Research Article

360-degree Image Processing on NVIDIA Jetson Nano

Prio Adjie Utomo[1], Arief Suryadi Satyawan[2*], Heni Puspita[3], Ike Yuni Wulandari[4]

[1,2,3,4] Nurtanio University Bandung, Bandung City, West Java, Indonesia
[2] Research Center for Telecommunication, National Research and Innovation Agency (BRIN), Indonesia
* Corresponding Author: a.s.satyawan@unnur.ac.id and arie021@brin.go.id

Abstract: A wide field of vision is required for autonomous electric vehicles to operate object-detecting systems. By identifying objects, it is possible to imbue the car with human intelligence, similar to that of a driver, so that it can recognize items and make decisions to prevent collisions with them. Using a 360-degree camera is a wonderful idea because it can record events surrounding the car in a single shot. Nevertheless, 360º cameras produce naturally skewed images. To make the image appear normal but have a bigger capture area, it is required to normalize it. In this study, NVIDIA Jetson Nano is used to construct software for 360-degree image normalization processing using Python. To process an image in real-time, first choose the image shape mapping that can give information about the entire item that the camera collected. Then, choose and apply the mapping. Using Python on an NVIDIA Jetson Nano, the author of this research has successfully processed 360-degree images for local and real-time video as well as image geometry modifications.

Keywords: Python; NVIDIA Jetson Nano; 360-degree Camera; Autonomous Electric Vehicle; Realtime; Conversion.

Citation: P.A.Utomo, A.S.Satyawan, H.Puspita, I.Y.Wulandari, "360-degree Image Processing on NVDIA Jetson Nano", *Iota*, **2024**, ISSN 2774-4353, Vol.04, 02. https://doi.org/10.31763/iota.v4i2.722

Academic Editor : Adi, P.D.P

Received : March, 14 2024

Accepted : April, 23 2024

Published : May, 1 2024

1. Introduction

Future electric vehicles that are autonomous will offer a host of benefits to an expanding number of vehicle users. Future owners of electric vehicles are expected to be able to steer the car without the need for human intervention. There will probably be an increasing need for this kind of convenience [1]. Autonomous electric vehicles need to be able to see a large area of space for their image-processing systems to function. When drivers identify objects and choose to avoid crashes with them, the car can be endowed with intelligence much like humans do. Therefore, to record the phenomena surrounding the vehicle, a broad vision system is required.

While standard cameras can be used for this task, about four cameras are needed to record the surrounding phenomenon simultaneously. These cameras should be positioned at the front, left, and right sides, as well as the back. Because of the numerous cameras and intricate processing, this is inefficient. Using a single 360-degree camera is an excellent alternative. The phenomenon surrounding the car can all be captured at once by the camera. As a result, in addition to using fewer devices, the processing is also simpler because it does not need to merge images from several cameras. But unlike images from regular cameras, 360º camera images are essentially warped, and that's why they're needed for real-time image capturing.

Consequently, normalizing the image is required to maintain the same capture area as a 360º camera while making it appear normal [2]. A 360° camera, also known as an omnidirectional camera, is a type of camera that can take larger or more comprehensive pictures simultaneously than traditional cameras. This comprises cameras with a 360° horizontal field of view as well as cameras with a 360° horizontal field of view and a 90° (preferably 180°) vertical field of view [3]. Although the photos from a 360° camera contain a lot of information, they are frequently distorted, producing images that are blurry or faint. As a result, to present images that are crisper and more precise, 360° camera photos must go through a particular process [4]. Additionally, the domains of robotics, localization and mapping, autonomous navigation systems, surveillance systems, and video communication systems employ 360-degree cameras [2]. Prior studies have included 360º video and picture compression, Visual Quality Assessment (VQA), and 360º image processing utilizing a visual attention modeling approach [5]. In the study, data sets and approaches for visual attention modeling for 360º videos and images were reviewed. Subsequently, the subjective and objective quality (VQA) of 360º videos and images was surveyed. Finally, methods for compressing 360º videos and images using spherical characteristics or visual attention models were reviewed.

The following study examines 360-degree image processing for autonomous electric vehicle systems' surveillance applications [6]. The study covers image processing using MATLAB software, which creates software for half cubes, RGB, Mercator, and real-time retrieval from 360-degree images. The following study examines the Ricoh Theta S camera's 360º dual-imaging picture calibration methods [7]. The study has relative orientation stability, including distance, basic element, and rotation matrix between distinct camera coordinate systems, and is based on the calibration of equidistant fisheye lens models. Because of this, an equidistant mathematical model that made use of the Conrady-Brown lens distortion model [8] worked well for determining the Ricoh Theta S's internal and relative orientation parameters. Some projects using the NVIDIA Jetson Nano can be seen in references [16-25], In this project, NVIDIA Jetson Nano is explored in detail by looking at all the performances that can be improved in each project, for example in Empowering Intelligent Manufacturing.

2. Method

This study aims to standardize images from 360º cameras using different normalizing techniques, and then employ the suitable mapping to project them in real-time. The NVIDIA Jetson Nano Developer Kit does all of this processing [9]. The chosen image processing outcomes will be used to project normalized images in real-time, followed by the collection of questionnaire responses to gather assessment findings and viewpoints from several sources. The phases of the research that has been conducted are described in Figure 1.

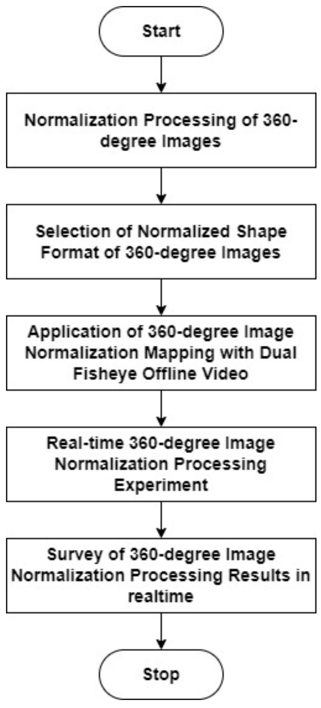

Start
Normalization Processing of 360-degree Images
Selection of Normalized Shape Format of 360-degree Images
Application of 360-degree Image Normalization Mapping with Dual Fisheye Offline Video
Real-time 360-degree Image Normalization Processing Experiment
Survey of 360-degree Image Normalization Processing Results in realtime
Stop

Figure 1. Flowchart of Research Stages

2.1 Processing 360-degree Images for Normalization

Several picture shape formats, including Fisheye, Dual-Fisheye, Square, Cubemap, Equirectangular, Perspective, and Tiles, are used in this study's 360º image normalization technique. The OmniCV library then helps with the processing of some of these formats [10]. The goal of image normalization mapping is to extract the geometry changes' shape from 360º photos. For 360º picture normalization processing in real-time, a shape format will be chosen. Experiments are needed in this process to obtain the proper normalizing image. Normalized photos must retain the object information captured by the camera while providing information from 360-degree camera images.

