plates in [8], both employing one-stage detection approaches without any area proposal technique.

One of the problems of license plate character detection is the condition of the observed plates. A noisy or defective plate could reduce the performance of any ANPR system. These plate conditions are commonly found in Indonesia which has high mobility of traffic. Furthermore, in some cases, license plates in Indonesia are modified into nonstandard forms. To tackle this problem, an adaptive approach to detect characters on the variation condition of the captured plate can be used. This research aims to create adaptive adjustment on an established license plate character detection method and to improve the performance. The established method is sliding windows was utilized in some mentioned studies. We added one more step to create a CNN to do regression on RoI (Region of Interest) of detection, such that the fixed-size bounding boxes could be more adaptive and capable of handling various conditions and perspectives of captured plates.

2. Methodology

2.1 Data Collection and Preparation

To accommodate two main activities, training and testing, the dataset used in this study is divided into two types. The first is the training dataset and the second is a testing dataset. The dataset contains 740 cropped black Indonesian license plate images with a size of 300×94 pixels. For training, 460 images are taken, which contain 3503 characters. And the rest, 280 images which contain 2026 characters are used for testing. The ground truth of the dataset is a collection of (x_1, x_2, y_1, y_2) coordinates of each character on a license plate.

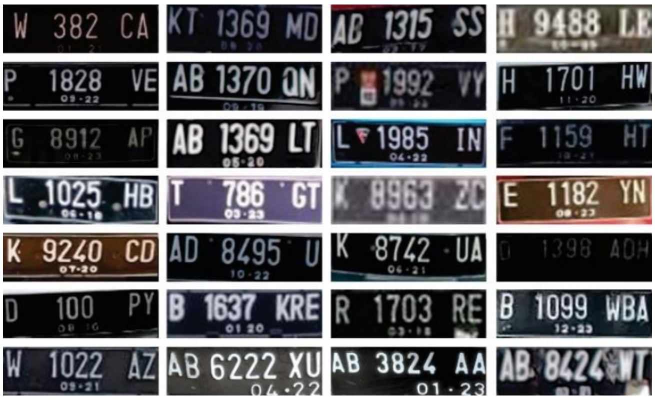


Fig. 1. Training set

Table 1. Dataset

Dataset	**Total**	
	Plate	**Character**
Training	460	3503
Testing	280	2026

2.2 Research Design and Implementation

A sliding window technique is utilized to detect characters on a license plate. This sliding window provides regions to a CNN classifier. This CNN determines whether a region is a character or not. The size of the sliding window is fixed at 25×95 pixels with a stride of 1 pixel. A window sweeps from left to right to extract and propose regions to the classifier.

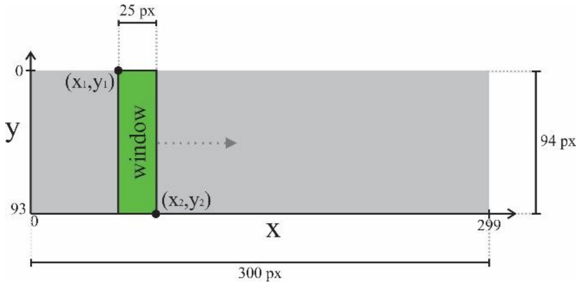


Fig. 2. Visualization of sliding window

The classifier (CNN) uses the binary classification concept as it only indicates if a region is considered a character or not. The total number of proposed regions is 274, these regions are fed into the CNN one by one. From these 274 classified regions, we need to associate each potential region with a character's estimated location. To solve this, a clustering algorithm like HAC (Hierarchical Agglomerative Clustering) is used to cluster each potential region into groups of related characters by calculating all distances.

The mentioned detection pipeline is called the detector in this paper from this point. To accomplish the improvement of this detector, a method is proposed as a continuation of our previous work [9]. This method uses additional CNN to do regression on detected characters. The regression is focused on the coordinates of bounding boxes. For further explanation, this method will be discussed in the following sections as a bounding box regressor. For the implementation of all processes, we use the TensorFlow framework and Python programming language.

2.3 Proposed Method

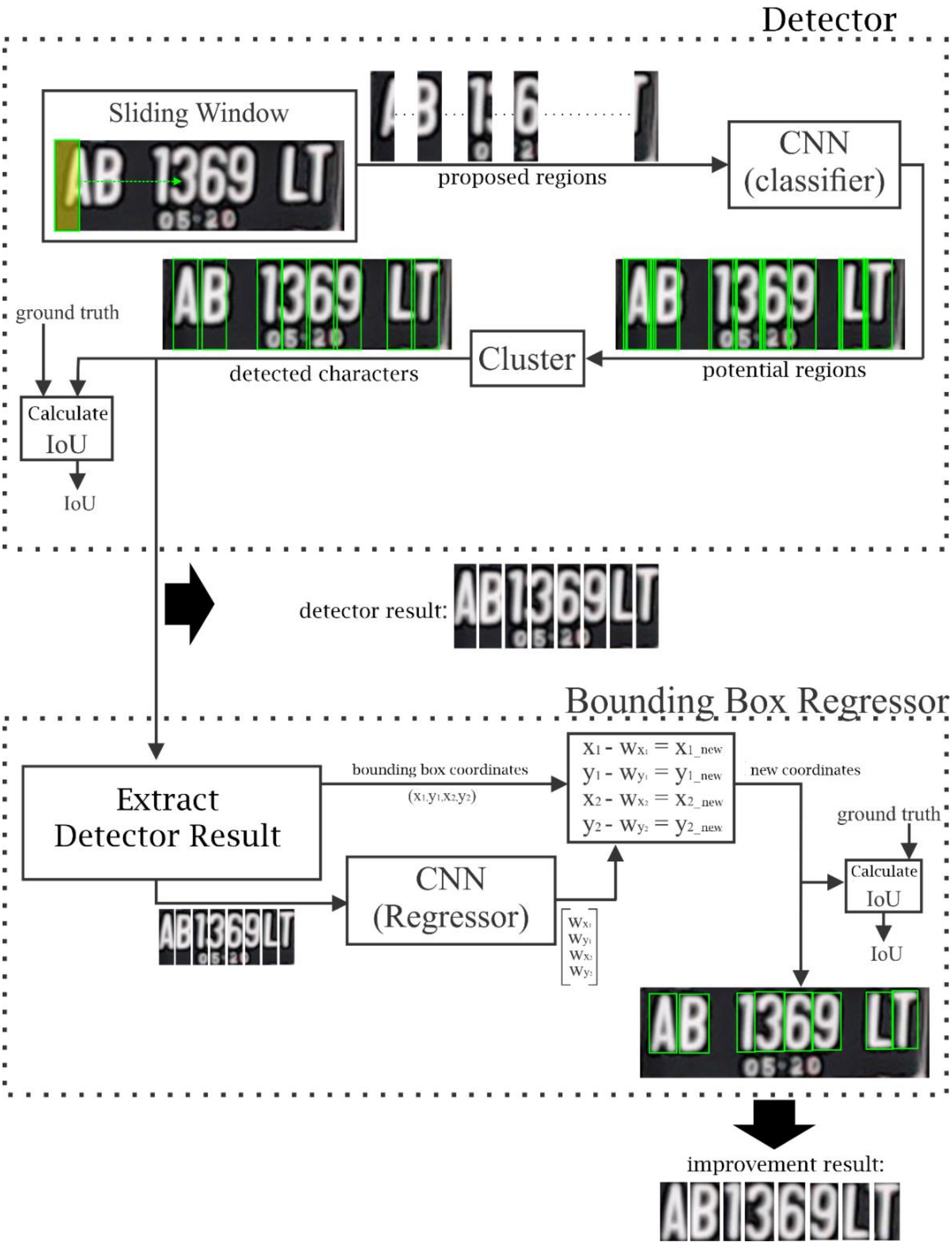


Fig. 3. Proposed Method

The bounding box produced by the character detector has a fixed size. This will have an impact on the low level of overlap of the bounding box with the GT (Ground Truth). To handle this problem, bounding box coordinate regression is carried out to increase IoU (Intersection over Union). The input of this regression is the spatial sub-area of a plate image extracted from the previous bounding box results. The output from the regression is four shift values ($W_{x1}, W_{x2}, W_{y1}, W_{y2}$) which are used to improve the bounding box coordinates. The 4 values are designated for ($\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{y_1}, \boldsymbol{y_2}$).

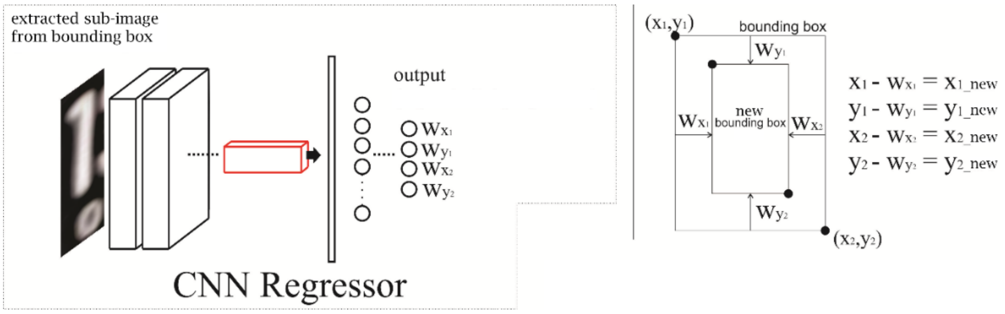

extracted sub-image from bounding box
output
W_{x_1}
W_{y_1}
W_{x_2}
W_{y_2}
CNN Regressor
(x_1,y_1)
bounding box
W_{y_1}
W_{x_1}
new bounding box
W_{x_2}
W_{y_2}
(x_2,y_2)
$x_1 - W_{x_1} = x_{1\_new}$
$y_1 - W_{y_1} = y_{1\_new}$
$x_2 - W_{x_2} = x_{2\_new}$
$y_2 - W_{y_2} = y_{2\_new}$

Fig. 4. Illustration of predicting 4 shift values and bounding box improvements

Fig. 4 shows the bounding box improvement mechanism utilizing regression results from CNN. Initially, a regressor in the form of a CNN is used to predict the coordinate shift values of a bounding box based on the 25×94 plate sub-image within it. These displacement values are then used to update the coordinate values. (x_1, x_2, y_1, y_2) of the bounding box with a subtraction operation. A value in a coordinate will increase or decrease depending on the predicted shift, whether the shift is positive or negative. Improvements to the bounding boxes on a plate are carried out iteratively based on the number on the plate.

Table 2. CNN Classifier Configuration

Layer	**Depth**	**Size**	**Size of Filter**	**Stride**	**Activation**
Input	1	25×94×1	-	-	-
Convolution	64	23×92×64	3×3	1	ReLU
Convolution	64	21×90×64	3×3	1	ReLU
Max pooling	64	10×45×64	2×2	2	-
Convolution	128	8×43×128	3×3	1	ReLU
Convolution	128	6×41×128	3×3	1	ReLU
Max pooling	128	3×20×128	2×2	2	-
Flatten	-	7680	-	-	-
Dense	-	100	-	-	ReLU
Output	-	2	-	-	Softmax

Table 3. CNN Regressor Configuration

Layer	**Depth**	**Size**	**Size of Filter**	**Stride**	**Activation**
Input	1	25×94×1	-	-	-
Convolution	32	25×94×32	3×3	1	ReLU
Convolution	32	25×94×32	3×3	1	ReLU
Max pooling	32	12×47×32	2×2	2	-
Convolution	64	12×47×64	3×3	1	ReLU
Convolution	64	12×47×64	3×3	1	ReLU
Max pooling	64	6×23×64	2×2	2	-
Convolution	128	6×23×128	3×3	1	ReLU
Convolution	128	6×23×128	3×3	1	ReLU
Convolution	128	6×23×128	3×3	1	ReLU
Max pooling	128	3×11×128	2×2	2	-
Flatten	-	8448	-	-	-
Dense	-	4096	-	-	ReLU
Dense	-	4096	-	-	ReLU
Output	-	4	-	-	-

The CNN used in the bounding box regressor has a deeper architecture than the character detector CNN (classifier), where 7 convolution layers and 3 max-pooling layers are used. The convolution layers have 32, 64, and 128 filters. The convolution operation uses zero padding on the input image for each convolution layer to maintain the size of the feature map. The reduction in feature map size is only achieved by max pooling. In FCL (Fully Connected Layer), there is one flattening layer, two dense layers, and one output layer. The activation function is ReLU for convolution and dense layers. Meanwhile, for the output layer, no activation function is used, so the input and output relationship at this layer is $f(x) = x$. The CNN regressor size and hyperparameter configuration are shown in Table 2.

The proposed CNN regressor model learning uses data of the extracted character sub-images from the detector model and the original training data. The regression target in learning is the difference between the bounding box coordinate and the closest ground truth coordinate. Hence, by setting this target, the model is expected to be able to produce values for the difference between the bounding box coordinates and the ground truth coordinates in the test data. In the previous explanation, these difference values are called shift values. This regression target is visualized in Fig. 5.

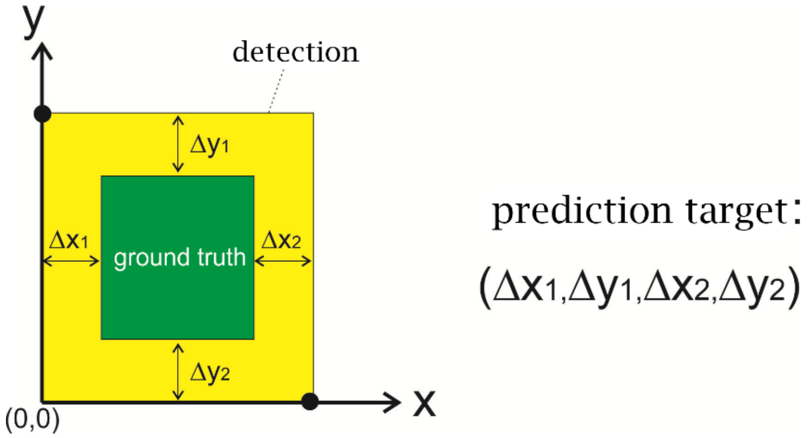


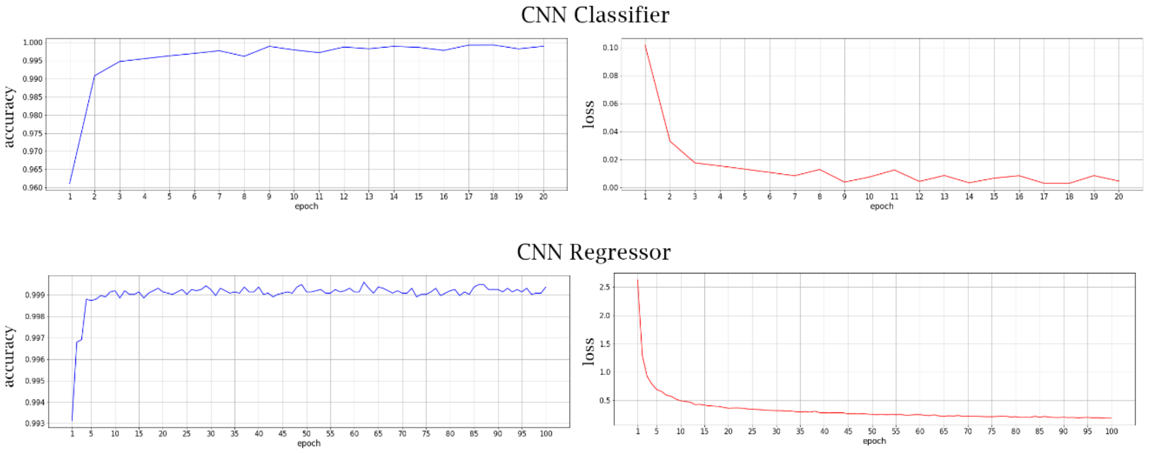
Fig. 5. Illustration of prediction target of CNN regressor training

2.4 Data Augmentation

To achieve effective character detection, the CNN classifier model should be trained with extensive data, so the initial set of 3,503 character images is augmented by extracting new images within a 25×94 pixel window, positioned within 2 pixels of the ground truth (GT) on the x-axis, yielding 17,439 new character images. Additionally, for classification, non-character images are extracted by defining areas at least 10 pixels away from any GT on the x-axis with an IoU ≤ 3, using a 25×94 pixel window and a stride of 3 pixels, resulting in 20,044 non-character images. The training data for the bounding box regressor model comes from detection results, consisting of 25×94 images. To increase the dataset, image brightness is varied for augmentation, resulting in 17,486 images.

