Journal of Mechatronics, Electrical Power, and Vehicular Technology 15 (2024) 57-67

Journal of Mechatronics, Electrical Power, and Vehicular Technology

CrossMark

e-ISSN: 2088-6985
p-ISSN: 2087-3379

mev.brin.go.id

Design of image classification system for fabric inspection process using Raspberry Pi

Emmanuel Agung Nugroho [a, b], Joga Dharma Setiawan [b, *], Munadi [b], Diki [a]

[a] *Department of Mechatronics Engineering, Indorama Engineering Polytechnic*
Purwakarta, 41152, Indonesia
[b] *Department of Mechanical Engineering, Faculty of Engineering, Diponegoro University*
Semarang, Indonesia

Abstract

This research is designed as a prototype of defect inspection system on fabric production using machine learning-based image processing technology using the open source Google teachable machine application integrated with Raspberry Pi-3B. The prototype of fabric defect inspection system is built by utilizing two rollers that function as a fabric roll house before and after the inspection process. On both rollers, a fabric is stretched to be inspected, so that from a roll of fabric with a certain length, it can be seen how many defects occur on the fabric. The inspection system is carried out using a web camera with a certain level of lighting connected to a Raspberry Pi as a control device. Raspberry Pi functions as an image processing device and fabric rolling motor controller. In addition to the category of fabric in good condition, this system classifies into two categories of defects, namely slap defects and sparse defects. The test results show that this system has an average frame per second (FPS) of 4.85, an average inference time of 181.1 ms, with an accuracy of image classification results of 98.4 %.

Keywords: machine learning; image processing; Raspberry Pi; inference time; image classification.

I. Introduction

The textile and garment industry is the manufacturing sector that recorded the highest growth in the third quarter of 2019, which amounted to 15.35 % [1]. The textile industry is one of the five manufacturing sectors that are prioritized for development, especially in preparation for the industrial era 4.0, based on the Making Indonesia 4.0 road map [2]. Textile companies continuously strive to maintain the quality of their products to ensure customer satisfaction [3]. Defects in a product affect the level of customer satisfaction and have an impact on product quality standards and company production efficiency [4].

To get the quality standards and maintain customer satisfaction, quality control of fabric production is very important, especially to control fabric production defects [5]. In this study, we will take two samples of fabric defects as the object of research, which are sparse defects and slap defects [6].

The purpose of this research is to build a prototype of a computer vision-based fabric production defect inspection tool using a camera connected to the Google Teachable Machine application [7]. This machine prototype makes a new contribution to the existing fabric inspection system [8], as it is equipped with image classification technology that is not yet available on main stream fabric inspection machines in the industry today [9]. Teachable Machine is one of the

* Corresponding Author. joga.setiawan@gmail.com (J. D. Setiawan)

https://doi.org/10.55981/j.mev.2024.863

Received 16 March 2024; revision 10 June 2024; accepted 20 June 2024; available online 31 July 2024

How to Cite: E. A. Nugroho *et al.*, "Image classification system design for fabric inspection process using Raspberry Pi," *Journal of Mechatronics, Electrical Power, and Vehicular Technology*, vol. 15, no. 1, pp. 57-67, July, 2024.

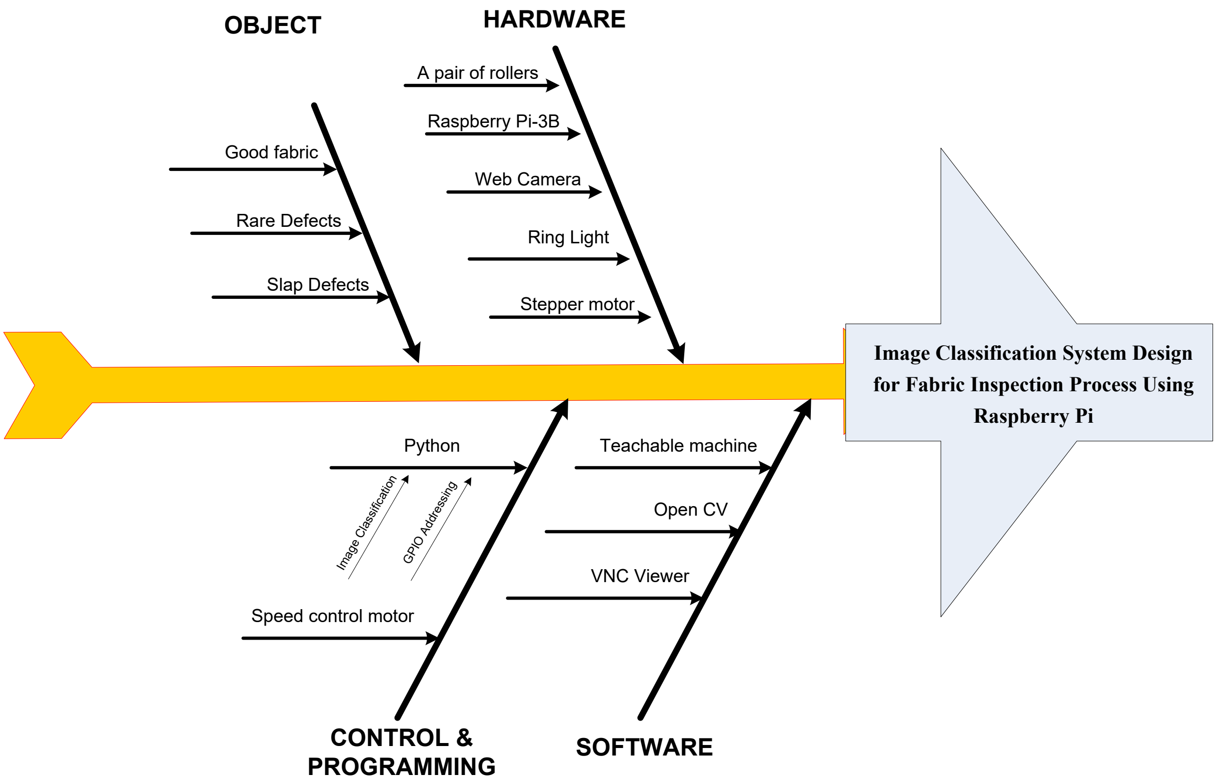

Figure 1. Fishbone diagram analysis of research problems.

open source machine learning application tools from Google that is capable of building an image classification system [10]. In this study, fabric production results are classified based on 3 categories, which are good fabric, sparse defects, and slap defects [11]. The output of Teachable Machine is Tensor flow or digital image data developed to run Machine Learning functions on Raspberry Pi [12]. Through the analysis of the fishbone diagram [13], the important factors for the implementation of this research can be explained as in Figure 1.