2.2 Spherical Coordinates

The Spherical Coordinates system or in mathematics denoted as (r, θ, φ) is a curvilinear coordinate system that describes the position of a sphere or spheroid. These coordinates are in the plane of the x, y, and z axes. The azimuthal angles in the x and y planes of the x-axis with $0 \leq \theta < 2\pi$ are denoted as θ (2), the polar angles of the positive z-axis with $0 \leq \varphi \leq \pi$ are denoted as φ (3), and the distance or radius from the starting point is denoted as r (1) [11]. Figure 2 is an illustration of spherical coordinates.

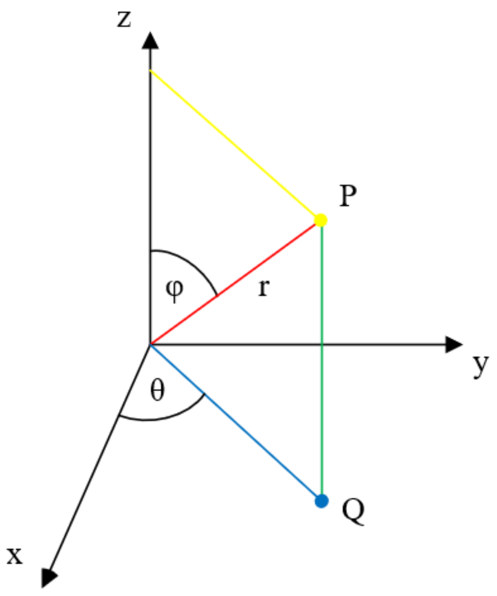


Figure 2. Illustration of Spherical Coordinates

The transformation from cartesian coordinates to spherical coordinates is:

$$r = \sqrt{x^2 + y^2 + z^2} \tag{1}$$
$$\theta = cos^{-1}\left(\frac{z}{r}\right) \tag{2}$$
$$\phi = tan^{-1}\left(\frac{y}{x}\right) \tag{3}$$

The back transformation or transformation from spherical coordinates to cartesian coordinates is [12]:

$$x = r\,sin\,\theta\,cos\,\phi \tag{4}$$
$$y = r\,sin\,\theta\,sin\,\phi \tag{5}$$
$$z = r\,cos\,\theta \tag{6}$$

2.3 Cubemap

A cubemap is a combination of six images that represent the six sides of a cube. This image provides information about the environment around the object, which is assumed to be the object in the center of the cube [11]. Each side has a horizontal and vertical Field of View of 90º. The shape of the Cubemap can be seen in Figure 3.

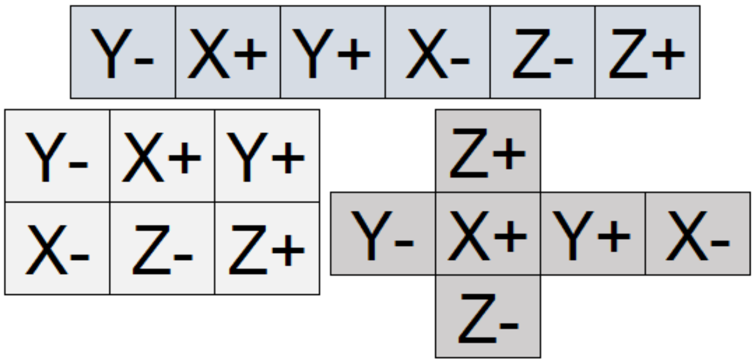


Figure 3. Cubemap Shape

2.4 Equirectangular

Equirectangular, cylindrical equidistant projection referred to as Plate Carrée (a carte parallélogrammatique projection) which is a form of map projection in which the equator line is made parallel to scale and free of distortion, and the meridian lines are equidistant in parallel and the square-shaped graticules are represented as two-dimensional images [11] [13]. An equirectangular image shape is a combination of images from two tilt-corrected camera sensors achieved by using attitude data (from image metadata) obtained from the camera's internal gyroscope sensor [7]. The graticule is the projection of the parallel lines of latitude and meridian lines of longitude drawn on the map and the edge lines on the map [14]. Figure 4 illustrates the equirectangular mapping.

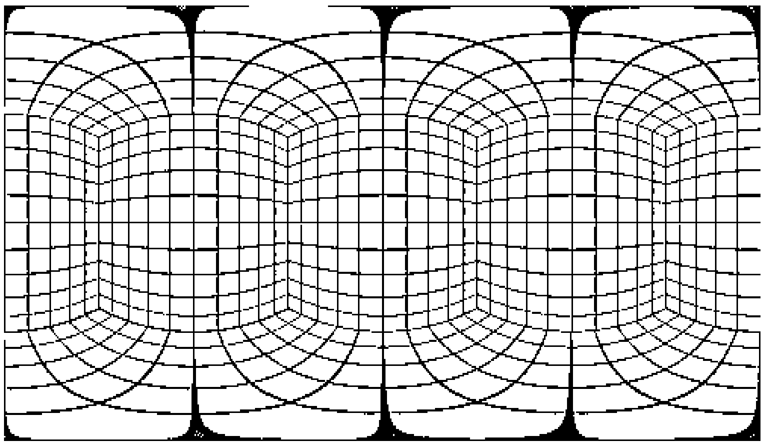


Figure 4. Equirectangular Mapping

Equirectangular images captured from the front and rear camera sensors have dimensions Nx and Ny, storing pixels whose image coordinates (column c and row l) have the upper left corner as the starting point. This image is normalized to xr and yr coordinates that range from 1 to +1, considering the starting point as the center of the image (7). Equirectangular images are created as a function of fisheye images where it is possible to assume polar coordinates (q, k), in equirectangular images ranging from p to +p for q and p/2 to +p/2 for k obtained by equation (8), these polar coordinates can be converted to three-dimensional spherical coordinates (px, py, pz) with a starting point or center point of the sphere using equation (9). The spherical coordinates (px, py, pz) can be used to estimate the spherical coordinates (θ, rs) of the point as in equation (10). The new coordinates (xf and yf) of the three-dimensional point of the sphere and the two-dimensional plane can be calculated by equation (11) [7].