3. Model Training

In the training of the CNN classifier, binary cross entropy was used as the loss function, and Adam was the optimizer with a learning rate of 0.001. The training was conducted for 20 epochs with a batch size of 32. After 20 epochs of training, the CNN classifier achieved an accuracy of 0.9989 and a loss of 0.0044. For the CNN regressor training, the MAE (Mean Absolute Error) loss function was used. Adam was also used as the optimizer with a learning rate of 0.001. The training was conducted over 100 epochs with a batch size of 64. The CNN regressor training results showed an accuracy of 0.9992 and a loss of 0.1878.


CNN Classifier

CNN Regressor

Fig. 6. Training results

4. Result and Discussion

4.1 Testing Results on Detector

The character detector model was tested on test data of 280 Indonesian vehicle plates with a total of 2026 characters. In this test, the results of character detection without involving a bounding box regressor are presented. Of the 2026 characters, the character detector model produced 2030 detections. The obtained average IoU was 0.4811. The average computation of character segmentation on a plate image was 0.0658 seconds. A sample of character segmentation results using this character detector model is visualized in Fig.7, the red box is GT and the green box is the detection result.

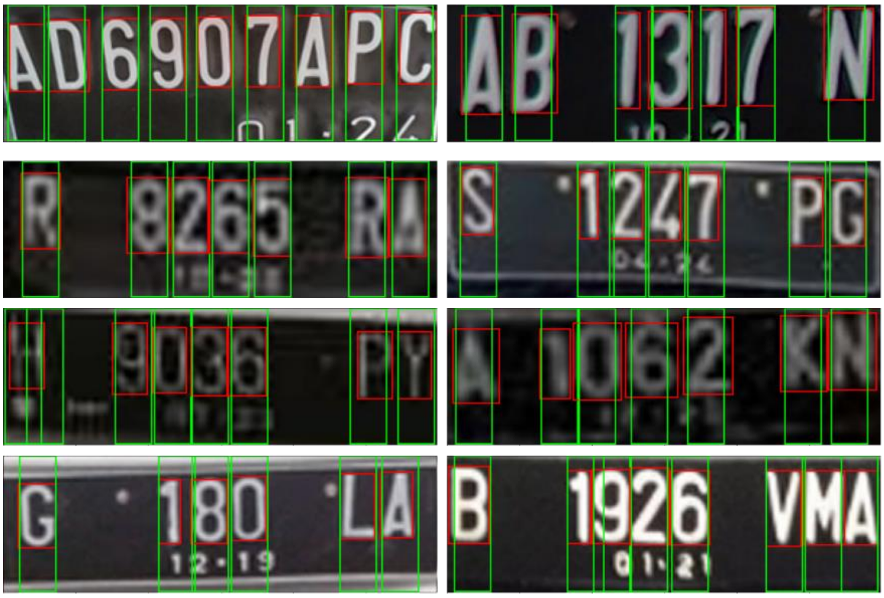


Fig. 7. Sample of character detector result

4.2 Testing Results with Bounding Box Regressor

After testing the character detector model separately, combining it with a bounding box regressor was carried out and tested. The total bounding box regressed is 2030, which is the bounding box from the character detector test. The results of both tests produced an average IoU of 0.8980. The computation to perform character segmentation for a plate image is 0.0827 seconds on average.

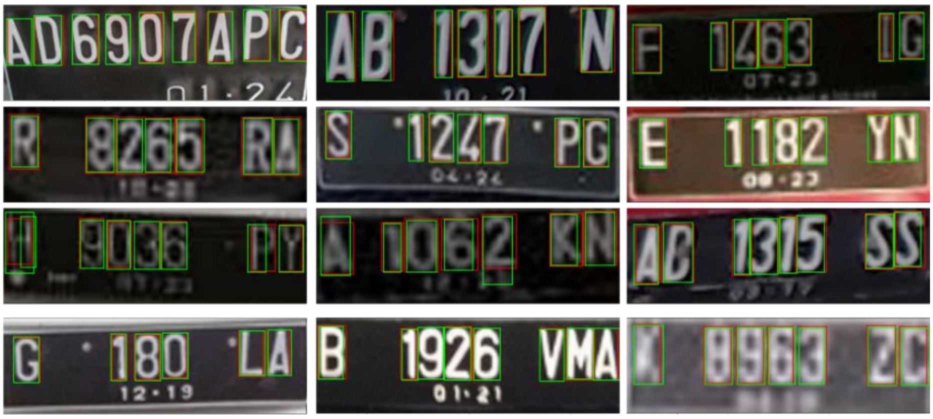


Fig. 8. Sample of bounding box regressor result

4.3 Performance Comparison

The first comparison is a comparison of the average IoU results produced by the two testing stages that have been carried out along with average computation for a plate image. This comparison is shown in Table 4. In this table, CD is the character detector and BBR is the bounding box regressor. An increase in the average IoU of 0.4169 occurred when BBR was implemented in the system. This can be proven visually by looking at the IoU distribution from the test in Fig. 9. As seen in the figure, the IoU distribution on CD-BBR is centered on higher values.

Table 4. IoU and Computation Comparison

Metrics	Model	
	CD	**CD-BBR**
Avg. IoU	0.4811	0.8980
Avg. Computation/img	0.0658 s	0.0827 s

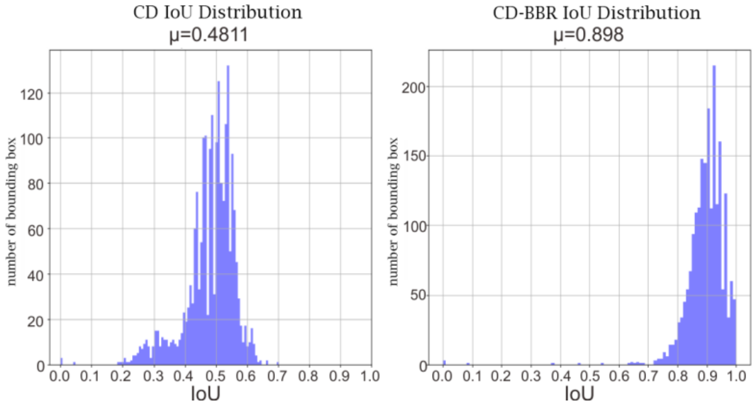


Fig. 9. IoU distribution

TP (True Positive), FP (False Positive), and FN (False Negative) in the character segmentation results are determined by the specified IoU threshold. We use 0.8 as the IoU threshold to calculate the precision and recall of the model. Not only our BBR backbone, we also present a test on VGG16 as a bounding box regressor, the comparison of the two is presented in Table 5.

Table 5. Comparison of Performance on BBR backbone (IoU threshold of 0.8)

Model	TP	FP	FN	Avg. IoU	*Precision*	*Recall*	Avg. s/img
CD-VGG16[10]	1856	174	170	0.8818	0.9142	0.9160	0.1154 s
CD-BBR	1943	87	83	0.8980	0.9571	0.9590	0.0827 s

5. Conclusion

From 280 license plate images containing 2026 characters, CD produced character detections with an average IoU of 0.4811. Implementing the BBR increased the average IoU to 0.8980, improving the IoU of CD alone. By comparing bounding box backbone performance, we can see that CD-BBR outperformed CD-VGG16. This is motivated by a statement in [11] that better character segmentation tends to improve recognition. This study's CNN regressor model successfully predicted coordinate shifts, making bounding boxes more adaptive and increasing overlap with characters, thereby reducing the inclusion of non-character areas when used in the recognition phase.

References

[1] Badan Pusat Statistik, "Jumlah Kecelakaan, Korban Mati, Luka Berat, Luka Ringan, dan Kerugian Materi, 2022." Accessed: May 15, 2024. [Online]. Available: https://www.bps.go.id/id/statistics-table/2/NTEzIzI=/jumlah-kecelakaan--korban-mati--luka-berat--luka-ringan--dan-kerugian-materi.html

[2] M. M. Rezapour Mashhadi, P. Saha, and K. Ksaibati, "Impact of traffic Enforcement on Traffic Safety," International Journal of Police Science & Management, vol. 19, no. 4, pp. 238–246, 2017, doi: 10.1177/1461355717730836.

[3] Hendry and R. C. Chen, "A New Method for License Plate Character Detection and Recognition," in ACM International Conference Proceeding Series 2018, pp. 204–208, doi: 10.1145/3301551.3301592.

[4] C. H. Lin and Y. Li, "A License Plate Recognition System for Severe Tilt Angles Using Mask R-CNN," *in International Conference on Advanced Mechatronic Systems, ICAMechS*, vol. 2019-Augus, pp. 229–234, doi: 10.1109/ICAMechS.2019.8861691.

[5] A. Abdussalam, S. Sun, M. Fu, H. Sun, and I. Khan, "License Plate Segmentation Method Using Deep Learning Techniques Amr," Springer Singapore, 2018. [Online]. Available: https://doi.org/10.1007/978-981-13-1733-0_8

[6] I. W. Notonogoro, Jondri, and A. Arifianto, "Indonesian License Plate Recognition Using Convolutional Neural Network," in 2018 6th International Conference on Information and Communication Technology, ICoICT, doi: 10.1109/ICoICT.2018.8528761.

[7] R. Laroca *et al.*, "A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector," Proceedings of the International Joint Conference on Neural Networks, vol. 2018-July, 2018, doi: 10.48550/arXiv.1802.09567.

[8] S. Abdullah, M. Mahedi Hasan, and S. Muhammad Saiful Islam, "YOLO-Based Three-Stage Network for Bangla License Plate Recognition in Dhaka Metropolitan City," in 2018 International Conference on Bangla Speech and Language Processing, ICBSLP 2018, pp. 1–6, doi: 10.1109/ICBSLP.2018.8554668.

[9] A. T. Musaddid, A. Bejo, and R. Hidayat, "Improvement of Character Segmentation for Indonesian License Plate Recognition Algorithm using CNN," in 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI) 2020, pp. 279–283, doi: 10.1109/ISRITI48646.2019.9034614.

[10] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, pp. 1–14, doi: 10.48550/arXiv.1409.1556.

[11] G. R. Gonçalves, S. P. G. da Silva, D. Menotti, and W. R. Schwartz, "Benchmark for License Plate Character Segmentation," J Electron Imaging, vol. 25, no. 5, p. 053034, Oct. 2016, doi: 10.48550/arXiv.1607.02937.

JRAEE

Journal of Robotics, Automation, and Electronics Engineering
Vol. 2, No. 1, 2024, pp. 20-29
e-ISSN 3025-3780
https://journal.uny.ac.id/publications/jraee

Design and Implementation of a Student Counting and Monitoring System in a Laboratory Using Human Tracking Method with OpenCV and TensorFlow

Nancy Febriani Taek[a,1], Arya Sony[a,2,*]

[a] Universitas Negeri Yogyakarta
[1] nancyfebriani.2019@student.uny.ac.id; [2] arya.sony@uny.ac.id
[*] Corresponding Author

ARTICLE INFO

Article History
Received 10 Dec. 2023
Revised 28 Jan. 2024
Accepted 5 Mar. 2024

Keywords
Laboratory;
Object Detection;
Human Tracking;
Machine Learning.

ABSTRACT

Laboratories serve as crucial facilities supporting practical activities, with a recommended maximum of 20 students, necessitating periodic monitoring to count the dynamic number of students within. The system utilizes the COCO dataset labeled ”person,” involving an approach with entry and exit preference lines, ID identification implementation, and object detection models YOLO v3 Tiny and Faster R-CNN ResNet50. The main system components, Raspberry Pi 3 Model B+, Raspberry Pi Camera 5 MP (f/1.3), and Raspberry Pi 7-inch Touch Display, are integrated for processing, real-time video recording, and image display functions. Test and evaluation results reveal that YOLO v3 Tiny achieves an 88.24% accuracy for entry counting and 75% for entry-exit counting, with an average processing rate of 4.89 FPS, while Faster R-CNN ResNet50 demonstrates lower accuracy, reaching 70.59% and 45.83%, with an average processing rate of 0.58 FPS.

Laboratorium sebagai fasilitas penting pendukung kegiatan praktik, dengan jumlah siswa maksimum yang direkomendasikan adalah 20 orang, memerlukan pemantauan berkala guna menghitung dinamika jumlah siswa yang berada di dalamnya. Program sistem memanfaatkan dataset COCO yang berlabel ”person” juga pendekatan yang melibatkan garis preferensi masuk dan keluar, penerapan identifikasi ID, dan model deteksi objek YOLO v3 Tiny dan Faster R-CNN ResNet50. Komponen utama sistem, yakni Raspberry Pi 3 Model B+, Raspberry Pi Camera 5MP (f/1.3), dan Raspberry Pi 7-inch Touch Display, saling terintegrasi untuk fungsi pemrosesan, perekaman video real-time, dan penampilan citra. Hasil pengujian dan evaluasi menunjukkan YOLO v3 Tiny berhasil mencapai akurasi 88.24% untuk penghitungan masuk dan 75% untuk penghitungan masuk-keluar, dengan rata-rata pemrosesan 4.89 FPS, sementara Faster R-CNN ResNet50 menunjukkan akurasi yang lebih rendah yaitu 70.59% dan 45.83%, dengan rata-rata pemrosesan 0.58 FPS.

1. Introduction

The laboratory serves as a critical facility supporting practical activities, necessitating attention to the number of students present during its use. Generally, the recommended maximum number of students is 20. This regulation is reinforced by the Decree of the Minister of National Education of the Republic of Indonesia Number 234/U/2000 on Guidelines for the Establishment of Higher Education Institutions, the Higher Education Law Number 12/2012, and Government Regulation Number 4/2014 on the Implementation of Higher Education, which state that the ideal ratio of lecturers to students is 1:20 for Exact Sciences and 1:30 for Social Sciences.

Although considered an important task, tracking the number of students is a complex challenge that requires a solution such as developing a tool for automatically monitoring the number of students in the laboratory by applying digital image processing techniques. One of the challenges of image processing in the context of image applications is computer vision, which involves creating digital images from original images according to human perception [1]. This technology can identify objects and classify them as human or non-human [2]. Subsequently, it can count the number of students in a video frame. However, several implementations still face challenges, particularly related to modeling the multi-object tracking process, as evidenced by research on people counting systems for attendance monitoring [3].

The application of human tracking technology, normally developed by utilizing reference lines on an image spatial area, can effectively monitor the movement and dynamics of students in the laboratory [4]. The object detection models used throughout the detection process and for counting students in the laboratory are YOLO v3 Tiny and Faster R-CNN ResNet50. These models are implemented using OpenCV and TensorFlow, with Raspberry Pi 3 Model B+ serving as the primary processor in the system.

2. Methods

2.1 Raspberry Pi 3 Model B+

The Raspberry Pi is a credit card-sized Single Board Computer (SBC) equipped with an integrated circuit chip that functions as a system-on-chip (SoC), providing general processing capabilities, graphics rendering, and input/output. The Raspberry Pi 3 Model B+ runs on Raspberry Pi OS and serves as the main processor in the system. The appearance of the Raspberry Pi 3 Model B+ is shown in Fig. 1.

Fig. 1. Raspberry Pi 3 model B+

2.2 Mechanical Design of the Device

The main components of the system device include the Raspberry Pi 3 Model B+, Raspberry Pi Camera 5 MP (f/1.3), and Raspberry Pi 7-inch Touch Display, along with supporting components such as the power supply and other voltage sources, designed to be housed in a box container. The dimensions of this device enclosure measure 256 x 180 x 150 mm. The mechanical design of the device can be viewed in Fig. 2.

