Jurnal Polimesin
Department of Mechanical Engineering
State Polytechnic of Lhokseumawe
http://e-jurnal.pnl.ac.id/polimesin
e-ISSN : 2549-1999 No. : 5 Month : October
p-ISSN : 1693-5462 Volume : 22 Year : 2024

Article Processing Dates: Received on 2024-05-08, Reviewed on 2024-09-17, Revised on 2024-09-28, Accepted on 2024-10-01 and Available online on 2024-10-30

Enhancing textile quality control with the application of teachable machine and Raspberry Pi as machine learning-based image processing

Emmanuel Agung Nugroho[1,2*], Joga Dharma Setiawan[2], M. Munadi[2], M. Diki[1]
[1]Department of Mechatronics Engineering, Indorama Engineering Polytechnic, Purwakarta, 41152, Indonesia
[2]Department of Mechanical Engineering, Diponegoro University, Semarang, 50275, Indonesia
*Corresponding author: emmanuel.agung@pei.ac.id

Abstract

The adoption of image processing-based technologies in the textile sector is rising. This technology is commonly utilized to replace traditional sensor systems that are limited to a single function while also improving product quality control functions. Defects during the manufacturing process are a common problem in the textile business, particularly with fabric products. This study created a fabric quality control system that detects fabric problems using machine learning-based picture classification techniques. A D320p web camera detects rare and slap flaws, which are classified using open-source Google teaching machine software and processed on a Raspberry Pi 3B device. The laboratory-scale measurement was carried out on a prototype cloth rolling machine using the confusion matrix method. The test results reveal an average inference speed of 143.5 milliseconds, a frame rate of 6.45 fps, and a 98.56% accuracy rate. These results demonstrate that the proposed system is effective and efficient for detecting fabric defects, offering a promising solution for enhancing quality control in the textile industry. Future research could focus on scaling the system for industrial use and enhancing real-time performance.

Keywords:
Machine learning, image classification, Google Teachable Machine, Raspberry Pi-4, confusion matrix.

1 Introduction

According to the official website of the Ministry of Industry of the Republic of Indonesia on Thursday, 14 July 2022, the Minister of Industry said that the Textile and Textile Products (TPT) industry has a strategic role in national development and contributed to absorbing a workforce of 3.65 million people as of August 2021. The rapid growth of the textile industry is a problem in itself with its challenges in the form of increasing the competitiveness of product quality to meet customer satisfaction. According to Crosby, quality is conformity to needs, which include availability, delivery, reliability, maintainability, and cost effectiveness [1]. Production defects are one of the factors that affect the quality of a product. In fabric production, there are types of defects that often occur, namely, double warp, double pick, uneven thickness, slap, rare, inappropriate fabric color, and missed patterns.

This research aims to realize a tool that can be used to detect fabric defects as an early warning inspection so that the recommendations can be used to determine product quality factors. This tool is placed on the fabric rolling machine before the fabric is distributed to consumers with 2 types of defects detected in this study, namely rare defects and slap defects. To carry out its function as a fabric defect detector, modeling of fabric conditions is carried out using Google Teachable Machine software. Image processing for classification function using Raspberry Pi 3B. The system used for the fabric inspection process in this study uses the image classification method. Image classification is one part of digital image processing that belongs to the machine learning family. It is called part of machine learning because in the image classification system there are the stages, namely the supervised learning process, the feature extraction process, the process of collecting machine learning algorithms, improving performance by using additional data, and the model adaptation process marked by variations in training data [2], [3]. This research narrows its focus to rare defects and slap defects using Google Teachable Machine software for machine learning based image classification and Raspberry Pi 3B for image processing. The novelty lies in the integration of an accessible, low-cost system aimed at providing early warning inspection for fabric defects directly on fabric rolling machines, a placement that could significantly impact quality control before products are distributed to consumers.