Figure 1 explains in general this research classifies 3 fabric objects, which are fabrics in good condition, fabrics with slap defects, and fabrics with sparse defects [14]. The Hardware section consists of mechanical equipment for fabric rollers and computer vision system support equipment, including ring light, Raspberry Pi-3B and web camera [15]. The software used is the Teachable Machine application to generate datasets, open CV for image processing on Raspberry Pi and VNC Viewer to display data displays and classification results [16]. Python programming is used for image classification processing and stepper motor control on Raspberry Pi [17].

When the system is implemented, a camera is in charge of reading fabric objects that rotate above the winding machine mechanism and reporting the inspection results to the Raspberry [18]. Through the image classification program process built in the Raspberry, the program system in the Raspberry will perform 3 operations at once, which are, continue to rotate the rolling motor if the classification results state that the fabric condition is good, stop the motor while turning on the buzzer if a defective fabric condition is found [19], and provide information on the type of defect captured by the camera through the VNC Viewer while displaying accuracy analysis and inference time during the classification process [20].

II. Materials and Methods

A prototype of the mechanical system of a fabric rolling device made to resemble a fabric rolling machine that is actually used in a weaving fabric producing company is shown in Figure 2.

This prototype machine for fabric production inspection consists of a mechanical system in the form of a set of fabric rolling mechanisms driven by a stepper motor [21], an electric system in the form of a DC voltage source, 4 relays, and indicator lights, a mechatronics system in the form of a Raspberry Pi, stepper motor driver TB 6600 and an informatics system that utilizes computer vision technology supported by ring light, web camera and image classification system technology [22].

The working principle of the image classification system on this fabric defect inspection machine is explained through Figure 3. The explanation of Figure 3 consists of two stages: first is the image classification stage where the web camera takes a number of fabric

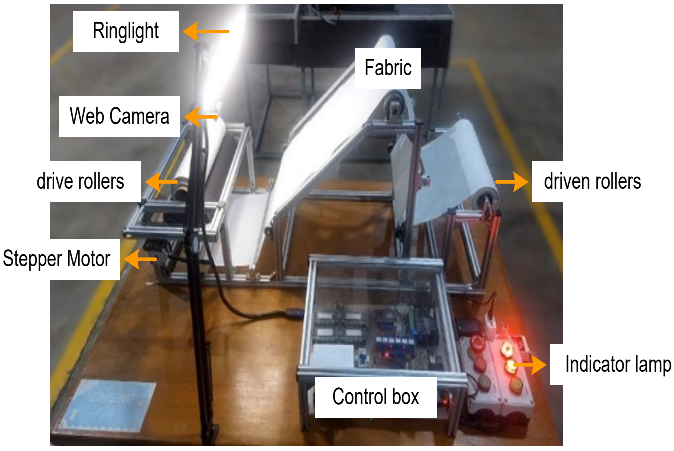

Figure 2. Fabric defect inspection machine prototype.

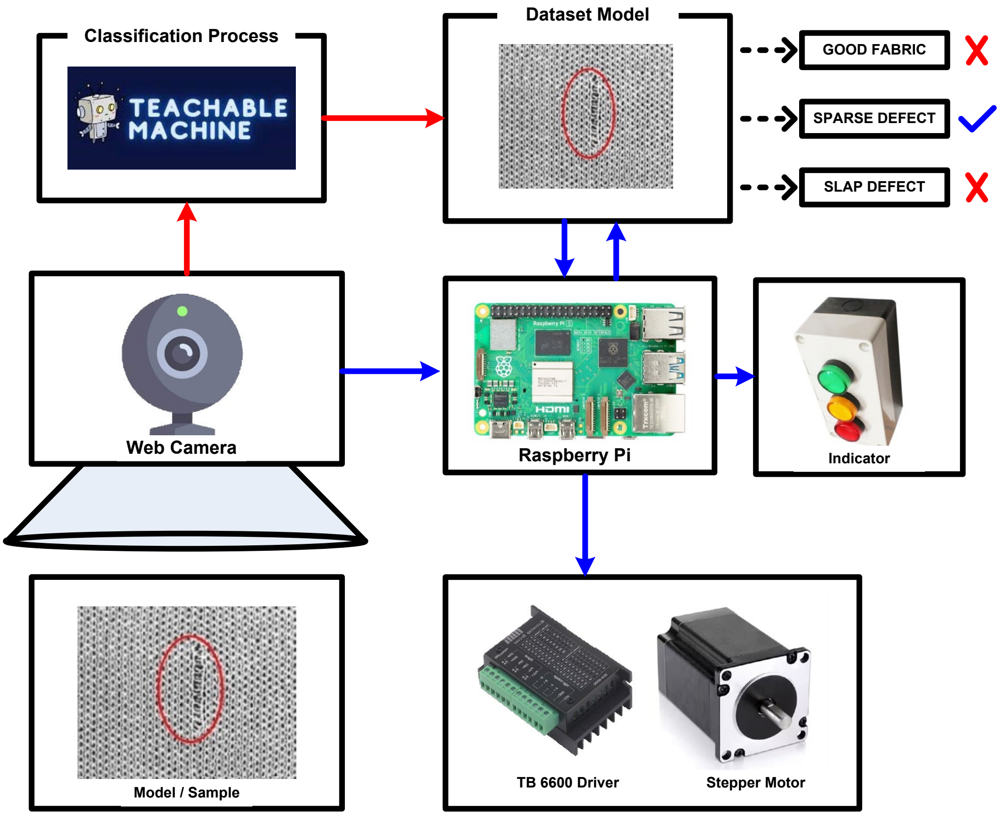

Figure 3. Image classification process diagram.

model samples consisting of good fabrics, fabrics with sparse defects, and fabrics with slap defects which are processed into model datasets through the teachable machine application [23]. The second is the deployment stage where the camera will read the fabric sample and forward it to the Raspberry Pi to match the fabric model read with the previously trained dataset. If the fabric that is read is a good fabric model, the Raspberry Pi will instruct the GPIO to continue to rotate the stepper motor, and if the model reading finds a sparse or slap defective fabric condition, the GPIO will be instructed to stop the stepper motor rotation and sound an alarm and a red indicator lights up [24].

A. Image classification preprocessing

This section consists of five stages, namely problem scoping and data acquisition, data exploration, modeling, and evaluation [25].