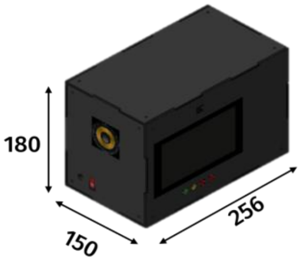


Fig. 2. Mechanical design of the device

2.3 Electronic System Design of the Device

The electronic design layout is formulated to determine the placement of each required component. The Raspberry Pi 3 Model B+ serves as the main processor controlling the system. It can be observed that the Raspberry Pi Camera is connected via the CSI connector port, and the Raspberry Pi Touch Display is connected via the HDMI port. Power is supplied through the Raspberry Pi adapter via the micro-USB power port. The diagram illustrating the Electronic System Design of the Device can be seen in Fig. 3.

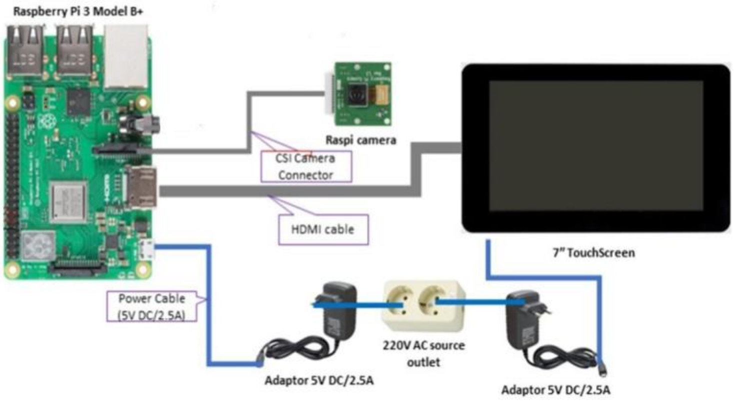
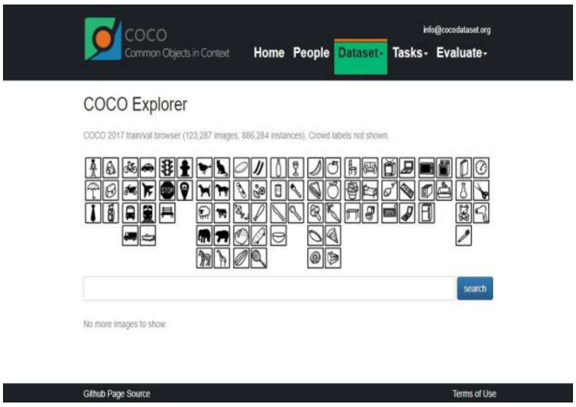


Fig. 3. Electronic system design of the device

2.4 COCO Dataset

The COCO (Common Objects in Context) dataset provides a large collection of data covering various labeled categories, including humans, vehicles, animals, and other domestic objects, as illustrated in Fig. 4. This research specifically focuses on object detection with the label "person". Both object detection models used in this study have been trained with this dataset. In YOLO v3 Tiny, the class ID 0 corresponds to "person" in the COCO names, while in Faster R-CNN ResNet50, the class ID 1 corresponds to "person" in the COCO dataset [5].



COCO Explorer

COCO 2017 train/val browser (123,287 images, 886,284 instances). Crowd labels not shown.

No more images to show.

Fig. 4. COCO dataset collection

2.5 Tracking Method

Vertical preference lines are utilized as indicators for the entry and exit paths of detected students within the room, captured by the camera. The system employs two vertical preference lines in the image display: a blue entry line and a red exit line. The monitoring process begins by marking individuals detected by the camera with a center point, followed by tracking their movement direction [6]. When a detected person moves toward and crosses the entry preference line, the system logs it as an entry event; conversely, moving toward and crossing the exit preference line is logged as an exit event [7]. Information regarding the count of entry and exit events generated by the system is displayed on the image. The design layout of the preference lines system is depicted in Fig. 5.

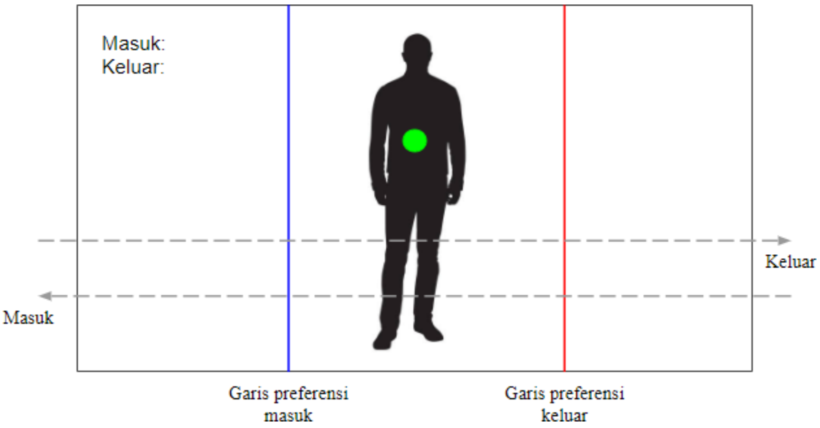


Fig. 5. Design of preference line display

2.6 Data Acquisition and Evaluation

The image preference line tracking method approach is implemented in both YOLO v3 Tiny and Faster R-CNN ResNet50 detection models [8]. Documented data components include a timestamp indicating the current measurement time, Person ID representing a unique identification for each detected student, Entry and exit counts recorded by the system compared to actual counts, and Confidence value indicating the model's prediction certainty that an object in the frame is a human.

Evaluation of the test results also utilizes error percentage (PE) and accuracy metrics, as follows:

$$PE = \frac{|x - x^{\wedge}|}{x} \times 100\% \quad (1)$$

$$Accuracy = 100\% - PE \quad (2)$$

Where,

x = Number of people tested
$x^{\wedge}$ = Number of people recorded

3. Result and Discussion

3.1 Implementation and Hardware Testing

The device enclosure is constructed from 2 mm thick acrylic using cutting techniques, and assembled by connecting each corner with nuts and bolts. Following this, the main system components and supporting elements are mounted, and wiring installation is performed for component interconnection. The device enclosure is also equipped with a 1360 mm tall tripod. The implementation of the mechanical and electronic design results is shown in Fig.6.

ISSN 3025-3780 (online)
ISSN 3025-4590 (printed)

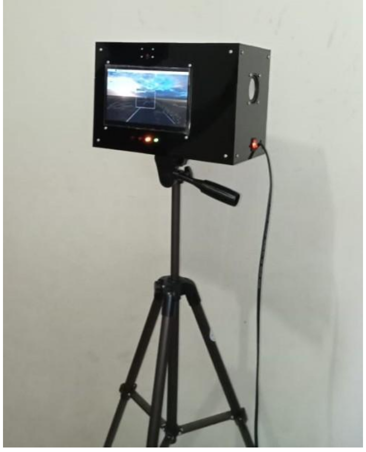


Fig. 6. Implementation of device design

The testing was conducted to measure the system's performance, starting with the evaluation of the hardware system performance. The results of the hardware performance testing are presented in Table 1 below.

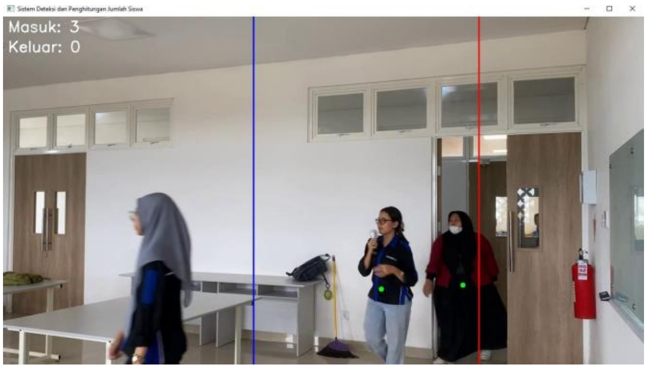
Table 1. Hardware System Testing

Main Component	Status
Raspberry Pi 3 Model B+	Working
Raspberry Pi Camera 5 MP (f/1.3)	Working
Raspberry Pi 7" Touch Display	Working

The table above shows the test results of the main hardware components used in the development of this student count detection system in the laboratory, namely the Raspberry Pi 3 Model B+, Raspberry Pi Camera 5MP (f/1.3), and Raspberry Pi 7" Touch Display, all of which operate effectively.

3.2 Entry Count Testing

The entry count testing process involves the participation of 17 individuals entering the room regularly, as shown in Fig. 7.



Sistem Deteksi dan Penghitungan Jumlah Siswa

Masuk: 3
Keluar: 0

Fig. 7. Entry count testing

The results of this testing are presented in Table 2 for entry count testing using YOLO v3 Tiny, and in Table 2 for entry count testing using Faster R-CNN ResNet50, as follows.

Table 2. Entry Count Testing YOLO v3 Tiny

Timestamp (02/12/2023)	Person ID	Expected Entry	Expected Exit	Recorded Entry	Recorded Exit	Confidence Score
12:03:12	person01	1	0	1	0	91.66%
12:03:15	person02	2	0	2	0	90.18%
12:03:27	person03	3	0	4	0	92.58%
12:03:42	person04	4	0	5	0	91.29%
12:03:51	person05	5	0	6	0	98.61%
12:03:54	person01	6	0	7	0	97.73%
12:04:05	person02	7	0	8	0	88.75%
12:04:15	person03	8	0	9	0	96.80%
12:04:20	person04	9	0	10	0	92.69%
12:04:35	person01	10	0	11	0	87.01%
12:04:40	person02	11	0	12	0	95.01%
12:04:54	person03	12	0	13	0	95.63%
12:05:00	person04	13	0	14	0	91.94%
12:05:40	person01	14	0	15	0	99.09%
12:06:00	person02	15	0	17	0	98.33%
12:06:05	person03	16	0	18	0	97.26%
12:06:28	person04	17	0	19	0	94.94%

The testing results using YOLO v3 Tiny indicate that the ID assignment can correctly reference up to 5 individuals, but there were 2 instances where different individuals were counted redundantly, specifically the 3rd and 15th persons. The average confidence value displayed during testing reached 94.09%.

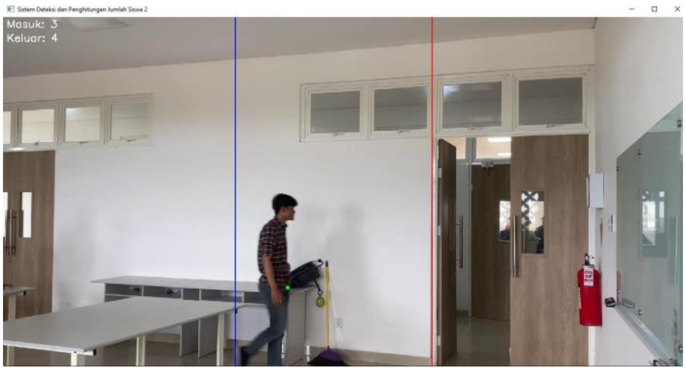
Table 3. Entry Count Testing Faster R-CNN ResNet50

Timestamp (02/12/2023)	Person ID	Expected Entry	Expected Exit	Recorded Entry	Recorded Exit	Confidence Score
02:13:21	person01	1	0	1	0	91.66%
02:13:33	person02	2	0	2	0	87.98%
02:13:37	person03	3	0	4	0	92.58%
02:13:51	person01	4	0	5	1	91.29%
02:14:00	person02	5	0	6	0	98.61%
02:14:23	person01	6	0	7	1	95.50%
02:14:26	person02	7	0	8	1	88.66%
02:14:45	person01	8	0	9	0	85.69%
02:14:51	person02	9	0	10	0	95.01%
02:15:12	person03	10	0	11	1	91.94%
02:15:17	person01	11	0	12	0	97.59%
02:15:25	person01	12	0	12	0	99.45%
02:15:59	person02	13	0	13	1	95.72%
02:16:13	person01	14	0	14	1	96.93%
02:16:14	person02	15	0	15	0	98.33%
02:16:19	person01	16	0	16	0	96.77%
02:16:40	person01	17	0	16	0	94.80%

The testing results using Faster R-CNN ResNet50 show that ID assignment can correctly reference up to 3 individuals, with one individual counted redundantly, specifically the 3rd person [9]. Additionally, 2 individuals were not counted, specifically the 12th and 17th persons, and there were errors in counting individuals exiting. The average confidence value displayed during testing reached 94.03%.

3.3 Entry-Exit Count Testing

The entry-exit count testing process involves the participation of 17 individuals entering and leaving the room regularly, as shown in Fig. 8.



Masuk: 3
Keluar: 4

Fig. 8. Entry-exit count testing

The results of this testing are presented in Table 4 for entry-exit count testing using YOLO v3 Tiny, and in Table 5 for entry-exit count testing using Faster R-CNN ResNet50, as follows.

Table 4. Entry-Exit Count Testing YOLO v3 Tiny

Timestamp (02/12/2023)	Person ID	Expected Entry	Expected Exit	Recorded Entry	Recorded Exit	Confidence Score
16:40:51	person01	1	0	1	1	97.55%
16:41:18	person02	1	1	1	2	95.80%
16:41:23	person01	2	1	2	2	97.57%
16:41:52	person02	2	2	2	3	95.54%
16:42:34	person01	3	2	3	3	94.02%
16:44:33	person01	3	3	4	3	96.56%
16:46:57	person01	4	3	5	3	97.51%
16:47:02	person02	5	3	6	4	97.00%
16:47:07	person01	5	4	6	5	97.19%
16:48:00	person02	5	5	7	6	99.18%
16:48:01	person01	6	5	7	6	97.72%
16:48:05	person02	7	5	8	6	95.71%
16:48:11	person03	8	5	9	7	97.45%
16:49:45	person01	8	6	9	8	91.39%
16:49:46	person02	8	7	9	9	97.38%
16:49:50	person03	8	8	10	9	90.76%
16:49:51	person04	8	9	10	10	92.22%
16:50:55	person01	9	9	10	11	98.66%
16:50:59	person01	10	9	11	11	95.30%
16:51:47	person01	10	10	12	12	95.12%
16:51:52	person02	10	11	12	12	96.64%
16:53:38	person01	11	11	15	13	95.53%
16:53:40	person02	11	11	16	13	90.60%
16:53:44	person03	13	11	17	13	97.96%

The testing results using YOLO v3 Tiny indicate that ID assignment can correctly reference up to 4 individuals. The expected counts for entries and exits were 13 and 11 respectively, but the system counted 17 entries and 13 exits. The average confidence value displayed during testing reached 95.85%.

Table 5. Entry-Exit Count Testing Faster R-CNN ResNet50

Timestamp (02/12/2023)	Person ID	Expected Entry	Expected Exit	Recorded Entry	Recorded Exit	Confidence Score
00:47:50	person01	1	0	1	1	97.55%
00:48:17	person01	1	1	1	2	95.80%
00:48:23	person01	2	1	2	2	97.57%
00:48:52	person01	2	2	2	3	95.54%
00:49:28	person01	3	2	3	3	94.02%
00:49:57	person01	3	3	3	4	98.41%
00:51:19	person01	4	3	4	4	98.66%
00:51:23	person01	5	3	6	5	96.51%
00:52:30	person01	5	4	6	5	94.61%
00:53:51	person01	5	5	7	6	95.90%
00:53:54	person01	6	5	8	6	96.77%
00:53:57	person01	7	5	8	7	97.00%
00:54:55	person01	8	5	9	7	99.18%
00:54:56	person01	8	6	9	8	98.93%
00:54:58	person02	8	7	9	9	97.72%
00:55:18	person01	8	8	11	10	96.44%
00:55:26	person01	8	9	11	11	98.24%
00:56:42	person01	9	9	13	12	97.55%
00:57:53	person01	10	9	14	14	99.09%
00:57:54	person01	10	10	16	15	94.47%
00:58:39	person01	10	11	17	16	95.12%
00:58:45	person01	11	11	18	17	96.23%
01:00:18	person02	12	11	19	17	93.24%
01:00:34	person01	13	11	20	17	97.96%

The testing results using YOLO v3 Tiny indicate that ID assignment was not accurately maintained. The expected counts for entries and exits were 13 and 11 respectively, but the system counted 20 entries and 17 exits. The average confidence value displayed during testing reached 96.77%.