1.1 Machine Learning

Machine learning as a branch of artificial intelligence plays a role in the development of pattern identification systems and digital image processing. In an image classification system, machine learning has the role of recognizing and categorizing images into classes or categories based on extracted features that are determined on the training data of an image [4]. Machine learning techniques that can be applied in image processing systems include:

1. Convolutional Neural Networks (CNNs). CNNs are a type of neural network architecture that is highly effective for image processing tasks. They are able to extract hierarchical features from images using layers of convolution and pooling [5], [6].
2. Transfer learning. Transfer learning involves using a model that has been previously trained on a large dataset (e.g., a model trained on the ImageNet dataset) and adapting it to a smaller or specific dataset. This allows using the information learned by the existing model for a new image classification task [7].
3. Data augmentation. Data augmentation involves generating additional variations of the training image by applying transformations such as rotation, cropping, shifting, and others. This helps to increase the diversity of the dataset and reduce over fitting [8].
4. Feature extraction. This technique involves extracting features from images using methods such as manual feature extraction or using pre-trained models, and then using those features as input to a simple classification model [9].

By combining these techniques, machine learning models can be built and trained to perform image classification with high accuracy in various applications, such as object recognition, object detection, and object classification [10]. Fig. 1 is an example of a machine learning application algorithm for image classification.

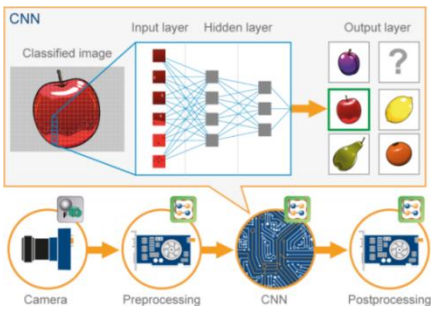

Fig. 1. CNN algorithm illustration in image classification application [11].

Fig. 1 provides an overview of the CNN algorithm for image classification applications equipped with the equipment needed for image classification processes such as cameras, preprocessing tools, devices for processing CNN algorithms and means for displaying algorithm results at the post processing stage. In this research, the object detected is the type of fabric, the preprocessing tool uses Google Teachable Machine, CNN algorithm and displays the output using Raspberry Pi 3B.

1.2 Image Classification

Image classification is the process of categorizing or grouping the digital image into predefined classes or categories based on the visual characteristics contained in the image. The main objective of image classification is to identify or predict the class or label corresponding to a given image [12]. Generally, there are two digital image classification systems, namely supervised learning and unsupervised learning [13]. Supervised learning is one of the paradigms in machine learning where models learn from examples that are already labeled. In supervised learning, the training dataset consists of known input and output pairs (labels) [14]. There are two main types of supervised learning:

1. Regression. Regression is to learn the relationship between continuous inputs and outputs, so that the model can predict new outputs for inputs that have not been seen before [15], [16].
2. Classification. In classification, the predicted output is a discrete class or label. A model is used to map each input to one of the predefined classes or labels. Examples of classification include image recognition, text classification, and email spam detection [17].

Unsupervised learning is a paradigm in machine learning where models are learnt from data that has no labels or pre-structured information. The main goal of unsupervised learning is to discover patterns or structures hidden in data without the help of labels [18]. There are two main tasks in unsupervised learning:

1. Clustering: tasked to separating data into distinct groups without any labels. Examples of clustering tasks include customer segmentation based on purchasing behavior, news clustering based on topics, and image clustering based on visuals [19].
2. Dimensionality reduction: tasked to reduce the number of dimensions or features in the dataset, while retaining as much relevant information as possible [20].

2 Research Methods

The method used in this research is supervised learning with training data in the form of fabrics classified into the categories of good fabrics, rare defective fabrics and slap defective fabrics. The software used is Google Teachable Machine tool to build pre-processing, and open CV to build CNN algorithms. While the hardware used is a Logitech D320 web camera to capture objects, and Raspberry Pi 3B in post processing training by utilizing GPIO to provide indicators and warnings of fabric classification results. The process, image classification diagram in this research is expressed in Fig. 2.