1) Problem scoping

The problem that has been investigated is the type of sparse defect and slap defect. Sparse defects are voids due to broken/lost threads in the warp. Slap defects are dirt or stains on the fabric. Figure 4 shows examples of fabric samples with Sparse defects and slap defects.

Figure 4 is a model view on the Teachable Machine tool. Based on Figure 4, the sparse defect and slap defect

Figure 4. Model of good fabric and defect fabric.

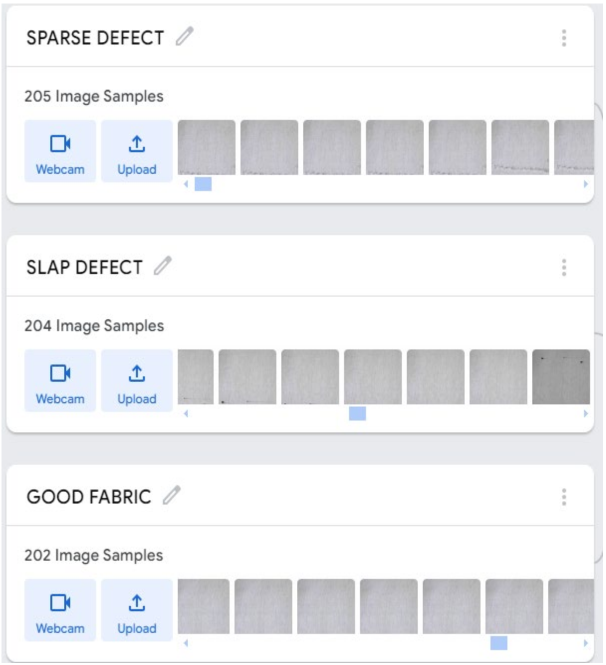

Figure 5. Collection of sample dataset.

models are used as examples to build an image classification model for the fabric inspection system. In this case, two main components are needed, namely the model and the image classification application. In this case, the model acts as the "brain" or core that performs the processing and decision making. While the Image Classification application acts as an interface that allows users or other systems to communicate with the model.

2) Data acquisition

The Data Acquisition stage is still in the Teachable Machine tool application [25], this stage serves to collect datasets and ensure dataset quality. Figure 5 shows the dataset obtained in the fabric model classification process. Figure 5 explains the stage of collecting fabric training data by classifying it into 3 class categories, which are good fabric, sparse defects and slap defects. Each training data has a number of samples to accommodate different types of models for more accurate classification. Table 1 shows the number of categories and the number of fabric model samples. Table 1 shows that there are 3 categories of samples from several models that were sampled. Good yarn 202 samples, Sparse defect 205 samples, and slap defect 204 samples.

Table 1.
The dataset sample.

No.	Category	Number of sample
1	Good fabric	202
2	Sparse defect	205
3	Slap defect	204

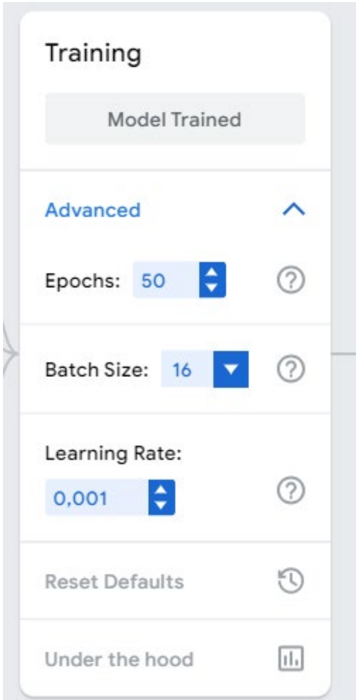

Figure 6. Teachable machine training process.

3) Modelling

Modelling is achieved through a training process on the Teachable Machine tool. Some important terms in this process are :

- Epochs: The total number of data sets used in the training model. The number of epochs determines how many iterations are required during the training process [26].
- Batch size: The number of data sets processed in a training iteration. Influences the speed of training and the use of computational resources [27].
- Learning rate: A parameter that controls how much the model parameters change during training. Determines how quickly or slowly the model converges to the optimal solution [28].

Figure 6 shows the training process on the teachable machine. Up to the training process as Figue 6 shows the Teachable Machine simplifies the machine learning training process, making it accessible to a broader audience. Keep in mind that while it's a great tool for introductory and educational purposes, more complex machine learning projects might require a deeper understanding of machine learning concepts and the use of traditional programming languages and frameworks.

4) Evaluation

The evaluation stage is used to test the classification value. In the teachable machine, the evaluation stage is specified in the "Preview" section. The Preview stage is shown in Figure 7.

Based on Figure 7, when the camera is given a model, the preview results show output with 71 % to 99 % data suitability confidence. From the data value of the evaluation results of the image classification model for each category made, the evaluation process shows a positive thing, where the classification value in each category shows a high enough value and the model has been able to show good performance.

5) Export model

The export stage of the model in Teachable Machine is the process of taking the trained training data and converting it into a format suitable for the microprocessor being used [29]. In this