3.4 System Performance

The system performance evaluation in counting the number of entries, along with the results obtained from each model, is also detailed in Table 6 as follows.

Table 6. System Performance

Timestamp (02/12/2023)	Person ID	Expected Entry	Expected Exit	Recorded Entry	Recorded Exit
Entry	YOLO v3 Tiny	17+0	19+0	11.76%	88.24%
	Faster R-CNN ResNet50	17+0	16+6	29.41%	70.59%
Entry-Exit	YOLO v3 Tiny	13+11	17+13	25%	75%
	Faster R-CNN ResNet50	13+11	20+17	54.17%	45.83%

Based on the table above, the performance evaluation results of the system in counting the number of student entries and exits show that the YOLO v3 Tiny model has higher accuracy compared to the Faster R-CNN ResNet50 model in both counting schemes [10].

During processing, YOLO v3 Tiny demonstrated better performance with an average FPS measured during entry-counting testing at 4.89 FPS and entry-exit counting at 4.17 FPS. In contrast, Faster R-CNN ResNet50 showed an average FPS measured during entry-counting testing at 0.58 FPS and entry-exit counting at 0.51 FPS.

4. Conclusion

The student monitoring system in the laboratory using OpenCV and TensorFlow-based human tracking method was designed and constructed with the main components: Raspberry Pi 3 Model B+, Raspberry Pi Camera 5MP (f/1.3), and Raspberry Pi 7-inch Touch Display, integrated for processing, real-time video recording, and image display functions. These main components, along with supporting elements, were housed in an acrylic-constructed device enclosure with a thickness of 2mm. The device enclosure dimensions are 256x180x150 mm. The device is also equipped with a 1360 mm tall tripod for support. Connections between the main system components involve linking each port on the Raspberry Pi 3 Model B+ to the Raspberry Pi Camera 5 MP (f/1.3) via the CSI camera port and the Raspberry Pi 7-inch Touch Display via the DSI display port.

Based on the testing and evaluation results, in the entry count scheme, YOLO v3 Tiny successfully detected 19 entry cases out of a total of 17 actual cases, resulting in an error rate of 11.76% and an accuracy of 88.24%. On the other hand, Faster R-CNN ResNet50 detected 16 entry cases out of 17 actual cases with misidentification leading to falsely counted exits, resulting in an error rate of 29.41% and an accuracy of 70.59%. In the entry-exit count scheme, YOLO v3 Tiny detected 30 entry-exit cases out of 24 actual cases, resulting in an error rate of 25% and an accuracy of 75%. Conversely, Faster R-CNN ResNet50 showed less satisfactory results by detecting 37 entry-exit cases out of 24 actual cases, with an error rate of 54.17% and an accuracy of only 45.83%. In both counting schemes, YOLO v3 Tiny demonstrated higher accuracy compared to Faster R-CNN ResNet50. The entry count scheme showed relatively higher accuracy compared to the entry-exit count scheme for both models. Additionally, during processing, YOLO v3 Tiny exhibited better performance with an average FPS of 4.89 FPS, whereas Faster R-CNN ResNet50 only achieved 0.58 FPS. Therefore, YOLO v3 Tiny was selected and implemented for the student monitoring system in the laboratory environment.

In the future, we will Improve the system performance by optimizing hardware resource utilization, especially on devices with limitations such as the Raspberry Pi 3 Model B+, by migrating to devices with higher specifications, such as the latest Raspberry Pi models or other types of processors. The use of more advanced hardware can enhance performance and processing capabilities, particularly in terms of processing speed like increasing FPS, enabling the system to operate more efficiently and responsively.

References

[1] F. A. Hariz, I. N. Yulita, and I. Suryana, “Human Activity Recognition Berdasarkan Tangkapan Webcam Menggunakan Metode Convolutional Neural Network (CNN) Dengan Arsitektur MobileNet,” JITSI: Jurnal Ilmiah Teknologi Sistem Informasi, vol. 3, no. 4, pp. 103-115, 2022, doi: 10.30630/jitsi.3.4.97.

[2] M. I. H. Azhar, F. H. K. Zaman, N. M. Tahir, and H. Hashim, “People Tracking System Using DeepSORT,” In 2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 2020, pp. 137-141, doi: 10.1109/ICCSCE50387.2020.9204956.

[3] M. C. Le, M. H. Le, and M. T. Duong, “Vision-Based People Counting for Attendance Monitoring System,” In 2020 5th International Conference on Green Technology and Sustainable Development (GTSD) IEEE, Nov. 2020, pp. 349-352, doi: 10.1109/GTSD50082.2020.9303117.

[4] F. Indaryanto, A. Nugroho, and A. F. Suni, “Aplikasi Penghitung Jarak Dan Jumlah Orang Berbasis YOLO Sebagai Protokol Kesehatan COVID-19,” Edu Komputika Journal, vol. 8, no. 1, pp. 31-38, 2021, doi: 10.15294/edukomputika.v8i1.47837.

[5] J. Luo, Y. Wang, and Y. Wang, “Real-Time Pedestrian Detection Method Based on Improved YOLOV3,” In Journal of Physics: Conference Series, vol. 1453, no. 1, p. 012149, 2020, doi; 10.1088/1742-6596/1453/1/012149.

[6] G. Gallo, F. Di Rienzo, F. Garzelli, P. Ducange, and C. Vallati, “A Smart System for Personal Pro- Tective Equipment Detection in Industrial Environments Based on Deep Learning at the Edge,” IEEE Access, 2022, pp. 110862-110878, doi: 10.1109/ACCESS.2022.3215148.

[7] S. A. Velastin, R. Ferna´ndez, J. E. Espinosa, & A. Bay, “Detecting, Tracking and Counting People Getting On/off a Metropolitan Train Using a Standard Video Camera,” Sensors, vol. 20, no. 21, p. 6251, 2020, doi: 10.3390/s20216251.

[8] W. Liu, X. Du, Q. Geng, J. Li, H. Li, and L. Liu, “Metro Passenger Flow Statistics Based on YOLOV3,” In IOP Conference Series: Materials Science and Engineering, IOP Publishing, vol. 688, no. 4, p. 044025, 2019, doi: 10.1088/1757-899X/688/4/044025.

[9] I. Ahmed, M. Ahmad, A. Ahmad, & G. Jeon, “Top View Multiple People Tracking by Detection Using Deep Sort and YOLOV3 With Transfer Learning: Within 5G Infrastructure,” International Journal of Machine Learning and Cybernetics, vol. 12, 3053-3067, 2021, doi: 10.1007/s13042-020-01220-5.

[10] Y. W. Hsu, T. Y. Wang, and J. W. Perng, “Passenger Flow Counting in Buses Based on Deep Learning Using Surveillance Video,” Optik, vol. 202, p. 163675, 2020, doi: 10.1016/j.ijleo.2019.163675.

JRAEE

Journal of Robotics, Automation, and Electronics Engineering
Vol. 2, No. 1, 2024, pp. 30-38
e-ISSN 3025-3780
https://journal.uny.ac.id/publications/jraee

Rotman Lens Size Reduction by Using Same-Size Double Rectangular Defected Ground Structures (DGSs) Method or Same-Size Double Rectangular Slots Method

Rizky Hidayat Prasetyo[a,1,*], Eko Tjipto Rahardjo[b]

[a] Universitas Negeri Yogyakarta
[b] Universitas Indonesia
[1] rizkyhidayat@uny.ac.id
[*] Corresponding Author

ARTICLE INFO

Article History
Received 20 Jan. 2024
Revised 15 Mar. 2024
Accepted 5 May 2024

Keywords
Rotman Lens;
Multi-beamforming;
Array;
Antenna.

ABSTRACT

The need for dedicated communication keeps increasing. A technique to realize that is by using multibeam radiation. A Beamforming Network (BFN) is required to enable multibeam capability in array antennas. This study uses the Rotman Lens as BFN in the frequency of S-Band. The common problem with using Rotman Lens is that its conventional design size is quite large, mainly due to its transition leg ports. Transition leg ports are important to ensure the matching impedance between the lens and the array antenna ports the lens and the beam ports or the lens and the dummy port. The goal of this study is to reduce the size of Rotman lens transition legs by implementing simple and uniform size of slots or Defected Ground Structures (DGSs) methods for the ports of the Rotman Lens BFN. The method can minimize the length of the transition leg and allow the BFN to operate efficiently. The results revealed that the use of the same double-rectangular DGS technique and the same double-rectangular slots in ports can reduce the size of the Rotman lens. Compared to the conventional methods, the proposed method can reduce the size to almost 85 percent from its original size for this S-Band implementation. The other performances of the BFN, besides the size reduction, are not degraded by implementing the proposed methods.

Kebutuhan akan komunikasi khusus terus meningkat. Salah satu teknik untuk mewujudkan hal tersebut adalah dengan membuat pola radiasi banyak arah. Beamforming Network (BFN) diperlukan untuk membuat pola radiasi banyak arah pada antena array. Penelitian ini menggunakan Rotman Lens sebagai BFN pada frekuensi S-Band. Masalah umum dalam menggunakan Rotman Lens adalah ukuran desain konvensionalnya yang cukup besar, terutama karena port kaki transisinya. Port kaki transisi penting untuk memastikan kesesuaian impedansi antara lensa dengan port antena atau lensa dengan port beam atau lensa dengan port dummy. Tujuan dari penelitian ini adalah untuk mengurangi ukuran kaki transisi Rotman Lens dengan menerapkan metode yang sederhana dan ukuran seragam yaitu dengan slot atau Defected Ground Structure (DGS) untuk port kaki transisi Rotman Lens. Metode ini dapat mengurangi panjang kaki transisi dan memungkinkan BFN beroperasi secara efisien.

Hasil simulasi maupun pengukuran menunjukkan bahwa penggunaan teknik DGS persegi panjang ganda maupun slot persegi panjang ganda dengan ukuran sama pada port dapat memperkecil ukuran lensa Rotman. Dibandingkan dengan metode konvensional, metode yang diusulkan dapat memperkecil ukuran hingga hampir 85 persen dari ukuran aslinya pada implementasi di frekuensi S-Band. Kinerja lainnya, selain pengurangan ukuran, tidak terdegradasi dengan diterapkannya metode yang diusulkan.

1. Introduction

In recent decades, multi-beamforming antenna systems have attracted increasing widespread attention. They have been widely used in many fields, such as autonomous vehicles, radar, and communications. In the development, three different types of beamforming networks can be used to enable the multibeam operation. They are digital, analog, and hybrid BFN. Digital (and also hybrid) BFN is still costly. For the low-cost BFN, analog BFN is usually used.

There are several types of analog BFN. The main idea for this is to use a time or phase delay mechanism to create radiation in certain directions. By using microstrip substrate, [1] [2] used the Butler matrix concept, [3] [4] used the Blass matrix concept, [5] [6] [7] used the Nolen matrix concept as their BFN. Meanwhile [8] [9] [10] [11] using Lens concept as their BFN. Another researcher also used biconical [12], strip delay lines [13], and meta surface [14] to form multibeam characteristics.

One of the interesting BFNs is the Rotman Lens. It has wideband resonant frequency characteristics, a relatively more flexible phase differentiating design, and relatively low complexity regarding the number of ports that can be assigned. But, the Rotman Lens structure is usually quite large, it is about 6λ [15]. The large conventional Rotman lens size is mainly due to its transition leg ports. Some researchers have tried to tackle this issue. [16] used Klopfenstein's taper lines concept. [17] used single energy distribution slots with different sizes in each port. [18] used dual-crescent-dumble-like DGS structures with different sizes in each beam port. And [15] used the meta-transmission line concept.

This study would like to reduce the size of Rotman lens transition legs by implementing simple and uniform size of slots or Defected Ground Structures (DGSs) methods for all the ports of the Rotman Lens in S-band frequency. The contribution of the research is that this research successfully reduced the size of the Rotman Lens by using the same-size double-rectangular DGS technique or by using the same-size double-rectangular slots in all ports. This technique also reduces the iterations that often have to be done a lot.

The rest of the paper is as follows. Chapter 2 will explain the basics of Rotman Lens, Chapter 3 will explain the reduction method, Chapter 4 will discuss the results of the reduction, and Chapter 5 will conclude the research.

2. Basic of Rotman Lens

The structure of the Rotman lens is composed of a pair of metal plates that are aligned parallel to each other. In its early development, the space between the two metal plates was left to be filled with air, but in more recent advancements, it is filled with a certain dielectric substrate. The size of the Rotman lens depends on the permittivity of the substrate. A greater permittivity results in a smaller size for the Rotman lens.

Rotman Lens has three main parts. The first part is array ports, where the lens is connected to the array antenna. The second part is beam ports where excitation ports are connected to (when used as a transmitter) or signal is gathered (when used as a receiver). The last part is dummy ports where internal reflections are absorbed.

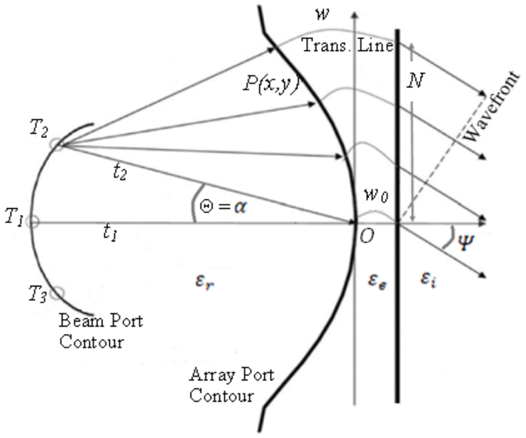


Fig. 1. Basic Geometry of Rotman Lens

In this paper, Trifocal formulations are utilized to design the Rotman Lens, like in [19]. First of all, three points should be determined from the origin point (O) with the radius of t1, t2, and t3, where t2 has the same length as t3. Then, the trifocal points T1, T2, and T3. Here we get that t2 is symmetrical with t1. After that, we have the following equations to determine the position of the array port contour.

$$T_1P\sqrt{\varepsilon_r} + W\sqrt{\varepsilon_e} = t_1\sqrt{\varepsilon_r} + W_0\sqrt{\varepsilon_e} \tag{1}$$

$$T_2P\sqrt{\varepsilon_r} + W\sqrt{\varepsilon_e} + N\sqrt{\varepsilon_i} sin \propto = t_2\sqrt{\varepsilon_r} + W_0\sqrt{\varepsilon_e} \tag{2}$$

$$T_3P\sqrt{\varepsilon_r} + W\sqrt{\varepsilon_e} - N\sqrt{\varepsilon_i} sin \propto = t_2\sqrt{\varepsilon_r} + W_0\sqrt{\varepsilon_e} \tag{3}$$

When substrate with a dielectric constant of ε_r is in the lens, substrate with the dielectric constant of ε_e is in the transmission lines, and substrate with a dielectric constant of ε_i is in the antenna; some more equations should be used

$$w = \frac{\sqrt{\varepsilon_e}}{\sqrt{\varepsilon_r}} \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{4}$$

$$x = \frac{\varepsilon_i\, N^2\, sin^2\Psi}{2\, \varepsilon_r\, (\beta cos\alpha - 1)} + \frac{(1-\beta)w}{\beta cos\alpha - 1}\sqrt{\frac{\varepsilon_e}{\varepsilon_r}} \tag{5}$$

$$y = \sqrt{\frac{\varepsilon_e}{\varepsilon_r}}\, \frac{N\, sin^2\Psi}{t_1\, sin \propto}\left(1 - \frac{w}{\beta}\sqrt{\frac{\varepsilon_e}{\varepsilon_r}}\right) \tag{6}$$

Where

$$a = 1 - \left(\frac{1-\beta}{1-\beta C}\right)^2 - \frac{\zeta^2}{\beta^2}\frac{\varepsilon_e}{\varepsilon_r} \tag{7}$$

$$b = -2 + \frac{2\zeta^2}{\beta}\frac{\varepsilon_i}{\varepsilon_r} + 2\left(\frac{1-\beta}{1-\beta C}\right) - \frac{\zeta^2 s^2 (1-\beta)}{(1-\beta C)^2}\frac{\varepsilon_i}{\varepsilon_r} \tag{8}$$

$$c = \left(-\zeta^2 + \frac{\zeta^2 S^2}{(1-\beta C)} - \frac{\zeta^2 S^4}{4(1-\beta C)}\right)\frac{\varepsilon_i}{\varepsilon_r} \tag{9}$$

$$\beta = \frac{t_2}{t_1} \tag{10}$$

$$\zeta = \frac{N\, sin\Psi}{t_1\, sin\alpha} \tag{11}$$

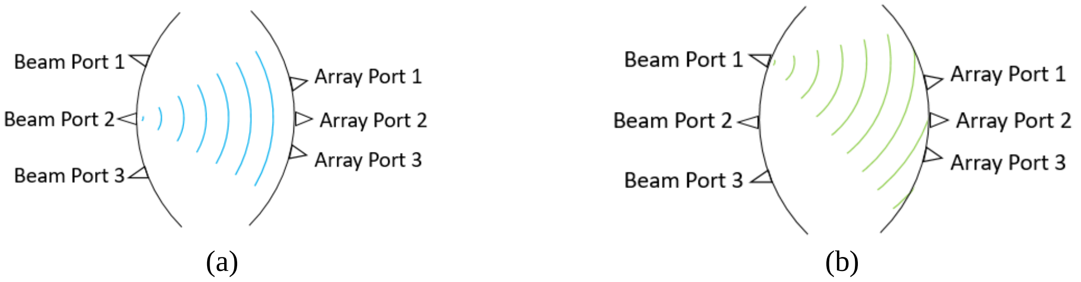


Fig. 2. (a) Phase front pattern when Beam Port 2 activated, while (b) Phase front pattern when Beam Port 1 activated

It is assumed that the array ports will receive the appropriate phases of the incoming waves based on the equations shown in Figure 2. The blue curves in Figure 2 (a) show the phase front pattern of the port with the excited center beam. It is possible to deduce that the array ports will get the same phase and will make radiation that is perpendicular to the plane.