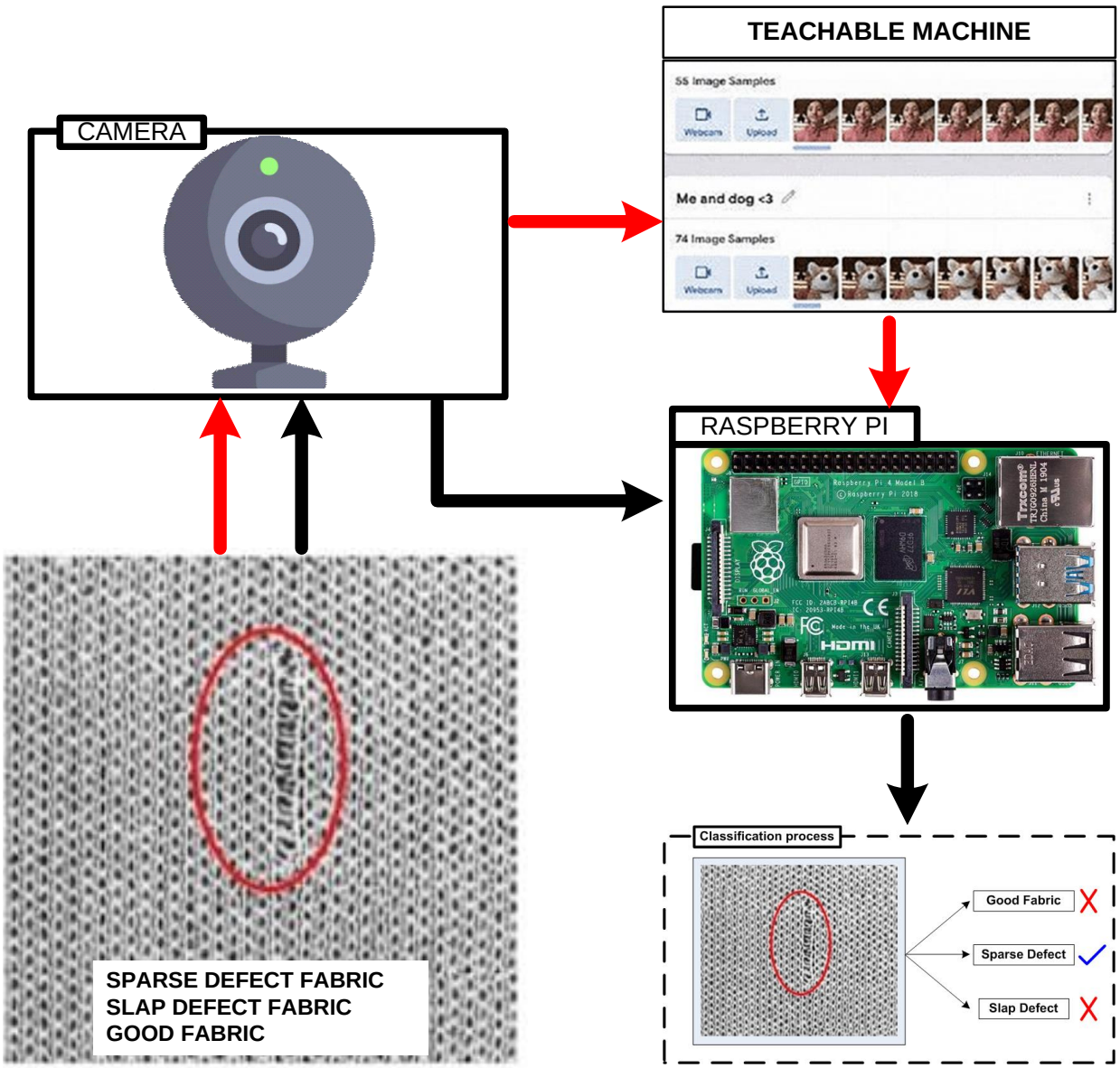

Fig. 2. Process diagram of fabric classification system.

Fig. 2 shows the process of the fabric classification system using Teachable Machine and Raspberry Pi.

Red dashes: these indicate the flow of the training and labeling process, beginning from the initial data capture to model training and XML export. On this process the fabric is sample data read by the camera and stored in the Teachable Machine according to the classification that has been made. The sample data is then stored in the Raspberry Pi as a training data library.

Black dashes: these represent the real-time model identification process, where the trained model is used to classify fabrics during production on the black process, the fabric is a model that will be read by the camera and executed by machine learning embedded in the Raspberry Pi to determine the classification results of the model.