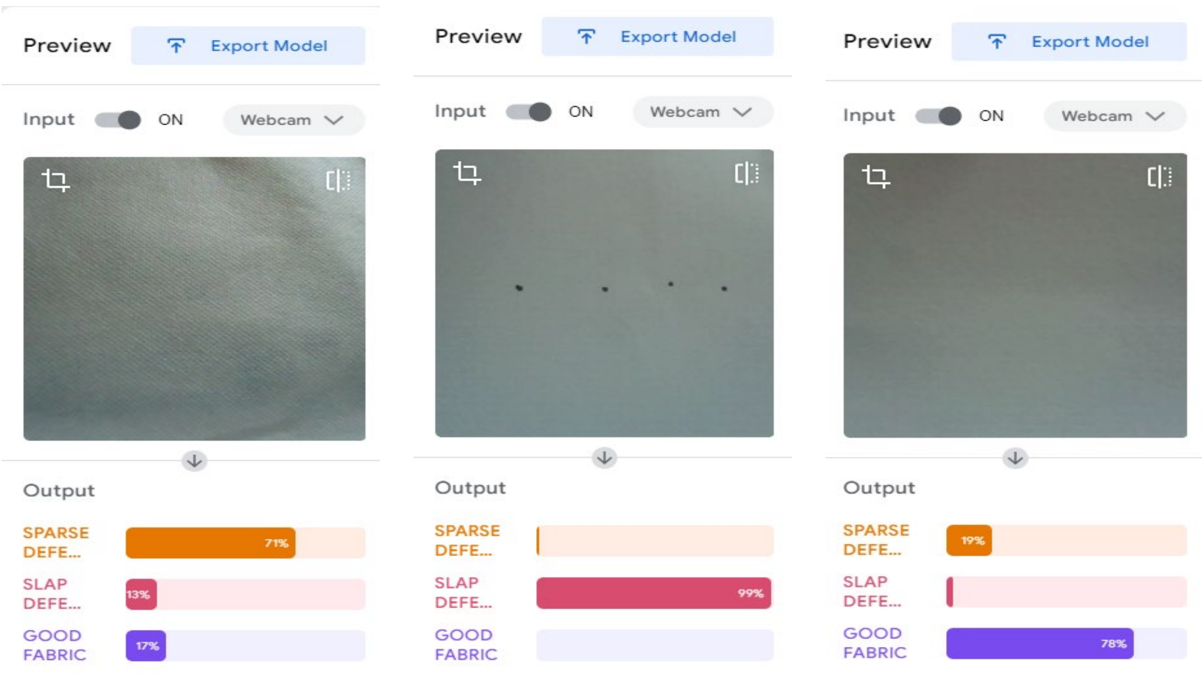

Figure 7. Initial classification result.

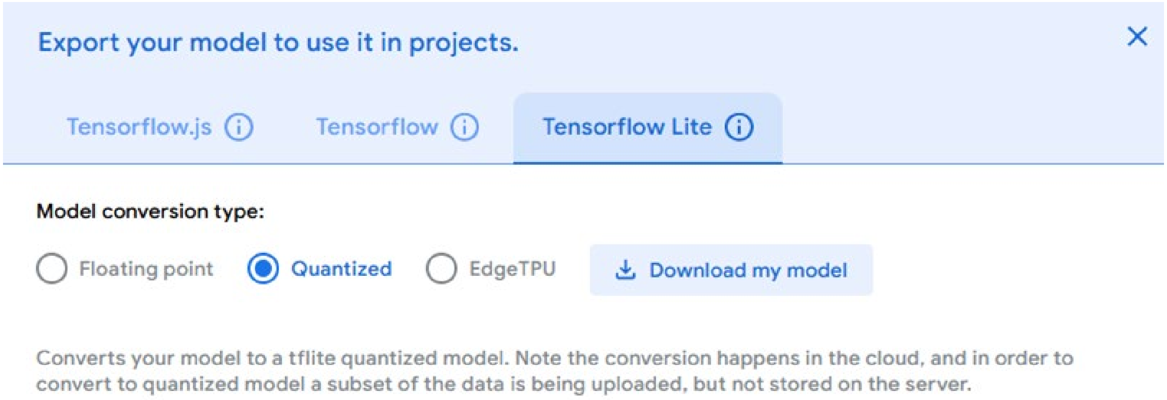

Figure 8. Export model on the teachable process.

implementation, the format used is tensor flow lite (quantized), which was chosen because the microprocessor used is a Raspberry Pi. The quantized Tensor Flow lite model is a machine learning model optimized for use on low-power devices, with model size and complexity that is more efficient, faster and suitable for real-time systems. Figure 8 shows the model export process in Teachable Machine.

Once the model has been successfully exported, as shown in Figure 8, it will produce tensorflow in 3 formats, which can be selected according to the microprocessor used. The results of the export are two files, namely files in .tflite format (model.tflite) and labels in .text format (labels.txt). The labels are in the form of category or class name information.

B. Deployment

The deployment stage is the implementation of tensor flow lite on the Raspberry Pi microprocessor using the Python application ApplicationImageClassification.py. This implementation stage is explained by the flowchart in Figure 9.

Based on Figure 9, this process starts with system initialization. This includes configuring the GPIO pins on the Raspberry Pi to control hardware such as relays and LEDs. Next, the program retrieves images from the camera source and performs the necessary pre-processing such as flipping and color conversion. The captured image from the camera is operated with a pre-initialized image classification model. The inference results of the model are used to perform classification on the image. If the classification result meets certain pre-defined conditions, the program will control the GPIO pins based on the classification result, including setting relays and LEDs according to the detected conditions. This process continues continuously, returning to the step of capturing the image from the camera after the classification and hardware control steps have been completed. If the user presses the ESC key, the program will stop and close.

III. Results and Discussions

Variables in the test data collection include frame per second (FPS) data, inference time and classification results in the image classification application. These data are important components in the retrieval of test data for image classification applications.

In the process of retrieving data from these two components, this program already has a built-in function. This function is responsible for saving all variables related to FPS and classification results in CSV format. The resulting CSV file contains information about the time and each of these variables.

- Value_fps.csv': This file is used to search for performance test data. This file is used to store frame per second (FPS) data, which reflects how fast the application is able to perform classification on each frame of the image. The information stored includes the time stamp, the time the test was performed and the FPS value at that time.
- Results_classification.csv': This file is used to search for model classification accuracy test data. This file stores the classification results obtained by the model at each iteration or frame during the application run. Each row in this file contains information such as the timestamp (the time the classification was performed), the name of the category predicted by the model, and the score of the classification result by saving this data in CSV format.

A. Performance tests

The performance test is a measurement of the operating speed of the 'Application of Image Classification', measured in units of frame per second (FPS), and testing the inference in each category. The data collected is the result of the data from the previous process in the form of a 'value_fps.csv' file. Table 2 show the data values of the inference times and fps parameters for each category. Table 2 shows the