The green curves in Fig. 2 (b) show the phase front arrangement of the port when it is excited. The phase differences in the neighboring array ports will also affect the direction of the radiation of the antenna. This is because the linearities of the successive array port's phase difference will shift the beam's direction not to be perpendicular to the plane.

When two array ports are excited at the same time, there will be two radiation patterns produced by the array antenna. This means that through a Rotman lens, multi-beam operation can be achieved.

3. Proposed Method for Reducing Rotman Lens Size

In this research, the Rotman Lens is designed to work in S-band frequency. The substrate that is being used for all ports and lens structures is FR-4 whose dielectric constant is 4.3. The number of the beam port is five (it is depicted in Fig 3 with ports 1,2,3,8 and 9; port 1 is symmetrical with port 9) to make five directional radiation patterns. While the number of the array port is six (ports 4,5,6,10,11, and 12). Array antenna elements used are also six. The space between array antenna elements is arranged to be half of the wavelength. The number of the dummy ports is two (ports 7 and 13).

There will be 3 Rotman structures shown here. They are the conventional one which employs a long transition leg, the reduced Rotman lens by using the DGS technique, and the reduced one by using the slot technique. All those 3 structures should qualify the same design parameter shown in Table 1 below.

Table 1. Rotman Lens Design Parameters

Parameter	Value
Frequency	S-Band
Number of Beam Ports	5
Number of Array Ports	6
Spacing Between Array	λ/2
α	36 °
β	0.9
ζ	1.1
Bandwidth	100 MHz
Return Loss	≤ -9,54 dB
Designated Beam Directions	±36, ±18, 0

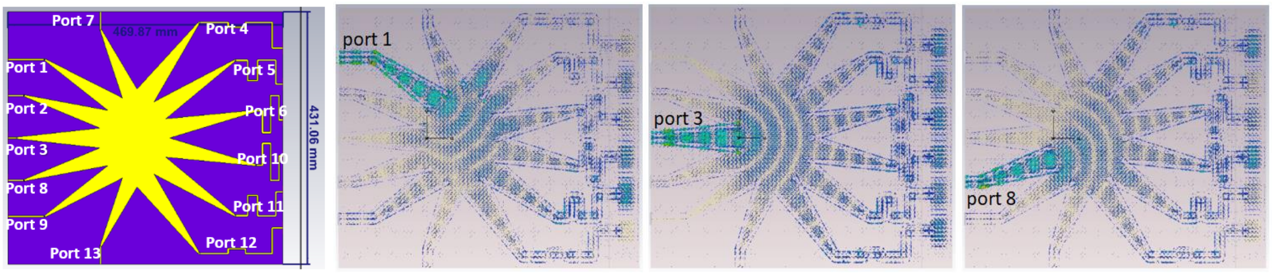


Fig. 3. Conventional Rotman Model and Its Surface Current Profile when Port 1, 3, and 8 are excited

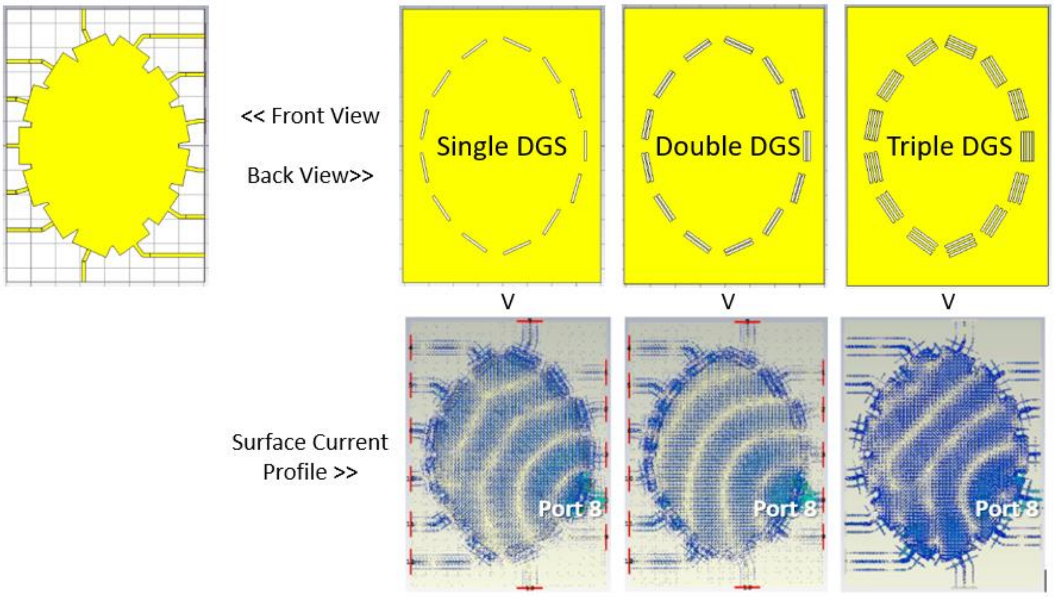


Fig. 4. Reduced Rotman Lens Models by DGS Technique and Its Surface Current Pattern when Port 8 is excited

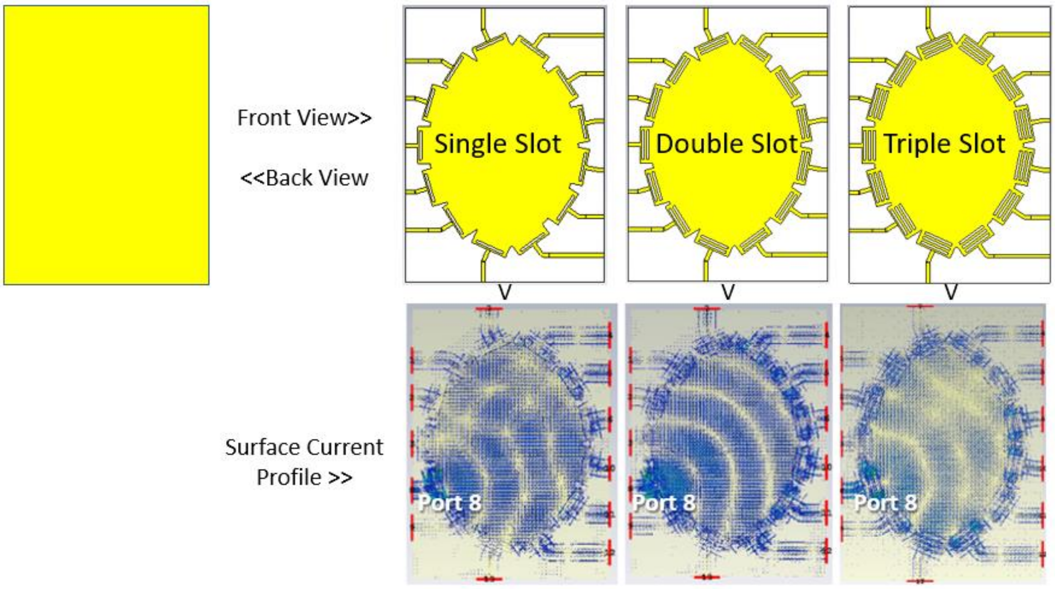


Fig. 5. Reduced Rotman Lens Models by Slot Technique and Its Surface Current Pattern when Port 8 is excited

3.1 Conventional Method

In this conventional method, the length of the leg transition is quite long due to the flare angle. The flare angle should be less than 12.5 degrees. This flare angle guarantees a smooth impedance continuity transition so that return loss can be low. The model built by this conventional model can be seen in Fig. 3. The resulting surface currents are also smooth for all time. It indicates that the impedance matching is good and impedance discontinuity is low.

3.2 Reduction by DGS Method

The long transition legs of the Rotman lens were truncated and DGS was applied to each port. Before determining what shape could reduce the Rotman lens well, investigating the characteristics of a simple DGS shape is needed. Parametric studies were conducted to examine the effect of the length and width of a single rectangular DGS on the return loss of each port. It was found that a single DGS structure will resonate in various frequencies in each port. The longer the length, the resonant frequency will shift to the higher frequency. The wider the width, the resonant frequency will shift to the higher frequency as well (except port number 2) with better return loss value. However, there had not been any single structure in all ports to resonate in S-Band.

Parametric studies were continued with dumbell structure. Dumbell structure was chosen because many researchers in antenna and or microwave devices showed good results/performances by employing this kind of structure. It investigated the effect of the length and the width of the head of the dumbell. The longer and wider of the head will result in a higher return loss.

Because there was no fit result, the study continued to employ two identical rectangular DGSs. It was hoped to create mutual resonant frequency so that the bandwidth is broad. It turned out that the idea was fruitful, the bandwidth became broad and it covered the 2.45 GHz ISM band as well.

Optimization was continued by using 3 identical DGSs. However, the results showed that the frequency was shifted to the lower frequency.

Fig. 4 shows the structure of the reduced Rotman lens by the DGS technique. From the surface current profile, it can be seen that the smooth current flow is only in double DGS whose return loss is low.

3.3 Reduction by Slot Method

The same as reduction by the DGS technique, before determining what shape could reduce the Rotman lens well, investigating the characteristics of a simple slot shape is needed. Parametric studies were also conducted to examine the effect of the length, width, and area of rectangular slots on the return loss of each port.

In the slot technique, it was also found that a single slot structure will resonate in various frequencies in one port and another. The longer the length of the slot, the resonant frequency will shift to the higher frequency. The wider the width, the resonant frequency will shift to the lower frequency. The larger area of the slot can’t be concluded. However, there was not been any single structure in all ports to resonate in the S-Band.

Because there was no fit result, the study continued to employ two identical rectangular slots. As in double DGS, it was hoped to create mutual resonant frequency so that the bandwidth is broad as well. It turned out that the idea was fruitful as well, the bandwidth became broad and it also covered the S-band including the 2.45 GHz ISM band.

Optimization was continued by using 3 identical slots. However, the results showed that the frequency was shifted to the lower frequency, as in the DGS technique.

Fig. 5 shows the structure of the reduced Rotman lens by slot technique. From the surface current profile, it can be seen that the smooth current flow is only in a double slot whose return loss is low, as in DGS. When there are 1 or 3 slots, the surface current flow is not smooth because the return loss is high. The phase front pattern is degraded when 1 or 3 slots are employed.

Table 2. Performance Comparison Among Conventional, double DGS, and Double Slot

	Conventional			**Double DGS**			**Double Slot**		
	Dir.	***Gain***	***SLL***	***Dir.***	***Gain***	***SLL***	***Dir.***	***Gain***	***SLL***
Port 1	35	0.5	-5.2	37	1.42	-8.8	36	2.26	-10.9
Port 2	18	0.92	-6.5	18	3.47	-9.8	19	3.37	-11.3
Port 3	0	1.45	-10.5	0	5.37	-11.6	0	4.67	-12.8
Average		0.96	-7.40		3.42	-10.07		3.43	-11.67

4. Results and Discussion

The performance of the reduced structure by double DGS or double slots is compared to the performance of the conventional Rotman model. After getting the appropriate technique, the reduced Rotman lens is integrated into the array antenna with transmission line adjustment. After that, a far field simulation is conducted to get the radiation pattern including the direction of the beam, gain, and sidelobe level.

Table 2. shows the comparison of the performance of the Rotman lens made by a conventional method, the reduced by Double DGS Technique, and the reduced by Double Slot Technique. There are no significant differences in terms of direction of the beam. While for the gain, the gain of the conventional is lower than the reduced, it is possible because the loss in the conventional is higher for its large structure. The sidelobe level is also lower in the conventional than the reduced, this could happen because the large structure of the conventional affects the radiation of the antenna for its nearfield is also larger.

To validate the result, fabrication and measurement were conducted. The one which is fabricated is the reduced with double slot. The bandwidth is quite similar in ports 1, 2, 8, and 9. Port 1, 2, 8, and 9 reach wide bandwidth characteristics. The port 3 shows multiband characteristic. This happens because port 3 receives more reflection for its design. Fig. 6 shows the measured Return Loss profiles of ports 1, 2, 3, 8, and 9 in the 2.3-2.6 GHz range.

Radiation pattern measurement is done in one full rotation for port 3, and -60 to 60 degrees for other ports. The simulations and the measurements show quite similar results. The measured gain and SLL are also similar to the simulation. The maximum difference is 2 dB. The multi-beam radiation patterns can be seen in Fig. 7. They indicate that the design with the new method is proper.