The Teachable Machine algorithm is shown in Fig. 3.

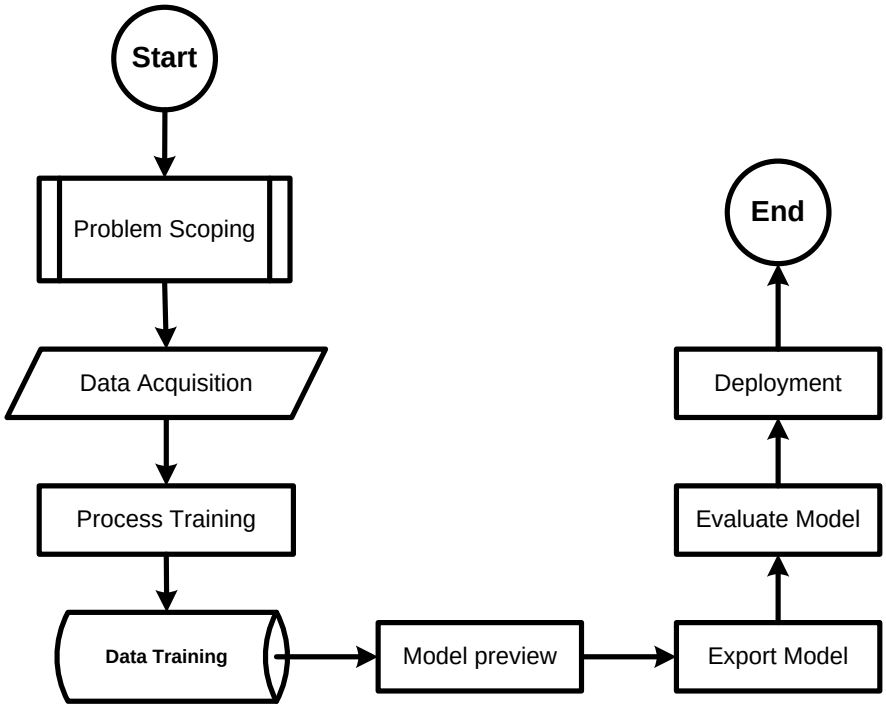

Fig. 3. Algorithm of fabric detection system with Teachable Machine.

Through Fig. 3, it can also be explained that there are 2 mechanisms for the image classification process in this study in the first stage, namely preprocessing which consists of the data acquisition process, training process, preview, model export, and testing. In the second stage, namely post processing which consists of the deployment process and GPIO addressing.

2.1 Data Acquisition

At this stage, the object dataset that will be used to train the model is collected. Each object is labeled according to its category. In this research, data collection is done by taking samples of various fabric conditions captured with a camera and stored in the Teachable Machine according to the category. The categories created are good fabric, rare defects, and slap defects. Table 1 shows the description of the amount of training data.

Table 1. The dataset samples

Category	Number of sample
Good fabric	251
Rare defect	325
Stain defect	204

Table 1 shows that there are 3 categories of samples from several models sampled: 251 good fabric samples, 325 rare defects samples, and 204 slap/stain defects samples. Sample collection was carried out using Teachable Machine tools and a Logitech D320p camera.

2.2 Training Process

At this stage, the machine learning model is trained using the collected dataset. This involves using Machine Learning algorithms, such as Convolution Neural Networks, to adjust the model parameters to recognize the desired objects or categories. In the training process, there are important terms that become parameters, namely:

1. Epochs: the number of times the entire training dataset is used in training the model. Determines how many iterations are required during training [21].
2. Batch size: the number of data instances processed in a single training iteration. Affects training speed and computational resource utilization [22], [23].
3. 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 [24].

Fig. 4 shows the training process on Teachable Machine.