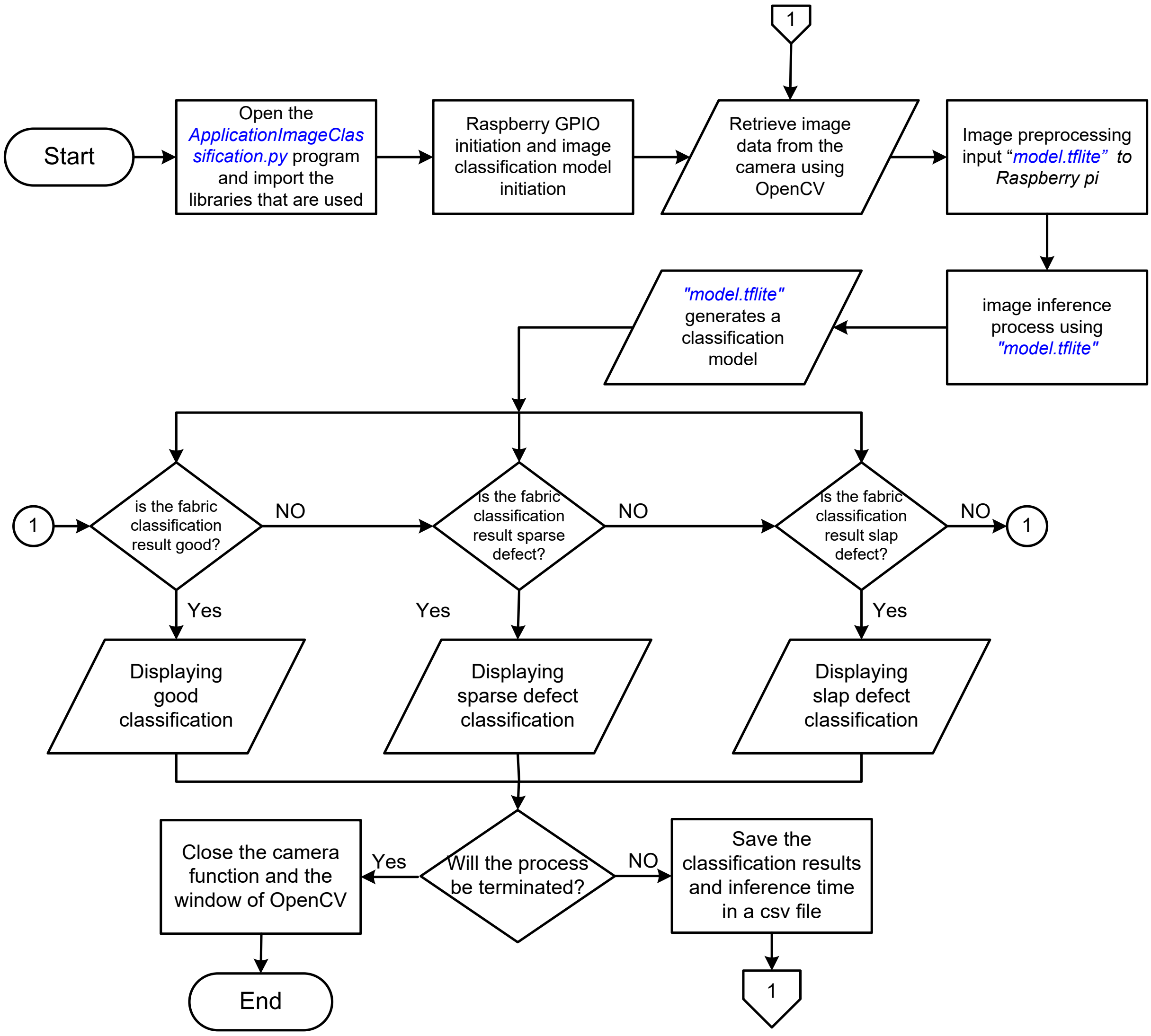

Figure 9. Flowchart of the ApplicationImageClassification.py program.

performance of the classification results by modeling a number of random samples performed on 3 categories involving ±300 read iterations for each category. The data obtained shows the inference results and FPS as shown in Table 2. Graphically Table 2 can be visualized as in Figure 10 and Figure 11.

Figure 10 displays an inference performance graph that shows Good fabric has the longest inference time while Slap defect has the fastest inference time. This means that the system is faster at recognizing slap defects than the other two categories.

Figure 11 displays the FPS performance graph that shows Good fabric and Sparse defects have the same frame reading speed and are slower when compared to Slap defects. The inference and FPS analysis for each category is explained in Figures 12, Figure 13, and Figure 14.

Figure 12 shows the inference and FPS graph for the good fabric category performed in 326 data captures. The average inference speed on good fabrics is 184.62 ms or about 4.72 frames per second. Figure 13 displays the inference and FPS graphs for the sparse defect detected fabric category performed in 309 data captures.

Table 2.
Performance results of image classification.

Category	Inference average (ms)	FPS average
Good fabric	184,62	4,72
Sparse Defect	182,21	4,72
Slap Defect	176,47	4,97
Average	181,1	4,80

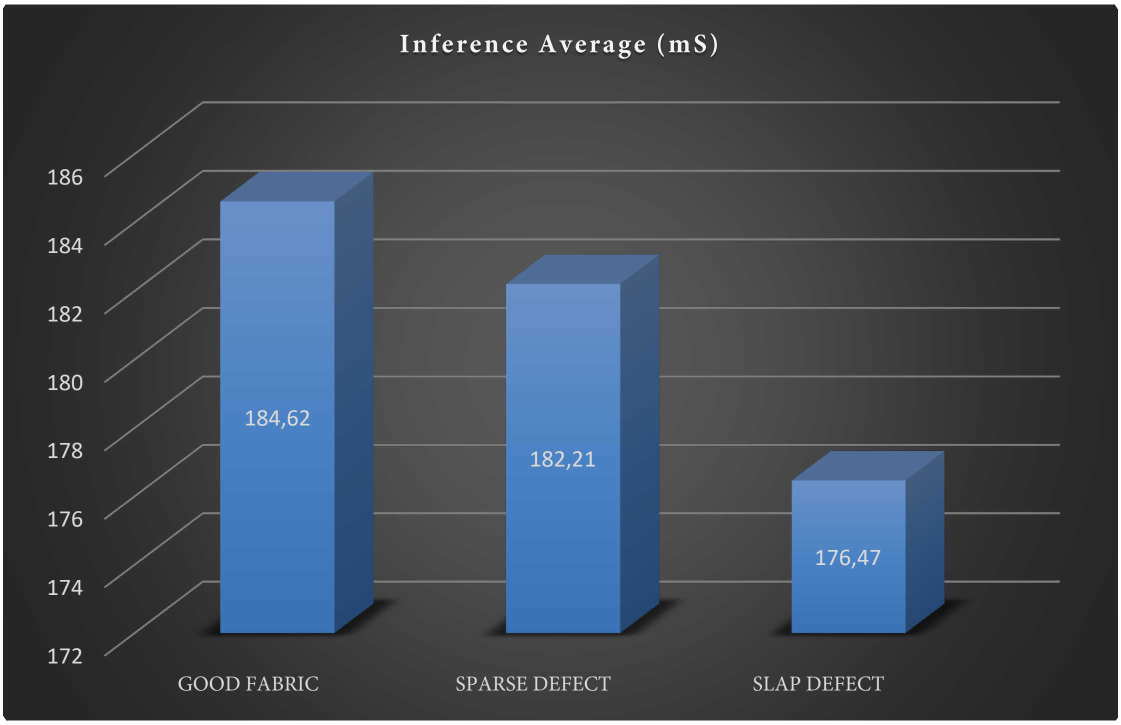

Figure 10. Inference performance results application image classification.