$$x_r = -\left(-1 + 2 \cdot \frac{c}{N_x}\right);\ y_r = \left(-1 + 2 \cdot \frac{I}{Ny}\right). \tag{7}$$

$$\rho = \frac{\pi}{2}\left(x_r \cdot \frac{\pi}{2}\right); \lambda = \left(y_r \cdot \frac{\pi}{2}\right). \tag{8}$$

$$p_x = cos(\lambda) \cdot cos(\rho);\ p_y = cos(\lambda) \cdot sin(\rho);\ p_z = sin(\lambda). \tag{9}$$

$$\theta = tan^{-1}\frac{p_z}{p_x};\ r_s = \frac{2}{FOV} \cdot tan^{-1}\frac{\sqrt{p_x^2+p_z^2}}{p_y} \tag{10}$$

$$x_f = r_s \cdot cos\,\theta;\ y_f = r_s \cdot sin\,\theta. \tag{11}$$

2.5 Realtime 360º Image Normalization Processing

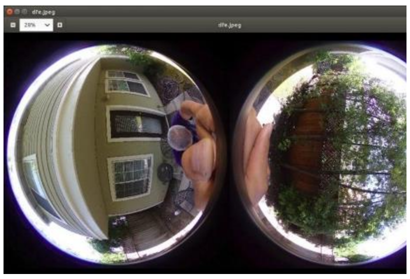
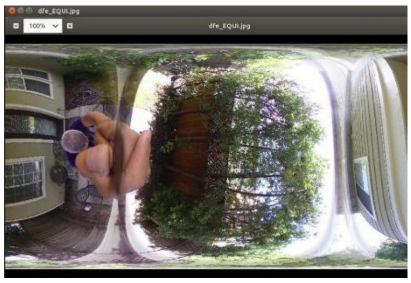
In this research, real-time 360º image processing uses the selected shape format to provide object information from 360º camera captures. Ricoh Theta S camera [15] is used to capture 360º images that will be normalized. But before the results are normalized in real-time, they will be tested first with a video recorded with the Ricoh Theta S camera. Real-time capture results are not only normalized but also processed to produce grayscale and canny images. This is used to test the simple processing of the 360º image.

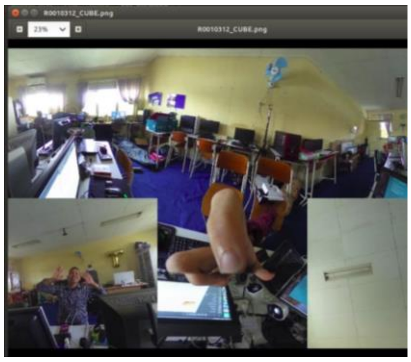
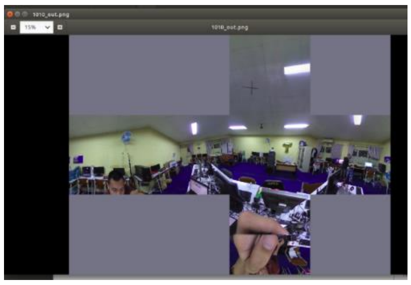
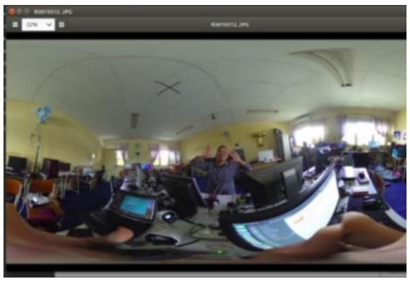
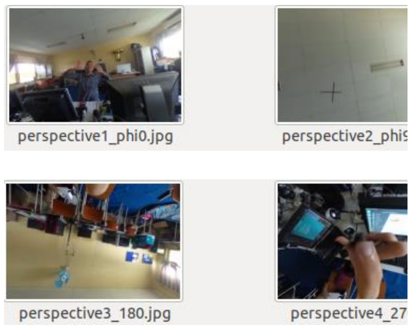
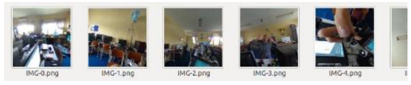
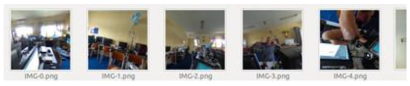
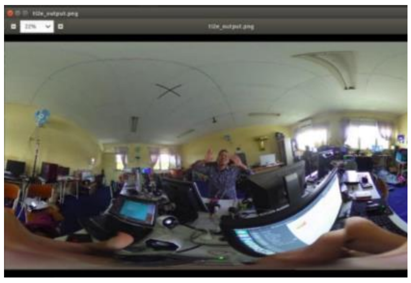
3. Result and Analysis

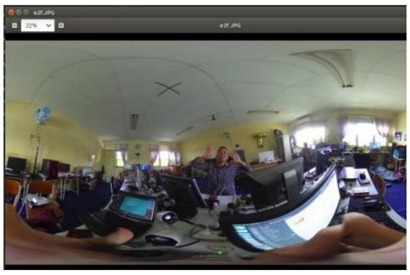
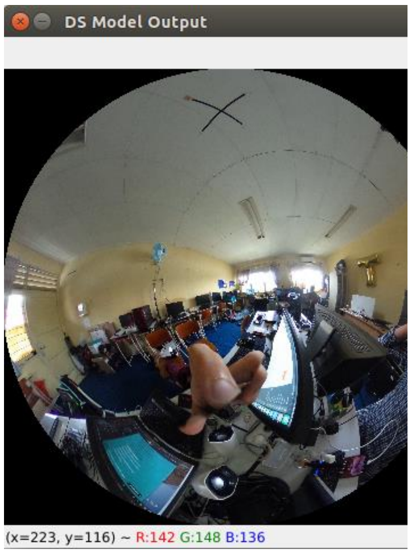
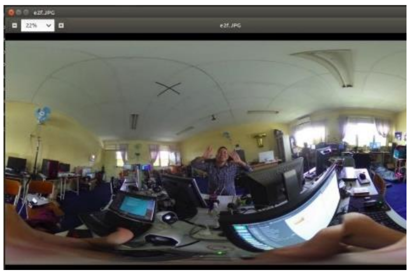
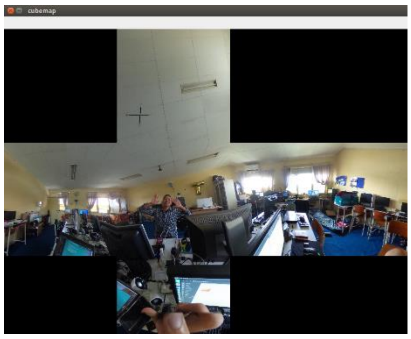
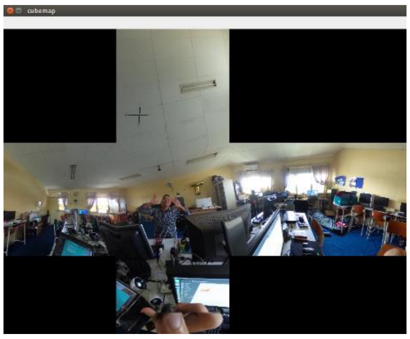
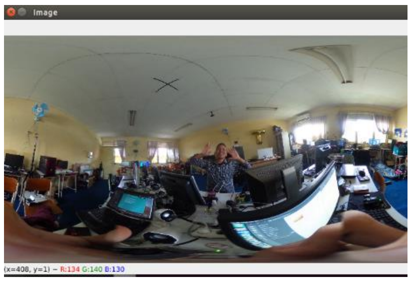
3.1 360-degree Image Normalization Processing Testing