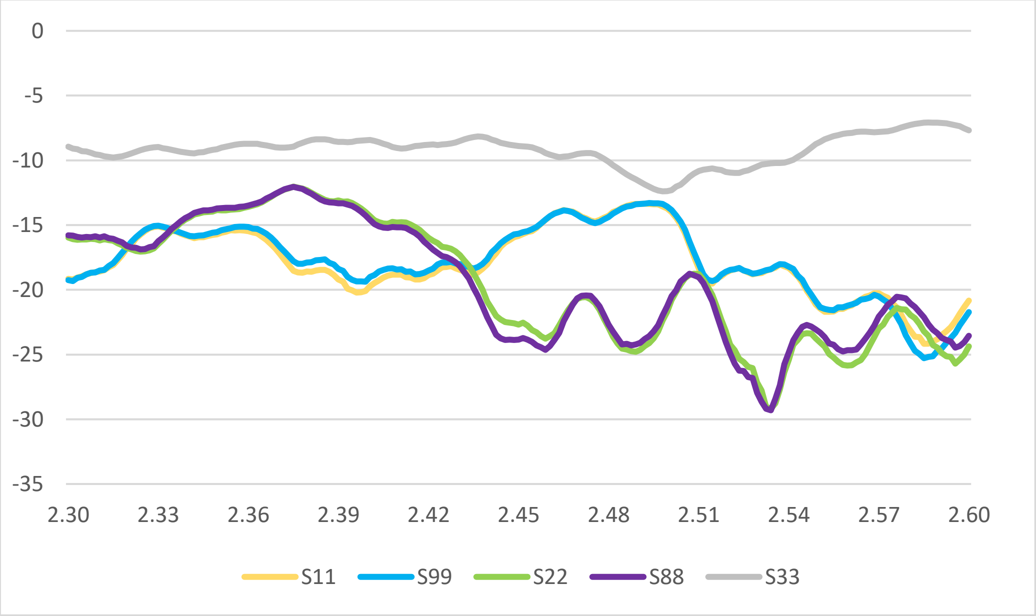


Fig. 6. Measured Return Loss Profiles of Port 1, 2, 3, 8, and 9 in 2.3-2.6 GHz Range

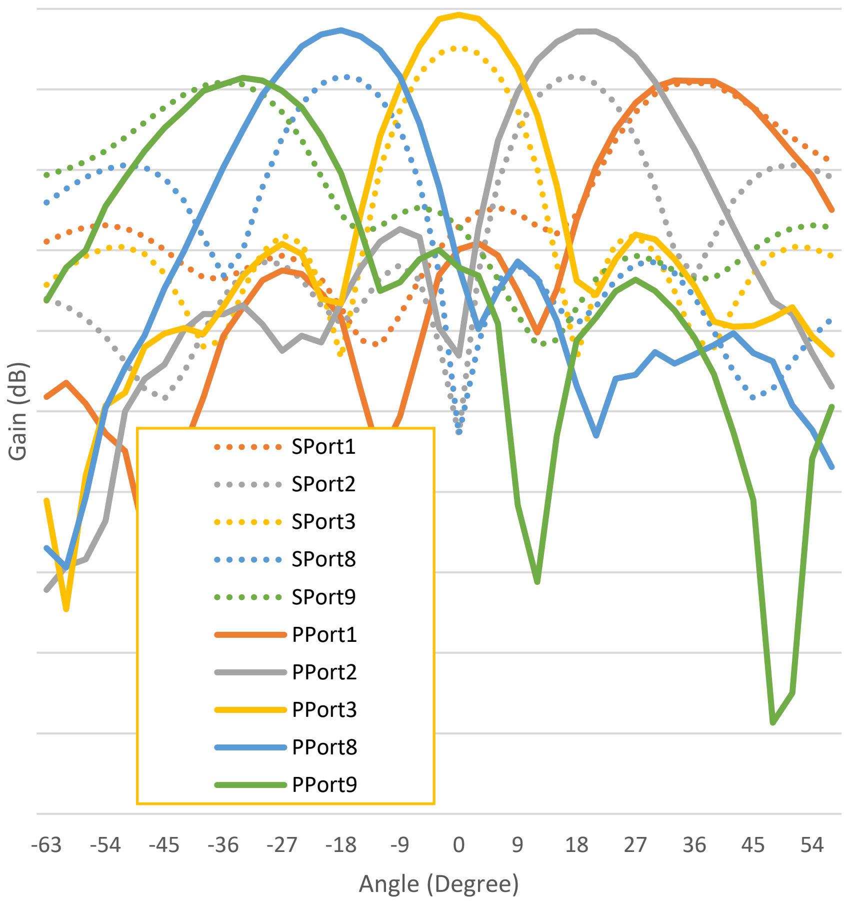


Fig. 7. Radiation Pattern of All Beam Port (f : 2.5 GHz) (dash line [SPort*x*] is simulation result, connected line [PPort*x*] is measurement result)

5. Conclusion

In this research, the miniaturization of Rotman by utilizing the DGS and slots method has been successfully designed, fabricated, and measured in S-Band frequency. The development of new techniques, namely the same-size double rectangular DGS technique and the same-size double rectangular slot, has been proven to decrease the size of the Rotman lens to become about 15 percent from its origin compared to the size of conventional design. This also means simpler design and fewer iterations can be achieved compared to the previous studies. No significant degradation of the performances resulted from the reduction countered as well.

References

[1] B. Shallah et al., "A Miniaturized Metamaterial-Based Dual-Band 4×4 Butler Matrix With Enhanced Frequency Ratio for Sub-6 GHz 5G Applications," in IEEE Access, vol. 12, pp. 32320-32333, 2024, doi: 10.1109/ACCESS.2024.3371027.

[2] Q. Yang et al., "A Low Complexity 16×16 Butler Matrix Design Using Eight-Port Hybrids," in IEEE Access, vol. 7, pp. 177864-177873, 2019, doi: 10.1109/ACCESS.2019.2958739.

[3] G. Buttazzoni et al., "A Simple Blass Matrix Design Strategy for Multibeam Arbitrary Linear Antenna Arrays," in IEEE Transactions on Antennas and Propagation, vol. 71, no. 11, pp. 8514-8524, Nov. 2023, doi: 10.1109/TAP.2023.3310148.

[4] D. I. Lialios, C. L. Zekios and S. V. Georgakopoulos, "A Compact mmWave SIW Blass Matrix," 2021 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting

(APS/URSI), Singapore, Singapore, 2021, pp. 961-962, doi: 10.1109/APS/URSI47566.2021.9703822.

[5] Y. Xu, H. Zhu and Y. J. Guo, "Compact Wideband 3 × 3 Nolen Matrix With Couplers Integrated With Phase Shifters," in IEEE Microwave and Wireless Technology Letters, vol. 34, no. 2, pp. 159-162, Feb. 2024, doi: 10.1109/LMWT.2023.3341791.

[6] H. Ren, H. Zhang, Y. Jin, Y. Gu and B. Arigong, "A Novel 2-D 3×3 Nolen Matrix for 2-D Beamforming Applications," in IEEE Transactions on Microwave Theory and Techniques, vol. 67, no. 11, pp. 4622-4631, Nov. 2019, doi: 10.1109/TMTT.2019.2917211.

[7] T. Djerafi, N. J. G. Fonseca and K. Wu, "Broadband Substrate Integrated Waveguide 4$\,\times\,$4 Nolen Matrix Based on Coupler Delay Compensation," in IEEE Transactions on Microwave Theory and Techniques, vol. 59, no. 7, pp. 1740-1745, July 2011, doi: 10.1109/TMTT.2011.2142320.

[8] K. Trzebiatowski, W. Kalista, M. Rzymowski, Ł. Kulas and K. Nyka, "Multibeam Antenna for Ka-Band CubeSat Connectivity Using 3-D Printed Lens and Antenna Array," in IEEE Antennas and Wireless Propagation Letters, vol. 21, no. 11, pp. 2244-2248, Nov. 2022, doi: 10.1109/LAWP.2022.3189073.

[9] M. Muhsin, K. Kamardin, Y. Yamada, Y. Sugimoto and K. Sakakibara, "Best Feed Positions and Radiation Patterns of Typical Multibeam Dielectric Lens Antenna," in IEEE Access, vol. 12, pp. 10497-10511, 2024, doi: 10.1109/ACCESS.2024.3352132.

[10] P. Castillo-Tapia et al., "Two-Dimensional Beam Steering Using a Stacked Modulated Geodesic Luneburg Lens Array Antenna for 5G and Beyond," in IEEE Transactions on Antennas and Propagation, vol. 71, no. 1, pp. 487-496, Jan. 2023, doi: 10.1109/TAP.2022.3217175.

[11] A. M. A. Najafabadi, F. A. Ghani, M. B. Ozdemir and I. Tekin, "5G Antenna Array Fed by a Microstrip Rotman Lens," 2023 IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications (APWC), Venice, Italy, 2023, pp. 141-141

[12] M. Zoghi, F. Hodjatkashani and M. E. Lajevardi, "An Ultra-Wideband Ridged Biconical Multibeam Antenna," in IEEE Access, vol. 11, pp. 58037-58045, 2023, doi: 10.1109/ACCESS.2023.3282871.

[13] A. M. H. Nasr and K. Sarabandi, "A Low-Cost Millimeter-Wave 5G V2X Multi-Beam Dual-Polarized Windshield Antenna," in IEEE Open Journal of Antennas and Propagation, vol. 3, pp. 1313-1323, 2022, doi: 10.1109/OJAP.2022.3224807.

[14] L. Stefanini et al., "Multibeam Scanning Antenna System Based on Beamforming Metasurface for Fast 5G NR Initial Access," in IEEE Access, vol. 10, pp. 65982-65995, 2022, doi: 10.1109/ACCESS.2022.3183754.

[15] B. G. Kashyap, R. Diaz and G. C. Trichopoulos, "Toward Hyper-Compact Rotman Lenses: Meta-Transmission Line Design and Characterization," 2023 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (USNC-URSI), Portland, OR, USA, 2023, pp. 797-798, doi: 10.1109/USNC-URSI52151.2023.10238073.

[16] R. K. Arya et al., "Compact Rotman Lens Design using Klopfenstein Taper Lines," 2022 IEEE 10th Asia-Pacific Conference on Antennas and Propagation (APCAP), Xiamen, China, 2022, pp. 1-2, doi: 10.1109/APCAP56600.2022.10069996.

[17] Q. Liang, B. Sun, G. Zhou, J. Zhao and G. Zhang, "Design of Compact Rotman Lens Using Truncated Ports With Energy Distribution Slots," in IEEE Access, vol. 7, pp. 120766-120773, 2019, doi: 10.1109/ACCESS.2019.2925000.

[18] M. K. Al-Obaidi, E. Mohd, N. Abdullah, and S. H. Dahlan, “A new approach for impedance matching rotman lens using defected ground structure,” Bulletin of Electrical Engineering and Informatics, vol. 9, no. 2, pp. 626–634, Apr. 2020, doi: 10.11591/EEI.V9I2.1850.

[19] H. -T. Chou and D. Torrungrueng, "Development of 2-D Generalized Tri-Focal Rotman Lens Beamforming Network to Excite Conformal Phased Arrays of Antennas for General Near/Far-Field Multi-Beam Radiations," in IEEE Access, vol. 9, pp. 49176-49188, 2021, doi: 10.1109/ACCESS.2021.3068831.

JRAEE

Journal of Robotics, Automation, and Electronics Engineering
Vol. 2, No. 1, 2024, pp. 39-50
e-ISSN 3025-3780
https://journal.uny.ac.id/publications/jraee

DVFS and Timing Optimization on GPU for Data Center Computation

Faris Yusuf Baktiar[a,1,*]

[a] Universitas Negeri Yogyakarta
[1] farisyusufbaktiar@uny.ac.id;
[*] Corresponding Author

ARTICLE INFO

Article History
Received 8 Feeb. 2024
Revised 28 Mar. 2024
Accepted 13 Jun. 2024

Keywords
Data Center;
GPU;
Computation;
DVFS;
Timing Optimization.

ABSTRACT

Data center computing requires efficient GPU support, both in terms of functionality and power consumption. GPU performance efficiency can be reduced due to high power usage and reduced GPU work stability. So it requires an analysis of computational performance and power efficiency to improve performance and reduce power usage. Core voltage, core frequency, and memory timings are parameters that affect the efficiency of computing performance, power efficiency, and stability. Increasing computational efficiency and GPU power with the effect of modifying parameters can be done through the Basic Input-Output System (BIOS). This study analyzes the efficiency of computational performance by optimizing memory timings and analyzing power efficiency and stability by modifying the DVFS algorithm. Tests are carried out using computational benchmarks commonly used in data centers including the tessellation algorithm, rendering, image processing, pi calculation, image stitching, deep learning, molecular simulation, and N-body. The efficiency of computing performance and GPU power efficiency can be increased by optimizing memory timings and changing the voltage and frequency values on DVFS. Increased performance efficiency ranged from 33.3% to 66.7% and power efficiency increased from 19.9% to 32.6%. Modification of the DVFS voltage state can increase voltage stability and GPU core frequency stability.

Komputasi pada Data center membutuhkan dukungan GPU yang efisien, baik dari segi kienrja maupu konsumsi daya. Efisiensi kinerja GPU dapat menurun dikarenkan penggunaan daya yang tinggi dan penurunan kestabilan kerja GPU. sehingga dibutuhkan analisis efisiensi kinerja komputasi dan daya untuk meningkatkan kinerja dan menurunkan penggunaan daya. Tegangan inti, frekuensi inti, dan timing memori merupakan parameter yang berpengaruh pada efisiensi kinerja komputasi, efisiensi daya, dan kestabilan. Peningkatan efisiensi komputasi dan daya GPU dengan modifikasi parameter yang berpengaruh dapat dilakukan melalui Basic Input-Output System (BIOS). Penelitian ini melakukan analisis terhadap efisiensi kinerja komputasi dengen optimasi timing memori, analisis efisiensi daya dan kestabilan dengan modifikasi algoritma DVFS. Pengujian dilakukan dengan menggunakan benchmark yang berisi komputasi yang biasa digunakan pada data center diantaranya algoritma tessellation, render, pemrosesan citra, kalkulasi pi, image stitching, deep learning, simulasi molekul, dan N-body.

ISSN 3025-3780 (online)
ISSN 3025-4590 (printed)

Komputasi pada Data center membutuhkan dukungan GPU yang efisien, baik dari segi kienrja maupu konsumsi daya. Efisiensi kinerja GPU dapat menurun dikarenkan penggunaan daya yang tinggi dan penurunan kestabilan kerja GPU. sehingga dibutuhkan analisis efisiensi kinerja komputasi dan daya untuk meningkatkan kinerja dan menurunkan penggunaan daya. Tegangan inti, frekuensi inti, dan timing memori merupakan parameter yang berpengaruh pada efisiensi kinerja komputasi, efisiensi daya, dan kestabilan. Peningkatan efisiensi komputasi dan daya GPU dengan modifikasi parameter yang berpengaruh dapat dilakukan melalui Basic Input-Output System (BIOS). Penelitian ini melakukan analisis terhadap efisiensi kinerja komputasi dengen optimasi timing memori, analisis efisiensi daya dan kestabilan dengan modifikasi algoritma DVFS. Pengujian dilakukan dengan menggunakan benchmark yang berisi komputasi yang biasa digunakan pada data center diantaranya algoritma tessellation, render, pemrosesan citra, kalkulasi pi, image stitching, deep learning, simulasi molekul, dan N-body. Efisiensi kinerja komputasi dan efisiensi daya GPU dapat ditingkatkan dengan optimasi timing memori dan pengubahan nilai tegangan dan frekuensi pada DVFS. Peningkatan efisiensi kinerja berkisar 33,3% hingga 66,7% dan efisiensi daya meningkat 19,9% hingga 32,6%. Modifikasi state tegangan DVFS dapat meningkatan kestabilan tegangan dan kestabilan frekuensi inti GPU.

1. Introduction

The data center was once considered only as a large place for data collection and just a place for data traffic. Data centers are now changing from providing storage only to providing computing services. Science and engineering computing is currently being done in Data centers, such as Deep learning, molecular simulation, image processing, and physical calculation simulation. Increased computational burden on the data center, requires a device that can speed up the computing process in the Data center. GPU is one of the accelerator solutions available today for a Data center [1].

The computing paradigm for the application of science and engineering has now shifted. The Central Processing Unit (CPU) which used to be the computational backbone for the application of science and engineering, has now been assisted or even replaced by a Graphic Processing Unit (GPU). The CPU is considered no longer able to perform all computing tasks quickly and efficiently, especially for science and engineering applications. This is supported by the presence of GPGPU [2].

GPU which is used in data centers, has considerable power requirements. This is due to the number of core processors used on GPUs numbering hundreds to thousands. The impact of the high power requirements is the emergence of high temperatures on the GPU chip. High power can cause a decrease in GPU computing performance efficiency [3]. In addition to decreasing performance, temperature, and high power consumption, it can disrupt the stability of the work of the GPU while handling computing [4]. Disturbed stability can cause data to be damaged error, or lost.

Data centers require GPUs that have fast performance, are stable at high computational durations that are long enough, but still have relatively low power to avoid high temperatures caused by power dissipation. One alternative to reduce power usage is by lowering the GPU voltage, decreasing the GPU core frequency, and optimizing memory timings.

This can be done by using one of the settings in the GPU kernel [5], but this method still has challenges where there are often voltage or frequency drops that still do not work according to the configuration given. So that it often disrupts work stability. Then a voltage modification method, timing, and GPU working frequency are needed. For the GPU to produce more stable, power-efficient performance, it can reduce GPU working temperature in computational intensive, such as its use in the data center.