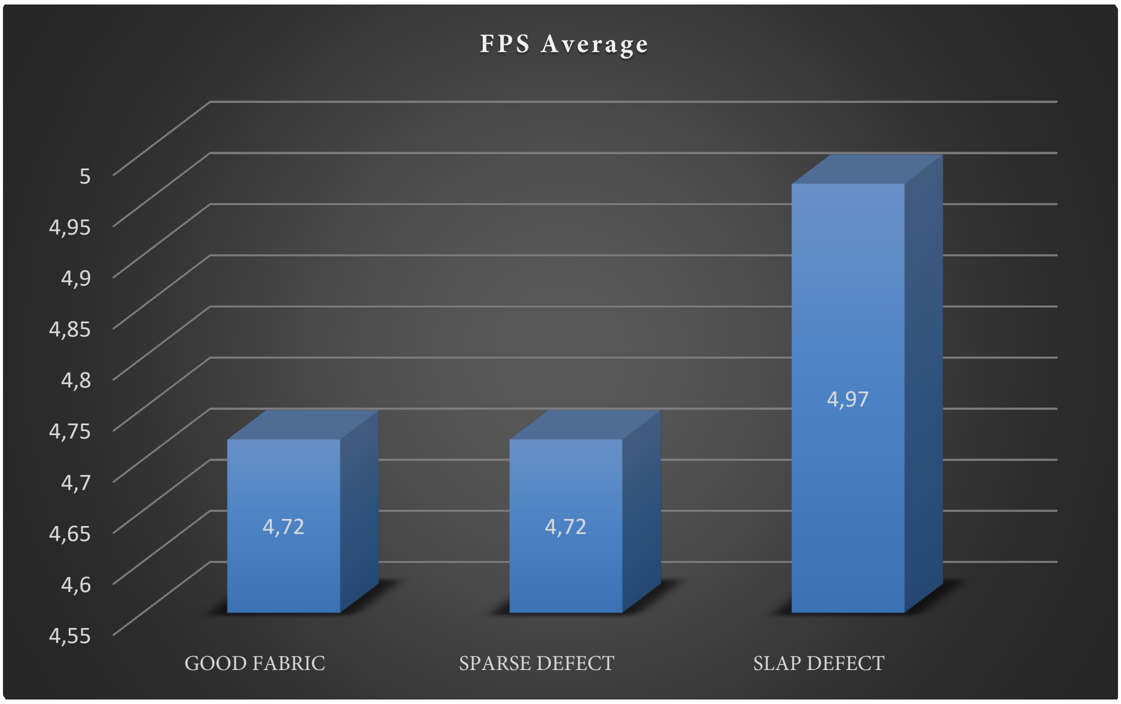

Figure 11. FPS performance results application image classification.

The average inference speed on sparse defective fabrics is 182.21 ms or about 4.72 frames per second. Figure 14 displays the inference and FPS graphs for the category of slap defect detected fabrics performed in 325 data captures. The average inference speed on slap defect fabrics is 176.47 ms or about 5.12 frames per second. From the data shown in Figure 12 to Figure 14, the classification model can produce classifications with an average inference time of 181.1 ms for all categories and an average fps of 4.80.

B. Model classification accuracy test

Model classification accuracy testing evaluates the extent to which an image classification model can correctly identify objects or attributes in an image. The results are presented in the form of tables or confusion matrix graphs to understand the model's accuracy capabilities. The data taken is the result of data retrieval in the previous process, namely in the form of the file 'Results_Klassification.csv'. Table 3 shows the data that has been taken from the CSV file.

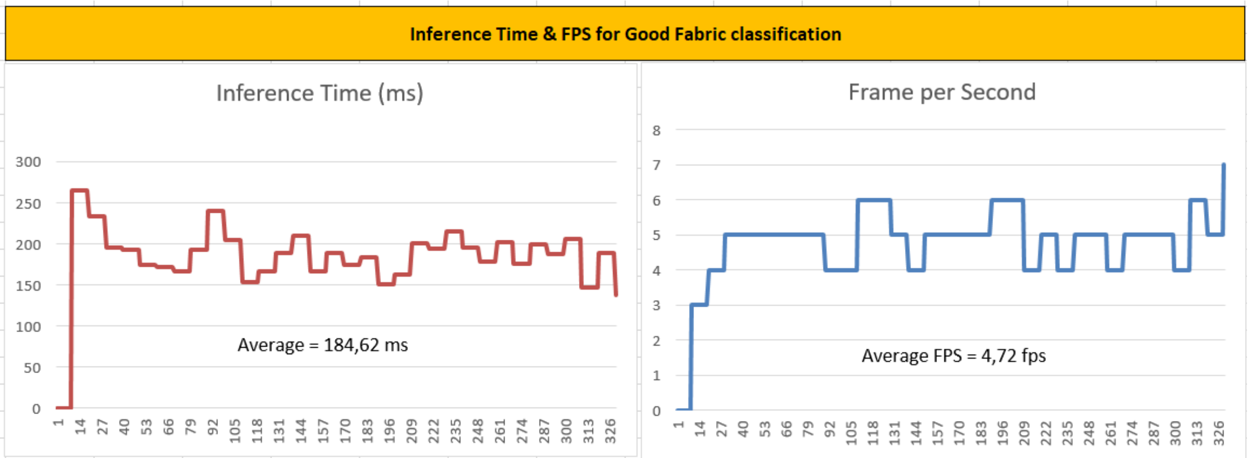

Figure 12. Graph of performance good fabric image classification application.