Testing is done using several forms of mapping, namely Dual-Fisheye to Equirectangular, Equirectangular to Cubemap, Equirectangular to Perspective, Cubemap to Tiles, then with the help of OmniCV Library, namely: Equirectangular to Fisheye, Equirectangular to Cubemap, and Cubemap to Equirectangular. The normalization test results can be seen in Table 1.

Table 1. Normalization Testing Results

Conversion Name	Input	Conversion Results
Dual-Fisheye to Equirectangular		

Conversion Name	Input	Conversion Results
Equirectangular to Cubemap		 
Equirectangular into Perspective		 perspective1_phi0.jpg perspective2_phi9 perspective3_180.jpg perspective4_27
Cubemap to Tiles		 IMG-0.png IMG-1.png IMG-2.png IMG-3.png IMG-4.png
Tiles to Equirectangular	 IMG-0.png IMG-1.png IMG-2.png IMG-3.png IMG-4.png	

Conversion Name	Input	Conversion Results
Equirectangular to Fisheye with OmniCV Library		 (x=223, y=116) ~ R:142 G:148 B:136
Equirectangular to Cubemap with OmniCV Library		
Cubemap to Equirectangular with OmniCV Library		 (x=408, y=1) ~ R:134 G:140 B:130

In this test, it can be concluded that the processing from the 360 camera can be changed geometrically with various shapes. The equirectangular shape can be used to provide comprehensive information about the image captured by the 360º camera.

3.2 Testing Equirectangular Mapping with Local Dual-Fisheye Video

This test is done by taking the Equirectangular mapping that has been obtained from the previous test. The equirectangular mapping will convert the Ricoh Theta S camera capture video that provides dual-fisheye video capture results. Figure 5 is a screenshot of the video captured with the Ricoh Theta S camera. Figure 6 is the result of processing the video with equirectangular mapping. The results are declared usable for local video, and can then be used for further development.

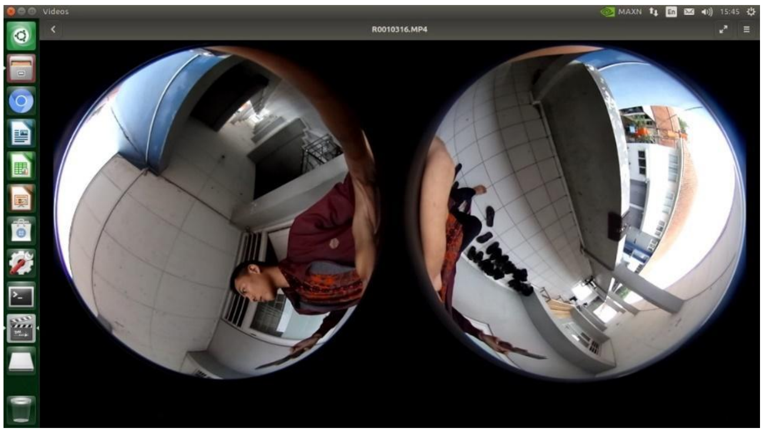


Figure 5. Screen Capture of Ricoh Theta S Video Results

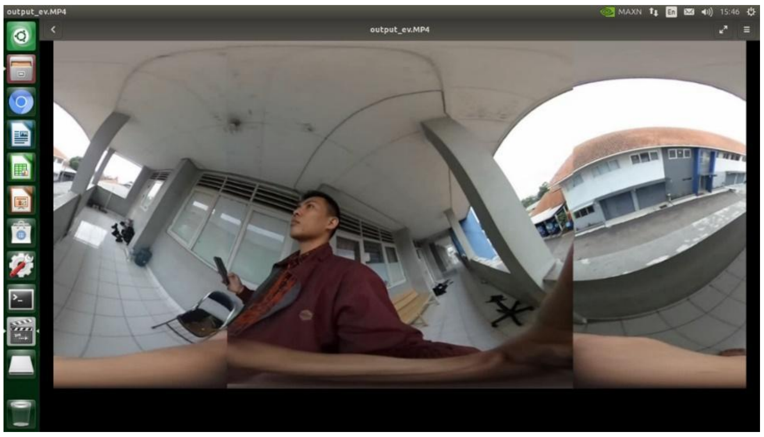


Figure 6. Screen Capture Results of Processing Results with Equirectangular Mapping

3.3 Realtime 360º Image Normalization Processing Testing

This test uses a Ricoh Theta S camera connected to an NVIDIA Jetson Nano. Initial testing by displaying a direct capture from the camera. The capture from the camera is in the form of a Dual-Fisheye shape which can be seen in Figure 7. Then testing is done by adding equirectangular mapping. The result of the mapping on the camera can be seen in Figure 8.