The Basic Input-Output System (BIOS) is used in hardware as part of initialization. All states on the hardware will be initialized when the hardware boots. The BIOS has more access parameters than the kernel. This includes state voltage, voltage value, frequency value on DVFS, memory timings, and work frequency. Power efficiency and stability can be increased by changing the DVFS state and voltage values. The efficiency of computational performance can be improved by optimizing memory timing.

Seeing the problem of GPU computing performance efficiency in the data center which can be reduced by high power consumption and reduced GPU stability [6], this study will analyze the efficiency of computing performance and GPU power by timing optimization memory, changing the DVFS algorithm to the core voltage state, core voltage values and core frequency values embedded in the BIOS. This study focuses on analyzing the efficiency of computing performance, power efficiency, and stability of performance. The efficiency of computational performance, power efficiency, and stability are expected to be improved for computing using the GPU in the data center.

2. Method

This section explains how to implement BIOS optimization via DVFS and timing optimization. This section also explains the hardware and software used in this research as well as test methods that demonstrate computing stability and performance.

2.1 Overall Research Methods

The efficiency of GPU computing performance in this study will be improved by maintaining performance together by reducing GPU power consumption while increasing GPU stability. This is done to improve the efficiency of computing performance along with increasing the efficiency of power consumption with stable performance. Power efficiency will be improved by modifying the DVFS algorithm, both on state voltage, voltage value, and core frequency values. Computing performance will be maintained by optimizing memory timings. GPU computing performance can be measured by providing a benchmark containing the computational load that is processed to produce a measure of performance. The depiction of GPU computing performance in the data center requires a benchmark that contains computing commonly used in data centers such as pi digit calculations, 3D rendering and tessellation, molecular simulations, n-body simulations, and digital image processing. This research will use several algorithms that represent each of these computations. Computational benchmarks will be assisted by benchmarking software GPU power requirements in the computational process can be measured by recording GPU Power draw data when running computational benchmarks. Stability can be measured by recording core voltage and core frequency data from VRM and GPU cores.

2.2 Hardware Spesification Used

The hardware system that will be used in this study has the following specifications :

a. CPU AMD Ryzen 9 7900X 12 cores 24 threads running on 4.7 GHz

b. Motherboard AMD B650 Platform

c. RAM 64 GB (16GB X 4) DDR5 4800 MHz

d. Two GPUs AMD RX 6800 16GB GDDR6

e. Operating System: Windows 10 Pro 64-bit.

2.3 DVFS Optimization

The voltage and frequency at the GPU core are regulated using a power management algorithm on a computer component that is popular today, DVFS. DVFS works by adjusting the frequency and voltage at the GPU core to the given computational load. This is done so that the core frequency and voltage can adapt to the given load conditions to increase power efficiency [7].

DFVS in a BIOS is embedded in the GPU in the form of a voltage-state state setting. This state of voltage and frequency is a reference for the GPU kernel to carry out voltage and frequency adaptations. Reducing state frequency and voltage is done through the BIOS, by writing the state and value that you want to implement on the GPU.

Changing the DVFS rule is done at the voltage state. DVFS changes are expected to increase power efficiency and stability which will impact on improving the efficiency of computing performance [8]. Unstable voltage can be caused by the use of a voltage state where each state is too close. The voltage at the core is the main parameter in maintaining the frequency that runs at the core. The voltage state influences the power used and the frequency that can run at the GPU core. State voltage will be changed to less. GPU on default DVFS applies eight state voltages. State voltage will be converted into three voltage states, to accommodate idle state, mid-power state, and high-performance state

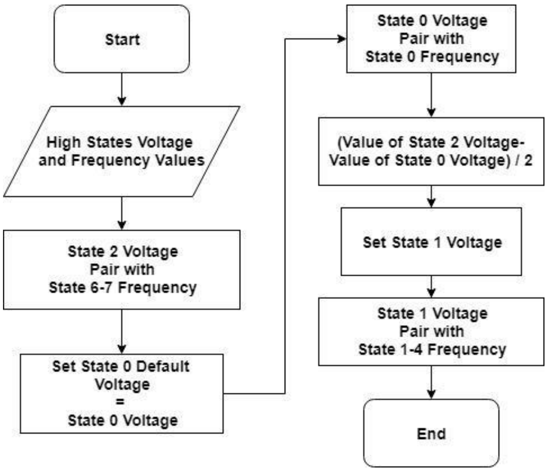

Start
High States Voltage and Frequency Values
State 2 Voltage Pair with State 6-7 Frequency
Set State 0 Default Voltage = State 0 Voltage
State 0 Voltage Pair with State 0 Frequency
(Value of State 2 Voltage-Value of State 0 Voltage) / 2
Set State 1 Voltage
State 1 Voltage Pair with State 1-4 Frequency
End

Fig. 1. DVFS modification flowchart

Many data center computations make the GPU work in two states, namely idle state, when not accepting computational loads, and high performance states when given high computational loads. This reduction in voltage state is expected to increase core and voltage frequency stability when given high computational loads to improve performance efficiency and reduce input power. The flow of changes in the voltage state of the DVFS is shown in Figure 1

2.4 Voltage and Frequency High-Performance State Tuning

Modification The state voltage in the DVFS algorithm requires a fixed voltage value for each frequency pair. Voltage states 1 to 7 on default DVFS have a variable value, only the state voltage 0 has a fixed value. Three voltage values are needed to meet the needs of three DVFS voltage states. **A**ssigning values to v0 will use the value v0 on the default DVFS. V2 value or upper voltage state will be taken based on the highest voltage reading from the kernel and then undervoled by 10-15%. This limit is given because the safe undervolting limit for computing use is 15-20%.

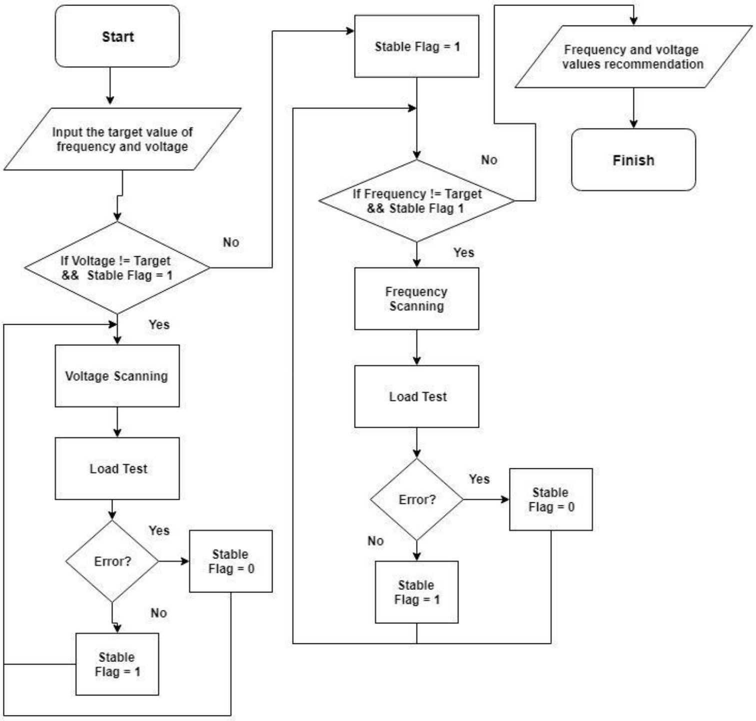


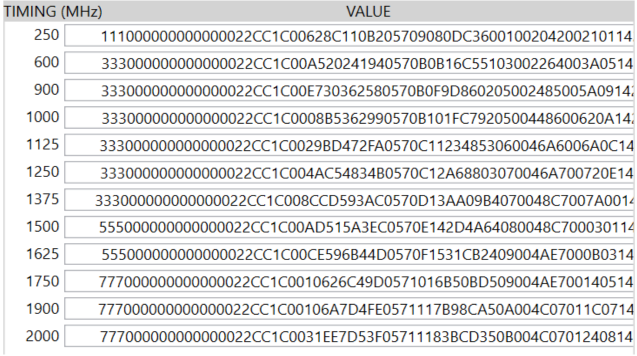
Fig. 2. Frequency tuning flowchart

Tuning is done on the kernel by scanning automatically through software, so it can be determined how much the undervolt can be done which will then be written on DVFS. The value v1 is the result of two from the difference between v0 and v2. Undervolting is done to produce lower voltage input at the GPU core so that the core power output can be lowered The upper frequency limit will be lowered to accommodate undervolting processes at the upper voltage state. Frequency tuning will be done on the kernel by automatic scanning through software with a 5% reduction limit. How it works tuning the upper limit of frequency and voltage using the scanning seen in the flowchart of Fig. 2.

The application of voltage and frequency values to be used at the highest state or state three will be determined by the tuning method. The other state, namely state one will follow the default voltage state zero in the default BIOS, and state two will be the difference between state one and three divided by two. Tuning is done by adjusting the frequency according to the voltage that want to achieve. The tuning process flowchart is shown in Fig. 2. The lowered voltage will result in the need for a decrease in frequency, so this study provides a limit of a decrease in the maximum frequency of 5% with a reduction in voltage targeted at 10% to 15%.

2.5 Memory Timing Optimization

Memory timing in the BIOS is in the form of a microcode as shown on the right side of **F**ig**.** 2. The microcode contains configuration timings at each work frequency level. The higher the frequency that is run, the timings used will be looser. The looser the memory timing, the slower the performance. Higher frequency states certainly have more loose timings, as seen in Figure 3. Timing optimization is done by utilizing each state**'s** timing. Microcode Timing at lower frequency states will be included at higher frequency states. This is intended to get high frequencies with tight timings. Increased memory computation perf**or**mance with a small power increase [9].



TIMING (MHz)	VALUE
250	111000000000000022CC1C00628C110B205709080DC3600100204200210114
600	333000000000000022CC1C00A520241940570B0B16C55103002264003A0514
900	333000000000000022CC1C00E730362580570B0F9D860205002485005A0914
1000	333000000000000022CC1C0008B5362990570B101FC7920500448600620A14
1125	333000000000000022CC1C0029BD472FA0570C11234853060046A6006A0C14
1250	333000000000000022CC1C004AC54834B0570C12A68803070046A700720E14
1375	333000000000000022CC1C008CCD593AC0570D13AA09B4070048C7007A0014
1500	555000000000000022CC1C00AD515A3EC0570E142D4A64080048C700030114
1625	555000000000000022CC1C00CE596B44D0570F1531CB2409004AE7000B0314
1750	777000000000000022CC1C0010626C49D0571016B50BD509004AE700140514
1900	777000000000000022CC1C00106A7D4FE0571117B98CA50A004C07011C0714
2000	777000000000000022CC1C0031EE7D53F05711183BCD350B004C0701240814

Fig. 3. Memory timing microcode

2.6 Stability Test and Computation Performance Test

Stability testing is done by providing a computational load on the GPU. Computation given as a stability test is compute-intensive and memory-intensive. Compute intensive will burden the GPU core with the intensity of the use of high GPU cores and intensive memory computing will burden memory with computing that requires high memory allocation. Computing that meets compute-intensive and also memory-intensive one of them is 3D image rendering, Tessellation 3D computing, and Deep Learning Model [10]. Computing that is run will be facilitated using software to provide the computations tested.

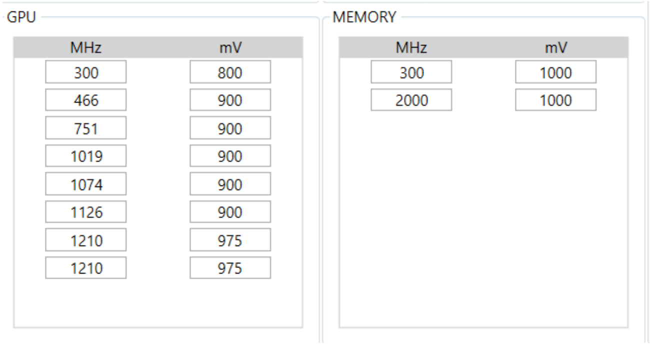
The stability test was used also to see the effect of changing the state voltage on the DVFS method. Reducing the voltage state is expected to increase stability by being able to adapt voltage based on 3-state voltage and reduce voltage and frequency drop when given a high computational load. Benchmarking is done to see the performance magnitude of a computing device. Benchmarking will be used to see how much GPU computing performance. Benchmarking will be done by providing computing based on computing in the data center. Tests are carried out using computational benchmarks commonly used in data centers including the tessellation algorithm, rendering, image processing, pi calculation, image stitching, deep learning, molecular simulation, and N-body. Benchmarking software for each computing load will also acquire GPU computing performance data.

3. Result and Discussion

3.1 DVFS and Timing Optimization Result

The DVFS modification produces three state voltages and eight state DVFS frequencies, and memory timing optimization by tightening 2 steps tighter. The best upper limit of the DVFS voltage produced is the lowest voltage that runs with the upper limit of the highest core frequency.

The modified DVFS results in three state voltages with a value of each that is read in the BIOS reader of 800 mV, 900 mV, and 975 mV shown in Fig. 4, with readable values of sensors namely 825 mV, 925 mV, and 1025 mV. The upper limit of the core frequency is 1210 MHz down by 50MHz or 4% of the default frequency value. A decrease in the upper limit of the core voltage read by the sensor in the log data is 150mV or decreases by 12.8% from the upper limit of the default voltage. Memory timing optimization is 2 steps higher without increasing memory voltage.



GPU

MHz	mV
300	800
466	900
751	900
1019	900
1074	900
1126	900
1210	975
1210	975

MEMORY

MHz	mV
300	1000
2000	1000

Fig. 4. State reduction and core frequency tuning result

3.2 Stability Test Analysis

Table 1 shows the modification of DVFS capable of increasing the level of stability compared to the default condition. Percentage percentages increased by 1.6% in 3D rendering computing. The difference is fairly insignificant but can describe the DVFS modification can achieve the same level of stability or even more than the default condition. DVFS modification also shows stable performance by being able to pass the test of stability in 3D tessellation computing.

Table 1. **Table 1.** Stability test result

3D Render Stability Test		3D Tessellation Stability Test	
Condition	Result	Condition	Result
Default	98.1%	Default	Pass
Modified	99.7%	Modified	Pass

The difference in the results of the stability of the default condition and after modification of DVFS in the 3D rendering computation test was due to a small difference in the stability of the framerate produced during the testing process. The 3D rendering stability test works by providing the workload on the GPU in the form of moving animated image frames, where the output is a series of images. Stability is measured by changes in frame rate. A decrease in framerate will reduce the level of stability. The decrease in framerate is caused by a decrease in core frequency as the main working parameter of the stability test. Further analysis of the effect of DVFS modification will be elaborated on the results of core voltage and frequency readings.

3.3 GPU core voltage stability analysis

The difference in the results of the stability of the default condition and after modification of DVFS in the 3D rendering computation test was due to a small difference in the stability of the framerate produced during the testing process. The 3D rendering stability test works by providing the workload on the GPU in the form of moving animated image frames, where the output is a series of images. Stability is measured by changes in frame rate. A decrease in framerate will reduce the level of stability. The decrease in framerate is caused by a decrease in core frequency as the main working parameter of the stability test. Further analysis of the effect of DVFS modification will be elaborated on the results of core voltage and frequency readings.