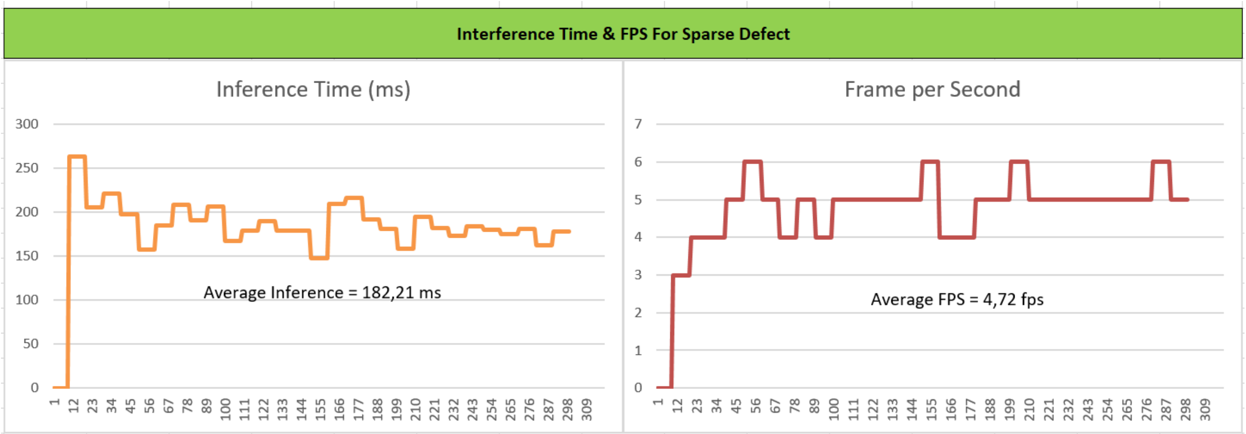

Figure 13. Graph of performance sparse defect application image classification.

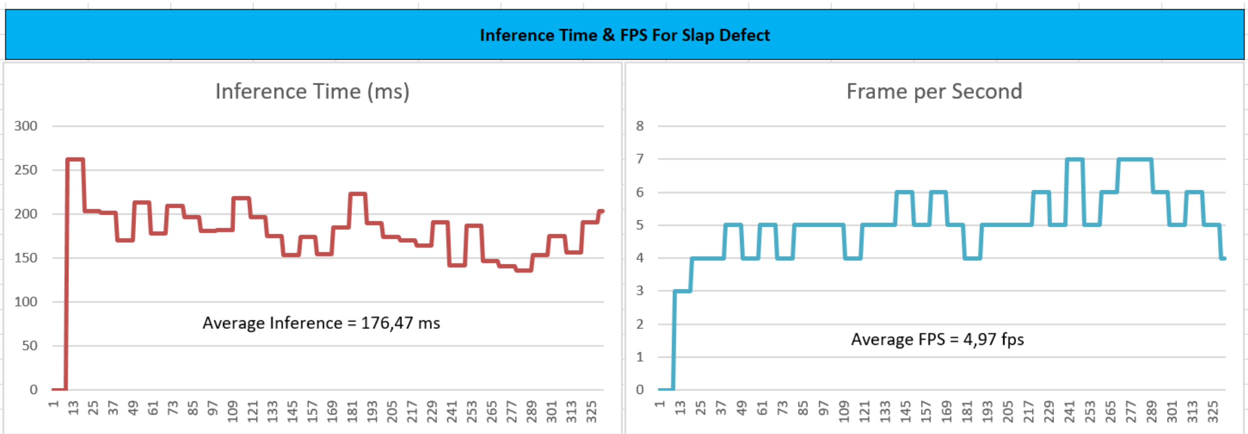

Figure 14. Graph of performance slap defect application image classification.

Table 3.
Classification results data.

Predictions	**Actual**		
	Good fabric	**Sparse defect**	**Slap defect**
Good fabric	293	0	0
Sparse defect	7	300	7
Slap defect	0	0	293

Based on Table 3, it explains that in 300 capture iterations for each category, it is obtained that in the good fabric test, 293 true positive data and 7 fals positive data are expressed as sparse defects. In the sparse defect test, 300 true positive data were read as sparse defects. In the slap defect test, 293 true positive

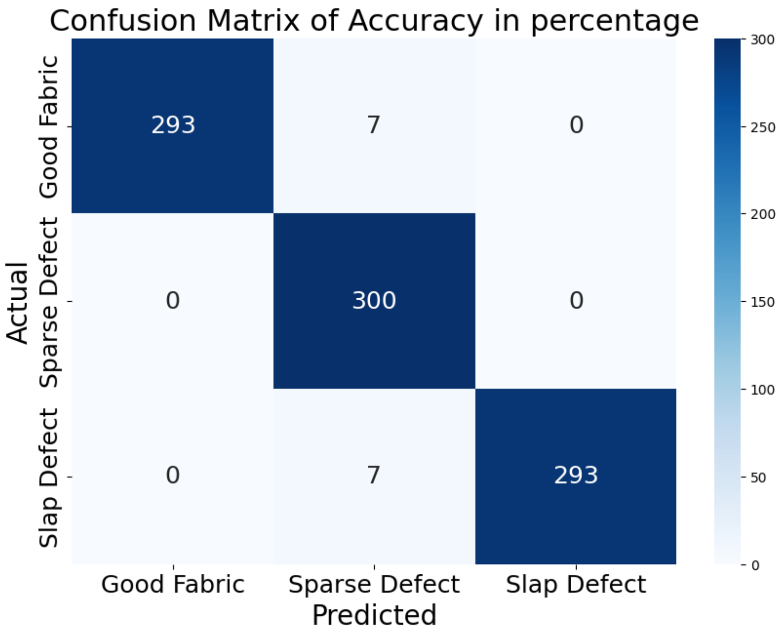

Figure 15. Confusion matrix in percentage.

data are tested as slap defects and 7 data are tested as sparse defects. Based on the data contained in Table 3, a visualization of the confusion matrix in percentage form is shown in Figure 15. Based on Figure 15, it can be concluded that the average accuracy of all model classification test results obtained an accuracy value of 98.44 %.

IV. Conclusion

The image classification system using Google Teachable Machine software and integrated with Raspberry Pi-3B in this study has been able to produce datasets in the form of tensor flow lite and is able to classify images according to samples that have been trained with an accuracy rate of up to 98.44 %. In addition to the accuracy rate, the results of this study also recorded data on the speed of the time needed to recognize objects called inference time with an average speed of 181.1 ms, with the speed of the system recognizing objects in one second of time is 4.81 frames per second (FPS). With the results of this study, the authors recommend using the same system using a type of Raspberry that has a higher speed level such as Raspberry Pi-4 and above.

Declarations

Author contribution

E.A Nugroho, J.D. Setiawan, M. Munadi, and Diki contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Additional information

Reprints and permission: information is available at https://mev.brin.go.id/.

Publisher's Note: National Research and Innovation Agency (BRIN) remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

[1] S. Tereza, "Balancing global success with local production challenges in the garment manufacturing sector, Textile Industry Triumph:," *Amberd Buletin*, pp. 38–47, 2023.