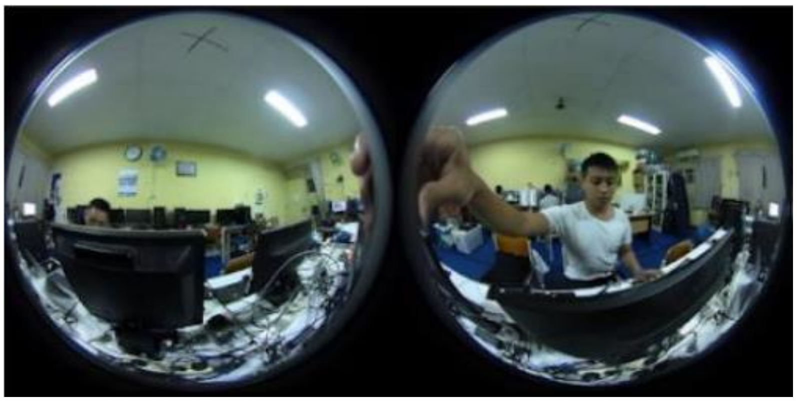


Figure 7. Camera Capture Display

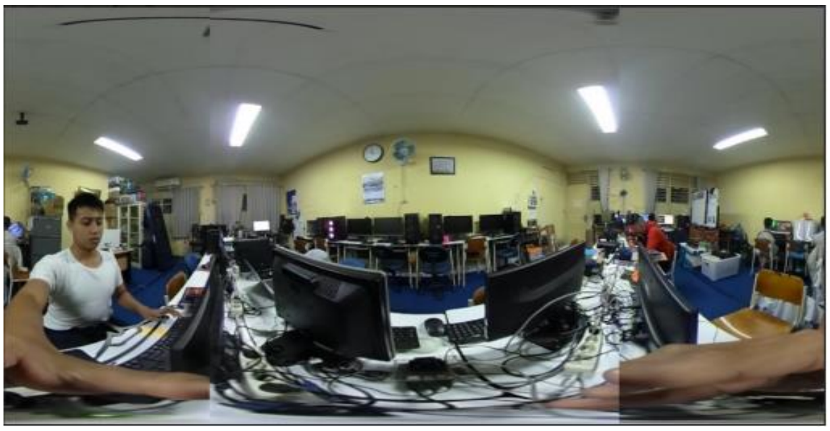


Figure 8. Display of Mapping Results

In addition to testing equirectangular mapping, processing of camera captures into grayscale and Canny Edge Detection was also carried out. This processing is also done in real-time using Python. In Figure 9, the results of grayscale processing and Canny Edge detection are shown in Figure 10. Then in Figure 11, the entire 360 image processing, grayscale, equirectangular normalization, and Canny Edge Detection are shown. The processing is displayed simultaneously on one screen.

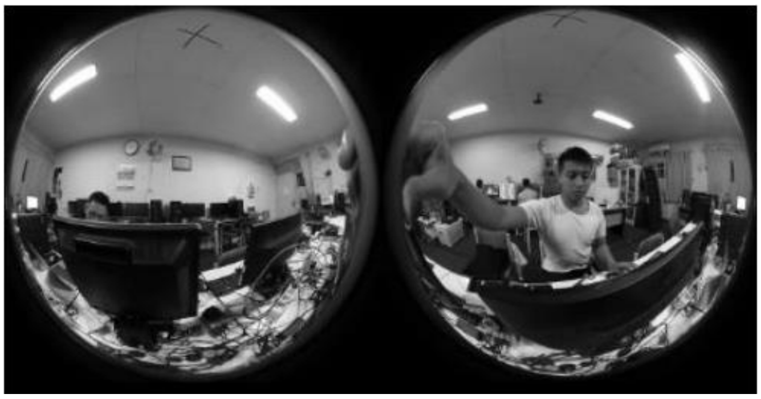


Figure 9. Grayscale Processing Result

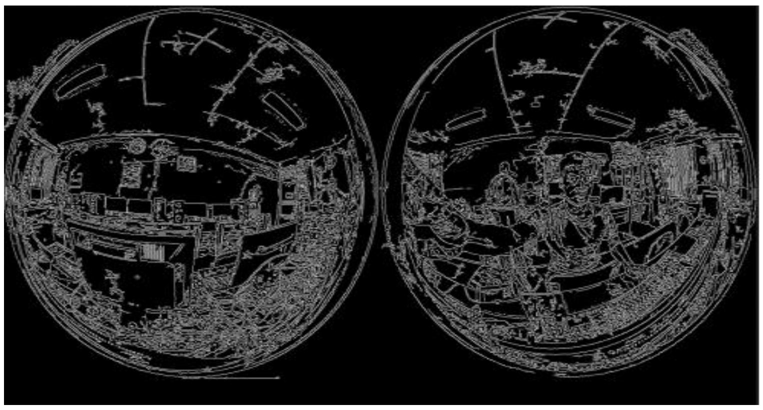


Figure 10. Canny Edge Detection Processing Result

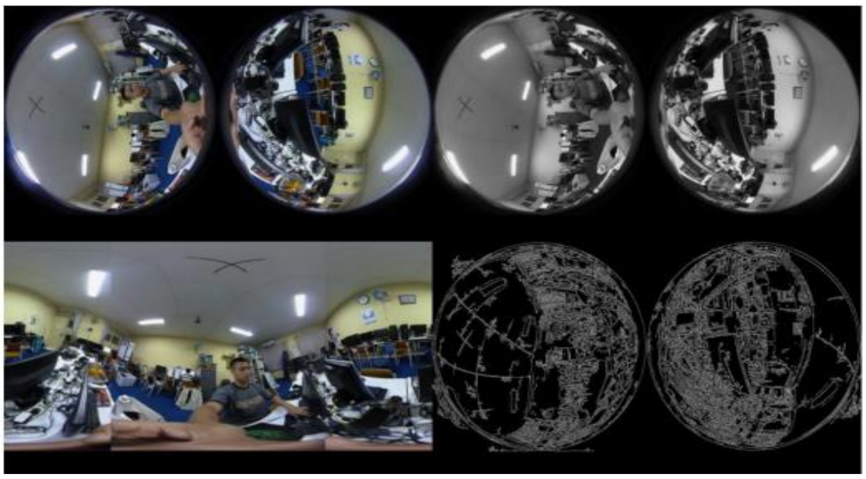


Figure 11. Processing Results of 360 Images, Grayscale, Equirectangular Normalization, and Canny Edge Detection

3.4 Realtime 360º Image Processing Software Assessment Survey

This survey was conducted to provide an assessment of the output results of real-time 360º image normalization processing. The assessment was carried out by filling out a voice questionnaire. The interviewee will be given information about the tool, the capture method, and a brief explanation of dual-fisheye images, grayscale, equirectangular, and canny edge detection. Then the interviewee will be shown the results of 360º image normalization processing in real time.

The questionnaire provided contains three options for assessing the quality of image processing results, namely good, not good, and not good. The image results are declared good if the main object does not change its physical form, it is declared less good if the main object has a slight change in its physical form, and it is declared bad if the object has a significant change in physical form. The survey results can be seen in Figure 12.