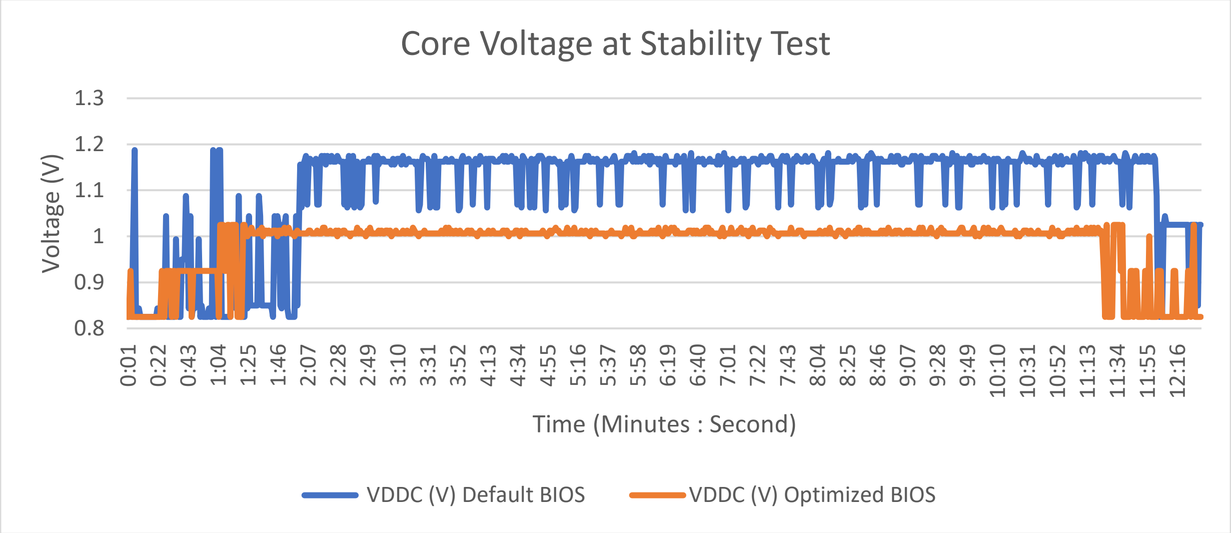


Fig. 5. Core voltage reading at 3DMark stability test

The voltage is capable of meeting the load given with the Modified DVFS three-state voltage shown in Fig. 5. The voltage can adapt to three states, namely at 825 mV, 925 mV, and 1025 mV. Look also at Figure 5 DVFS with three state voltages capable of increasing the stability of the core voltage by lowering the ripple when the GPU is given a test load. The ripples produced are only 25 mV compared to the default BIOS that uses eight DVFS states which produce ripples of 120 mV. DVFS with fewer state voltages can adapt more quickly to systems that run long in idle state or old with high-performance state as done in the stability test. Faster adaptation due to fewer state voltages, so that the transition between idle state or power state medium to high-performance state is faster by passing fewer state stages. Voltage ripples can also be reduced by reducing the voltage state on this DVFS, by minimizing voltage changes due to changes in the computational load

GPU core frequency stability analysis

Frequency ripple occurs in the default condition before modification of the DVFS voltage state when the stability test is performed, but does not occur after the DVFS voltage state is modified. Frequency ripples that occur under default conditions occur up to 40 MHz as shown in graph Figure 6. The core frequency ripples are influenced by the GPU core voltage.

The drop voltage will cause DVFS under default conditions to adjust the core frequency with a readable voltage level. DVFS with eight state voltages has a narrower frequency difference in each state so that the voltage drop will automatically cause a drop in the core frequency. DVFS with three state voltages has a wider voltage range for each frequency state, so that with a slight ripple on the voltage it will not cause a decrease in core frequency.

The core frequency is directly related to GPU computing performance. Decreasing core frequency values will have an impact on the decline of most computing performance. Drops that occur will reduce performance stability and reduce the efficiency of computational performance, therefore increasing core frequency alignment is one step to improve the efficiency of GPU computing performance.

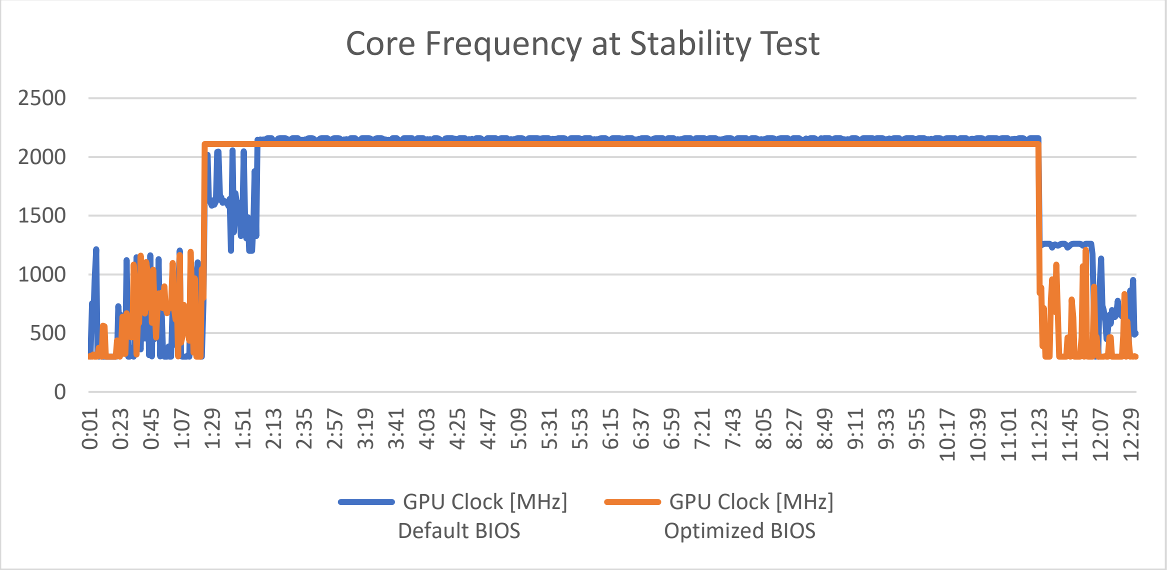


Fig. 6. Core frequency reading at 3DMark stability test

3.4 Computation Test Result and Analysis

Computational science benchmarks in this research use computation, namely: computational algorithm spigot for digit pi calculation through GPUPI software, computational simulation of molecule folding open MM by FAHBenchmark, computation of N-Body simulation by Compubench, execution performance of sobel operator algorithm and histogram equalization algorithm through Geekbench , computational image stitching, SIFT algorithm by Agisoft photoscan, and 3D rendering computation by 3DMark Fire Strike software, and Deep learning using Convolutional Neural Network (CNN).

Table 2. Benchmark results

Benchmark	Result		Memory Allocation (MB)	Difference (%)
	Default BIOS	Optimized BIOS		
Spigot Algorithm	989 seconds	1029 seconds	166	-4.0
OpenMM	63.2255 points	61,5068 points	120	-2.7
GeekBench	118663 points	116162 points	320	-2.1
N-Body Simulation 128k	332.169 iteration/s	340.697 iteration/s	520	2.6
N-Body Simulation 1024k	51.2394 iteration/s	52.5324 iteration/s	650	2.5
3D Render	24166 points	23893 points	1500	-1.1
Image Stitching (669 Gambar)	1211 seconds	1217 seconds	540	- 0.1
Image Stitching (129 Gambar)	141 seconds	141 seconds	620	0
VGG16 (CNN)	16.8 FPS	19.6 FPS	7616	16.7
VGG19 (CNN)	14.9 FPS	16.3 FPS	7571	9.4
MobileNet (CNN)	64.0 FPS	70.6 FPS	4247	10.3

The results of computational performance in Table 2 show a variety of computational performance response responses to optimizing memory timings and decreasing core frequencies. The minus value in Table 2 shows a decrease in computational performance after optimization of memory timings with decreasing frequency values, while a positive value difference indicates an increase in computing performance.

The spigot algorithm is the most effective computing effect on the decrease in core frequency and does not seem to be affected by the optimization of memory timings. N-body simulation is computing that has increased performance even though the core frequency has decreased. Based on the percentage increase, the n-body simulation shows the effect of optimizing the memory timing at the most increase in performance. Computing other than the N-body simulation, decreased performance due to a decrease in core frequency, but the decrease in performance was not as large as the percentage decrease in core frequency. This shows the effect of optimizing memory timings on computing performance.

The tendency of the magnitude of the effect of core or memory on computing can be classified by looking at changes in performance compared to memory usage as shown in Table 2. High memory usage with positive performance changes or indicates an increase in performance with the optimization of memory timings shows the tendency of memory bound. Low memory usage with negative performance changes indicates a decrease in performance where the effect of core frequency reduction is greater than the optimization of memory timings showing core bound tendency. Memory timing optimization will be optimal if the benchmark has a large GPU memory allocation and has low computational intensity so that the effect of data transfer speed on memory will be higher. Data transfer in memory is very dependent on memory frequency and memory timings.

3.5 GPU Output Power and Temperature Analysis

The GPU core is the component that consumes the most power from the entire GPU board. The decrease in core power consumption is due to a decrease in the upper limit of the GPU core frequency on DVFS. Decreasing the upper limit value of voltage in DVFS which is used as a voltage value that runs on high performance state is very influential on GPU power usage.

Table 3. GPU power draw and output temperature

Benchmark	**Power Default BIOS**	**Power Optimized BIOS**	**Diff (%)**	**Temperature (Celcius) - Default BIOS**	**Temperature (Celcius) - Optimized BIOS**	**Diff (%)**
Spigot Algorithm Both GPUs	109.8	76.1	-30.7	73.6	61.4	-16.6
	105.6	77.0	-27.1	64.2	56.2	-12.5
OpenMM	102.2	70.7	-30.8	60.4	52.8	-12.6
Image processing	86.2	58.4	-32.3	43.0	39.6	-7.9
N-Body	116.2	82.1	-29.3	57.0	51.0	-10.5
3D Render (GPU1)	166.8	118.5	-29.0	68.0	58.4	-14.1
3D Render(GPU 2)	161.4	118.7	-26.5	62.6	54.6	-12.8
Image Stitching 669 Images Both GPUs	162.1	112.7	-30.5	88.0	73.0	-17.0
	145.5	110.6	-24.0	79.0	67.0	-15.2
Image Stitching 129 images Both GPUs	156.0	113.1	27.5	79.0	66.0	-16,5
	150.5	103.1	31.5	72.0	61.0	-15.3
VGG16	175.7	122.6	-30.2	58.7	52.0	-11.4
VGG19	174.0	120.4	-30.8	60.3	52.0	-13.8
MobileNet	115.0	80.9	-29.7	49.3	45.7	-7.3

Table 3 shows a decrease in core voltage and a decrease in core frequency on DVFS which can reduce overall GPU power consumption in all computing power outputs.

It can be seen in Table 3 that there is a decrease in the output power modified by the DVFS voltage value compared to the default conditions in all tests which are marked with a percentage that is minus. Core voltage reduction has been proven to reduce the power drawn by the GPU which impacts system power down. GPU power ranges from 24.0% to 32.3% and system power supply ranges from 15.5% to 25.7%. System power is measured in the state of the system in high-performance conditions.

The varying power consumption values in Table 3 show the computational intensity performed at the GPU core. The higher computational intensity will increase the power drawn by the GPU core. The many calculations performed and data transactions will affect power. The core requires higher power when allocation, data size, transaction intensity, and calculation on compute units on the GPU core increases. The variety of power consumption can show the size of the intensity of each computation.

3.6 Performance per Watt

Performance per watt is a measure to see how much the efficiency of a computer component in carrying out its functions. Performance per watt in this study is used as a measure of **the** efficiency of the GPU. The purpose of using performance measures per watt is to see how far the efficiency of BIOS optimization increases. Performance per watt is calculated based on the score of each benchmark divided by the output power. Changes in performance per watt are written as percentages. The power size used is GPU power.

Table 4. Performance per watt

Benchmark	Performance Per Watt		Difference (%)
	Default BIOS	Optimized BIOS	
Spigot Algorithm	150137 digits/s/W	203024 digits/s/W	35.2
OpenMM	0.62 points/W	0.87 points/W	40.3
Image Processing	1377 points/W	1989 points/W	44.4
N-Body Simulation 128K	2.859 iteration/s/W	4.150 iteration/s/W	45.2
N-Body Simulation 1024K	0.441 iteration/s/W	0.640 iteration/s/W	45.1
VGG16 (CNN)	0.096 FPS/W	0.160 FPS/W	66.7
VGG19 (CNN)	0.085 FPS/W	0.135 FPS/W	58.8
MobileNet (CNN)	0.557 FPS/W	0.873 FPS/W	56.7
3D Render	73.6 points/W	100.7 points/W	36.8
Image Stitching 669 images	0.0018 images/s/W	0.0024 images/s/W	33.3
Image Stitching 129 images	0.0030 images/s/W	0.0041 images/s/W	36.7

Increased performance per watt in all tests carried out as shown in table 4. Benchmark computing computing shows an increase in efficiency of 35.2% to 45.2%. Deep learning benchmarks have increased efficiency by 45.8% -66.7%. The lowest increase occurred in the 3D rendering benchmark by 3DMark software, which amounted to 33.3%. The increase in efficiency was due to a combination of power losses caused by core frequency tuning and a decrease in core voltage coupled with GPU performance that was successfully maintained thanks to memory timing optimization.

4. Conclusion

DVFS and memory timing Optimization can increase the stability of GPU core voltage and frequency, decrease the temperature from 7.4% to 17%, decrease power from 19.9% up to 32.6%, and increase the GPU efficiency based on performance per watt from 33.3% up to 66.7% for data center applications.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] Y. Arafa, A. -H. A. Badawy, G. Chennupati, N. Santhi and S. Eidenbenz, "PPT-GPU: Scalable GPU Performance Modeling," in IEEE Computer Architecture Letters, vol. 18, no. 1, pp. 55-58, 1 Jan.-June 2019, doi: 10.1109/LCA.2019.2904497

[2] S. Najam, J. Ahmed, S. Masood and C. M. Ahmed, "Run-Time Resource Management Controller for Power Efficiency of GP-GPU Architecture," in IEEE Access, vol. 7, pp. 25493-25505, 2019, doi: 10.1109/ACCESS.2019.2901010.

[3] Y. Ma, J. Zhou, T. Chantem, R. P. Dick, S. Wang and X. S. Hu, "Improving Reliability of Soft Real-Time Embedded Systems on Integrated CPU and GPU Platforms," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 10, pp. 2218-2229, Oct. 2020, doi: 10.1109/TCAD.2019.2940681.

[4] [8] M. Smith, L. Zhao, J. Cordova, X. Jiang and M. Ebrahimi, "Energy-Efficient GPU-Intensive Workload Scheduling for Data Centers," 2023 International Conference on Machine Learning and Applications (ICMLA), Jacksonville, FL, USA, 2023, pp. 1735-1740, doi: 10.1109/ICMLA58977.2023.00263.

[5] S. . -K. Shekofteh, H. Noori, M. Naghibzadeh, H. Fröning and H. S. Yazdi, "cCUDA: Effective Co-Scheduling of Concurrent Kernels on GPUs," in IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 4, pp. 766-778, 1 April 2020, doi: 10.1109/TPDS.2019.2944602.

[6] C. A. García-Rodríguez, P. Quinto-Diez, J. A. Jiménez-Bernal, L. A. R. -D. León and A. Reyes-León, "Waste Heat Recovery System Applied to a High-Performance Video Card," in IEEE Access, vol. 8, pp. 6272-6281, 2020, doi: 10.1109/ACCESS.2020.2964207.

[7] J. Guerreiro, A. Ilic, N. Roma and P. Tomás, "Modeling and Decoupling the GPU Power Consumption for Cross-Domain DVFS," in IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 11, pp. 2494-2506, 1 Nov. 2019, doi: 10.1109/TPDS.2019.2917181.

[8] S. M. Nabavinejad, S. Reda and M. Ebrahimi, "Coordinated Batching and DVFS for DNN Inference on GPU Accelerators," in IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 10, pp. 2496-2508, 1 Oct. 2022, doi: 10.1109/TPDS.2022.3144614.

[9] H. Liu, S. Liu, C. Wen and W. E. Wong, "TBEM: Testing-Based GPU-Memory Consumption Estimation for Deep Learning," in IEEE Access, vol. 10, pp. 39674-39680, 2022, doi: 10.1109/ACCESS.2022.3164510.

[10] C. Zhang, F. Zhang, X. Guo, B. He, X. Zhang and X. Du, "iMLBench: A Machine Learning Benchmark Suite for CPU-GPU Integrated Architectures," in IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 7, pp. 1740-1752, 1 July 2021, doi: 10.1109/TPDS.2020.3046870.