[2] I. A. Juliannisa, P. Widodo, T. W. Nunuk, "Success Deteriminan of the Manufacturing Industry Sector to Survive During the Pandemic in Indonesia," *J. Econ. Financ. Manag. Stud.*, vol. 05, no. 08, pp. 2455–2467, 2022.

[3] L. Petrie, "Sustainability and Circularity in the Textile Value Chain - A Global Roadmap Textile Value Chain," *U. N. Environment Programme,* 2023.

[4] P. N. Aurelia, A. Firdiansjah, and H. Respati, "The Application of Service Quality and Brand Image to Maintain Customer Loyalty by Utilizing Satisfaction Strategy in Insurance Companies," *Int. J. Adv. Sci. Res. Eng*., vol. 5, no. 4, pp. 223–239, 2019.
[5] İ. İlhan, "Concept of Industry 4.0 in Textile Manufacturing Processes," *Pamukkale Univ. J. Eng. Sci*., vol. 25, no. 7, pp. 810–823, 2019.
[6] J. Butt, "A strategic roadmap for the manufacturing industry to implement industry 4.0," *Designs*, vol. 4, no. 2, pp. 1–31, 2020.
[7] D. P. Mishra, S. Mishra, S. Jena, and S. R. Salkuti, "Image classification using machine learning," *Indones. J. Electr. Eng. Comput. Sci*., vol. 31, no. 3, pp. 1551–1558, 2023.
[8] S. Sha *et al*., "Image Classification and Restoration of Ancient Textiles Based on Convolutional Neural Network," *Int. J. Comput. Intell. Syst*., vol. 17, no. 1, 2024.
[9] M. Seçkin, A. Ç. Seçkin, P. Demircioglu, and I. Bogrekci, "FabricNET: A Microscopic Image Dataset of Woven Fabrics for Predicting Texture and Weaving Parameters through Machine Learning," *Sustainability*, vol. 15, no. 21, p. 15197, 2023.
[10] B. G. Batchelor, "Machine Vision for Industrial Applications BT - Machine Vision Handbook," B. G. Batchelor, Ed., London: Springer London, 2012, pp. 1–59.
[11] J. Xu, Y. Wei, A. Wang, H. Zhao, and D. Lefloch, "Analysis of Clothing Image Classification Models: A Comparison Study between Traditional Machine Learning and Deep Learning Models," *Fibres Text. East. Eur*., vol. 30, no. 5, pp. 66–78, 2022.
[12] K. Wisnudhanti and F. Candra, "Image Classification of Pandawa Figures Using Convolutional Neural Network on Raspberry Pi 4," *J. Phys. Conf. Ser*., vol. 1655, no. 1, 2020.
[13] E. Pintelas, M. Liaskos, I. E. Livieris, S. Kotsiantis, and P. Pintelas, "Explainable machine learning framework for image classification problems: Case study on glioma cancer prediction," *J. Imaging*, vol. 6, no. 6, 2020.
[14] H. Alaskar and T. Saba, "Machine Learning and Deep Learning: A Comparative Review," *Prince Sultan University, Researchgate*, June, pp. 143–150, 2021.
[15] V. Radojcic and P. Spalevic, "Advancements in Computer Vision Applications for Traffic Surveillance Systems," *Sinergija University Scientific Conference*, February, pp. 10–15, 2023.
[16] R. Sathya, A. Asha, K. Muralikrishna, and V. Bakyalakshmi, "Bespoke Measurement based on Convolutional Neural Network using OpenCV Bespoke Measurement based on Convolutional Neural Network using OpenCV," *Ijatcse*, vol. 9, no. 3, May, 2020.
[17] Ibnu. Daqiqil. Id, (2021, Juli) "Machine Learning : Teori , Studi Kasus dan Implementasi Menggunakan Python," Riau, UR PRESS.
[18] Bonaccorso. G.,(2017, Juli), "*Machine Learning Algorithms*." Birmingham: Packt Publishing.Ltd.
[19] Ş. Çiğdem, I. Meidute-Kavaliauskiene, and B. Yıldız, "Industry 4.0 and Industrial Robots: A Study from the Perspective of Manufacturing Company Employees," *Logistics*, vol. 7, no. 1, 2023.
[20] V. Wiley and T. Lucas, "Computer Vision and Image Processing: A Paper Review," *Int. J. Artif. Intell. Res*., vol. 2, no. 1, p. 22, 2018.
[21] R. V. Sharan and G. C. Onwubolu, "Simulating the Arm Movements of a Stepper Motor Controlled Pick-and-Place Robot Using the Stepper Motor Model," *Int. J. Adv. Sci. Technol*., vol. 60, pp. 59–66, 2013.
[22] E. Ariesto, U. Malahina, R. P. Hadjon, and F. Y. Bisilisin, "Teachable Machine : Real-Time Attendance of Students Based on Open Source System," *Ijics*, vol. 6, no. 3, pp. 140–146, 2022.
[23] E. A. Nugroho, J. D. Setiawan, and M. Munadi, "Handling Four DOF Robot to Move Objects Based on Color and Weight using Fuzzy Logic Control," *jrc*, vol. 4, no. 6, pp. 769–779, 2023.
[24] P. S. Tsai, T. F. Wu, J. Y. Chen, and F. H. Lee, "Drawing system with dobot magician manipulator based on image processing," *Machines*, vol. 9, no. 12, Dec. 2021.
[25] H. Al-Jubouri, H. Du, and H. Sellahewa, "Adaptive clustering based segmentation for image classification," *2013 5th Comput. Sci. Electron. Eng. Conf. CEEC 2013 - Conf. Proc*., April, pp. 128–133, 2013.
[26] K. Azmi, S. Defit, and S. Sumijan, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat," *J. Unitek*, vol. 16, no. 1, pp. 28–40, 2023.
[27] P. M. Radiuk, "Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets," *Inf. Technol. Manag. Sci*., vol. 20, no. 1, pp. 20–24, 2018.
[28] N. Rochmawati, H. B. Hidayati, Y. Yamasari, H. P. A. Tjahyaningtijas, W. Yustanti, and A. Prihanto, "Analisa Learning Rate dan Batch Size pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam," *J. Inf. Eng. Educ. Technol*., vol. 5, no. 2, pp. 44–48, 2021.
[29] E. A. U. Malahina, M. Saitakela, S. J. Bulan, M. I. J. Lamabelawa, and Y. S. Belutowe, "Teachable Machine: Optimization of Herbal Plant Image Classification Based on Epoch Value, Batch Size and Learning Rate," *J. Appl. Data Sci*., vol. 5, no. 2, pp. 532–545, 2024.