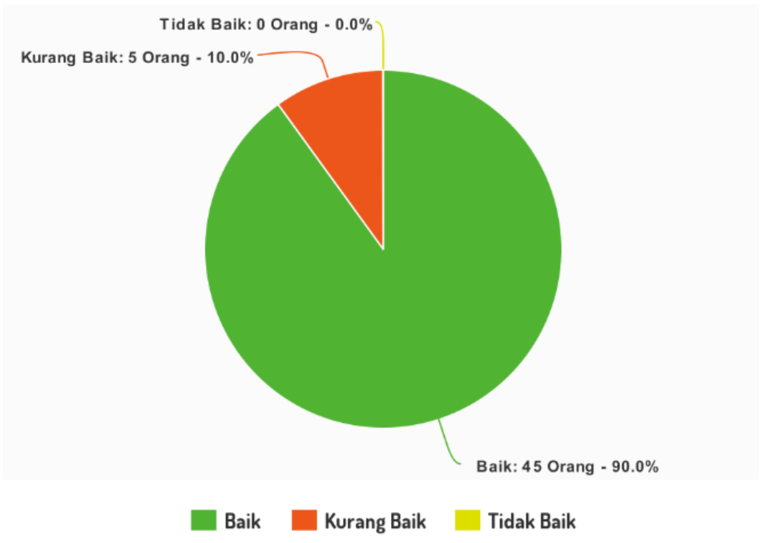


Figure 12. Image Normalization Processing Assessment Survey Results Chart

The survey was conducted with 50 people, giving the following results: 45 people gave a Good assessment, 5 people gave a Less Good assessment, and 0 people gave a Not Good assessment. From these results, it can be stated that the normalization processing of 360º images in real-time using equirectangular mapping can be understood by the public and can present information captured by 360º cameras as a whole.

3.5 Additional analysis for 360 degrees camera

Furthermore, the analysis of the 360-degree camera can be seen from the level of angle accuracy, Figure 13 uses a reference angle of 45 degrees, Figure 14 with an angle of 90, Figure 15 with an angle of 180, and Figure 16 with an angle of 360 degrees, from here we can see the value of angle accuracy compared to the reference angle. The line graph is a comparison of angle accuracy and perfect accuracy. The edge detection performance can be seen in Figures 17 and 18.

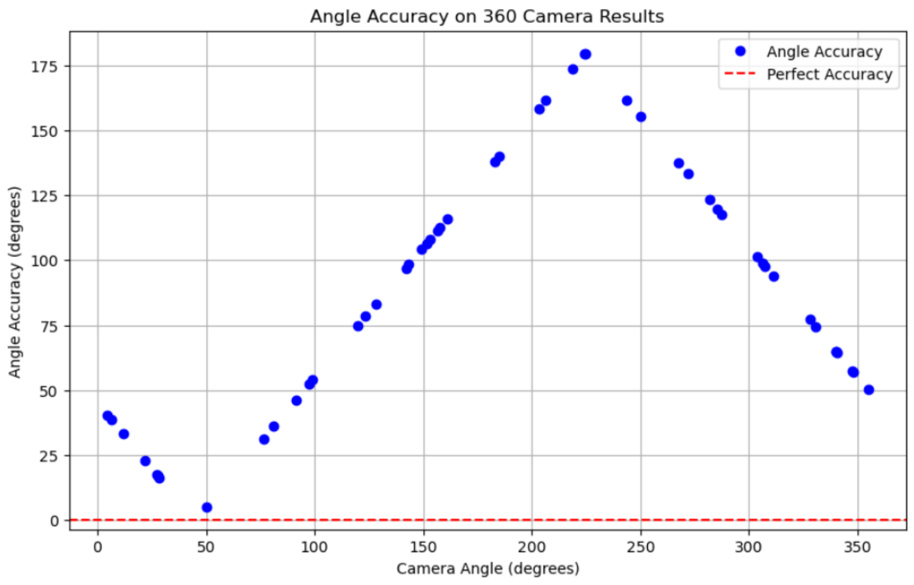


Figure 13. Angle Accuracy with reference_angle = 45.0

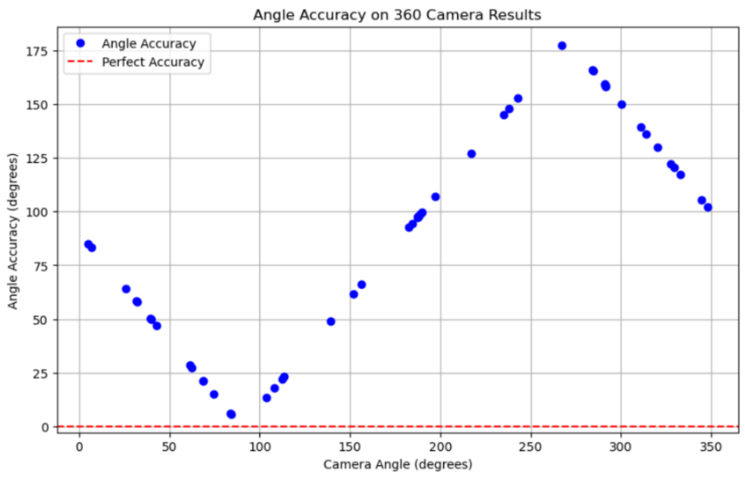


Figure 14. Angle Accuracy with reference_angle = 90

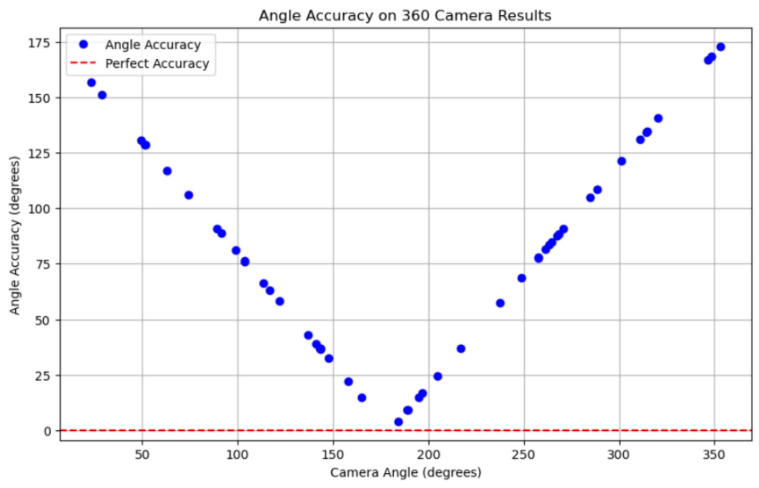


Figure 15. Angle Accuracy with reference_angle = 180

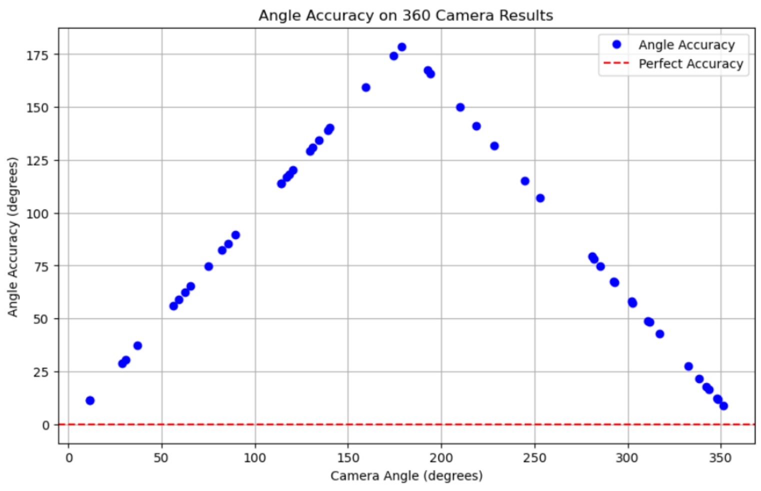


Figure 16. Angle Accuracy with reference_angle = 360

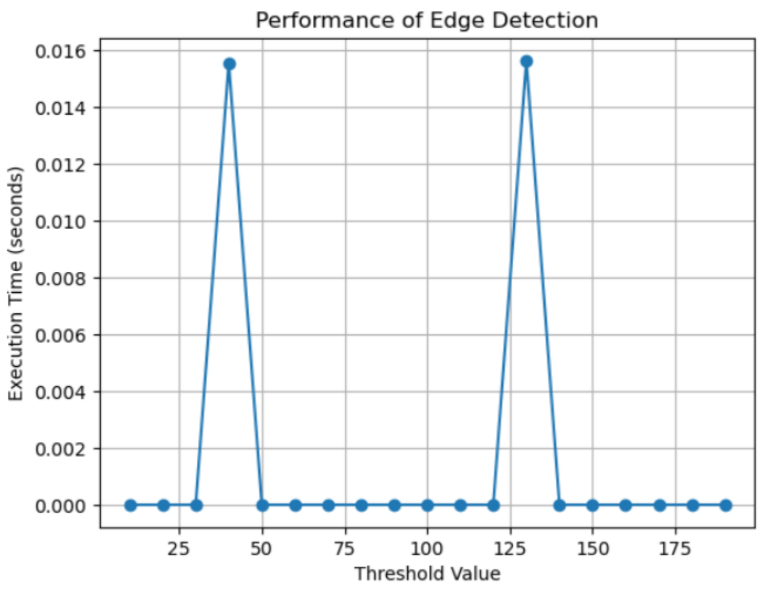


Figure 17. Performance of Edge Detection for 360 degress photo 1

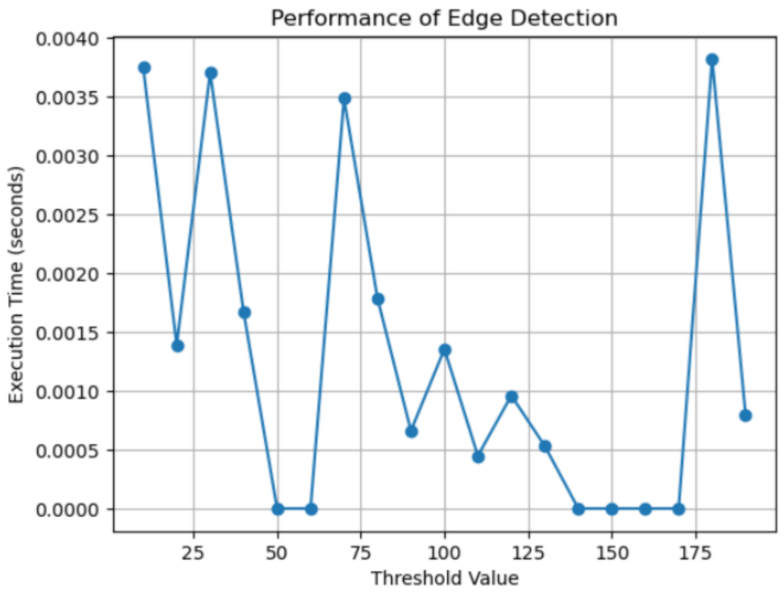


Figure 18. Performance of Edge Detection for 360 degrees photo 2

4. Conclusions and Suggestions

Based on the tests that have been carried out, it can be concluded that the normalization processing of 360º images using Python on the NVIDIA Jetson Nano Developer Kit mini computer can be done well. The normalization results with mapping provide ease of visualization of the 360 camera capture. The first test shows that the processing of the 360 camera (Dual-Fisheye) can be changed geometrically with various shapes (Equirectangular, Square, Fisheye, Cubemap, Tiles, and Perspective). The equirectangular shape can be used to provide information on the image captured by the 360º camera as a whole. In the second research, an experiment was conducted by utilizing equirectangular mapping on local video. The results obtained are that the mapping works well and provides video output that has the same quality. The third test was conducted using a Ricoh Theta S camera connected to an NVIDIA Jetson Nano. The test results obtained are 360 image normalization processing, equirectangular mapping, grayscale, and Canny Edge detection can also be applied in real-time. After the test is carried out, a survey is needed for image quality assessment of 50 people. With the survey results 90% gave a good assessment, 10% gave a poor assessment, and 0% gave an unfavorable assessment. It can be stated that the real-time 360º image normalization processing using equirectangular mapping can be understood by the public and can present the information captured by the 360º camera as a whole.

Acknowledgments: We would like to thank all parties, Lecturers, and Researchers, as well as the team at Nurtanio University Bandung, and the National Research and Innovation Agency (BRIN) who helped the publication process so that it could be completed properly, hopefully, this research can be useful for many people in general, and to the Internet of Things community in various places in Indonesia, especially in terms of the latest inputs that can be given from this research.

Author contributions: All authors are responsible for building Conceptualization, Methodology, analysis, investigation, data curation, writing—original draft preparation, writing—review and editing, visualization, supervision of project administration, funding acquisition, and have read and agreed to the published version of the manuscript.

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

References

1. M. N. Ahangar, Q. Z. Ahmed, F. A. Khan, and M. Hafeez, "A Survey of Autonomous Vehicles: Enabling Communication Technologies and Challenges," Sensors, vol. 21, no. 3, p. 706, Jan. 2021, doi: 10.3390/s21030706.
2. A. S. Satyawan, H. Watanabe, and J. Hara, "Scene flow from stereo fisheye images," in 2018 International Workshop on Advanced Image Technology (IWAIT), IEEE, Jan. 2018, pp. 1–4. doi: 10.1109/IWAIT.2018.8369690.
3. D. Scaramuzza, "Omnidirectional Camera," Computer Vision, pp. 552–560, 2014, doi: 10.1007/978-0-387-31439-6_488.
4. S. Abraham and W. Förstner, "Fish-eye-stereo calibration and epipolar rectification," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 59, no. 5, pp. 278–288, Aug. 2005, doi: 10.1016/J.ISPRSJPRS.2005.03.001.
5. M. Xu, C. Li, S. Zhang, and P. Le Callet, "State-of-the-Art in 360 Video/Image Processing: Perception, Assessment and Compression," IEEE Journal on Selected Topics in Signal Processing, vol. 14, no. 1, pp. 5–26, Jan. 2020, doi: 10.1109/JSTSP.2020.2966864.
6. Y. A. E. P. Manullang, "Pemrosesan Gambar 360 Unutk Aplikasi Surveillance Sistem Kendaraan Listrik Otonom," Bandung, 2022.
7. M. B. Campos, A. M. G. Tommaselli, J. Marcato Junior, and E. Honkavaara, "Geometric model and assessment of a dual-fisheye imaging system," The Photogrammetric Record, vol. 33, no. 162, pp. 243–263, Jun. 2018, doi: 10.1111/PHOR.12240.
8. J. P. de Villiers, F. W. Leuschner, and R. Geldenhuys, "Modeling of radial asymmetry in lens distortion facilitated by modern optimization techniques," D. P. Casasent, E. L. Hall, and J. Röning, Eds., Jan. 2010, p. 75390J. doi: 10.1117/12.838804.
9. "Jetson Nano Developer Kit for AI and Robotics | NVIDIA." Accessed: Feb. 09, 2023. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/
10. "Welcome to OmniCV Documentation — OmniCV 0.0.1 documentation." Accessed: Feb. 09, 2023. [Online]. Available: https://kaustubh-sadekar.github.io/OmniCV-Lib/
11. P. Reed, "Virtual Reality." Accessed: Feb. 08, 2023. [Online]. Available: http://www.paul-reed.co.uk/programming.html
12. R. Georgious, K. Elfeky, and R. R. Elrazky, "Scalar, phasors and vector magnitudes for electric and electronic engineering," in Encyclopedia of Electrical and Electronic Power Engineering, Elsevier, 2023, pp. 550–565. doi: 10.1016/B978-0-12-821204-2.00120-3.
13. J. P. Snyder, "Map projections: A working manual," Washington, 1987.
14. A. H. Robinson, "Elements of cartography. 6th ed.," p. 674, 1995, Accessed: Mar. 06, 2024. [Online]. Available: https://search.worldcat.org/title/704043771
15. "THETA S | RICOH IMAGING." Accessed: Mar. 06, 2024. [Online]. Available: https://www.ricoh-imaging.co.jp/english/products/theta_s/
16. E. Gawate and P. Rane, "Empowering Intelligent Manufacturing with the Potential of Edge Computing with NVIDIA's Jetson Nano," 2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2023, pp. 375-380, doi: 10.1109/ICCCIS60361.2023.10425278.
17. P. Inthanon and S. Mungsing, "Detection of Drowsiness from Facial Images in Real-Time Video Media using Nvidia Jetson Nano," 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Phuket, Thailand, 2020, pp. 246-249, doi: 10.1109/ECTI-CON49241.2020.9158235.
18. G. E. Edel, N. Borodina and M. E. Sukotnova, "Transfer Training of a Neural Network to Improve the Efficiency of Automatic Recognition of Children Using NVIDIA Jetson Nano," 2022 International Siberian Conference on Control and Communications (SIBCON), Tomsk, Russian Federation, 2022, pp. 1-4, doi: 10.1109/SIBCON56144.2022.10002974.
19. R. Dubey and R. Gajjar, "Real-Time Image Super-Resolution using Drone through GFPGAN and Nvidia Jetson Nano," 2023 9th International Conference on Signal Processing and Communication (ICSC), NOIDA, India, 2023, pp. 440-444, doi: 10.1109/ICSC60394.2023.10441545.

20. A. Basulto-Lantsova, J. A. Padilla-Medina, F. J. Perez-Pinal and A. I. Barranco-Gutierrez, "Performance comparative of OpenCV Template Matching method on Jetson TX2 and Jetson Nano developer kits," 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2020, pp. 0812-0816, doi: 10.1109/CCWC47524.2020.9031166.
21. A. A. Süzen, B. Duman and B. Şen, "Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN," 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 2020, pp. 1-5, doi: 10.1109/HORA49412.2020.9152915.
22. M. I. Uddin, M. S. Alamgir, M. M. Rahman, M. S. Bhuiyan and M. A. Moral, "AI Traffic Control System Based on Deepstream and IoT Using NVIDIA Jetson Nano," 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), DHAKA, Bangladesh, 2021, pp. 115-119, doi: 10.1109/ICREST51555.2021.9331256.
23. S. Holly, A. Wendt, and M. Lechner, "Profiling Energy Consumption of Deep Neural Networks on NVIDIA Jetson Nano," 2020 11th International Green and Sustainable Computing Workshops (IGSC), Pullman, WA, USA, 2020, pp. 1-6, doi: 10.1109/IGSC51522.2020.9290876.
24. R. S. Adari, S. S. Mirthipati, N. Malothu and R. K. Jatoth, "Real-Time Implementation of Modified LD-NET using Docker on NVIDIA Jetson Nano for Image Dehazing Application," 2023 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI), Gwalior, India, 2023, pp. 1-5, doi: 10.1109/CVMI59935.2023.10464528.
25. M. Choe, S. Lee, N. -M. Sung, S. Jung and C. Choe, "Benchmark Analysis of Deep Learning-based 3D Object Detectors on NVIDIA Jetson Platforms," 2021 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2021, pp. 10-12, doi: 10.1109/ICTC52510.2021.9621072.