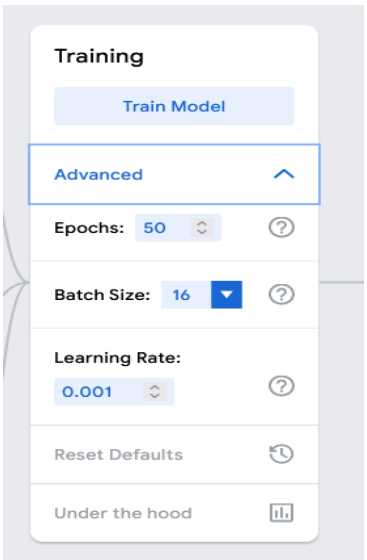

Fig. 4. Training model Teachable Machine.

Fig. 4 displays the training process carried out on fabric defect sampling, there are 50 epochs, 16 batch sizes and a learning rate of 0.001.

2.3 Preview

This stage involves developing the interface or appearance of the application that utilizes the trained model. This may include user interface design, model integration, and other settings for the application to interact with the model effectively. In the Teachable Machine implementation, this stage is called the preview stage, which displays the conformity between the model and the training data in a certain percentage. Fig. 5 shows the preview results of the model that has been trained.

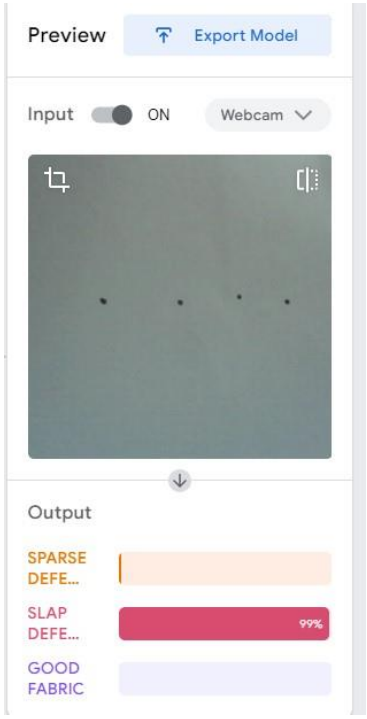

Fig. 5. Preview model Teachable Machine.

Fig. 5 shows an example of the preview process, the fit of the model to the training data on the test defects are rarely detected with a fairly high classification rate of 99%.

2.4 Export

The next step is export model. In Teachable Machine export model is the process of taking the trained machine learning model and generating it in a format that can be used in various applications. In this case, export the model into the tensor flow lite format with the quantization method. Tensor flow lite with quantized is a machine learning model that has been optimized for use on limited power devices. Quantization is a technique that reduces the size and complexity of a model by changing its numerical representation to be more efficient, which allows the model to run faster on devices such as Raspberry Pi. This kind of model is suitable for real-time applications that require fast response. Fig. 6 shows how to export the model on Teachable Machine.

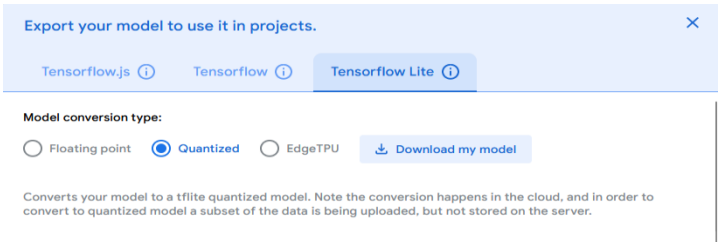

Fig. 6. Export model Teachable Machine.

After the model is successfully exported, the results of the export are in the form of two files, namely files in *.tflite* format (*model.tflite*) and labels in .text format (*labels.txt*). Labels are in the form of category or class name information.

2.5 Testing

The purpose of model testing is to ensure that the model provides accurate predictions and that the application works properly. Testing includes functionality tests and verification that the model provides expected results on new data. The Evaluation stage is to evaluate the performance of the machine learning model by testing the model on data that has never been seen before. The Evaluation stage used is accuracy evaluation. To calculate accuracy using the Eq. 1.

$$Accuracy = \frac{number\ of\ correct\ trials}{Total\ trials} \times 100\% \qquad (1)$$

2.6 Deployment

This is the mechanism of embedding the tensor flow into the Raspberry Pi device using the ApplicationImageClassification.py application. To explain the deployment process into the Raspberry Pi 3B, the program algorithm is made as shown in Fig. 7.

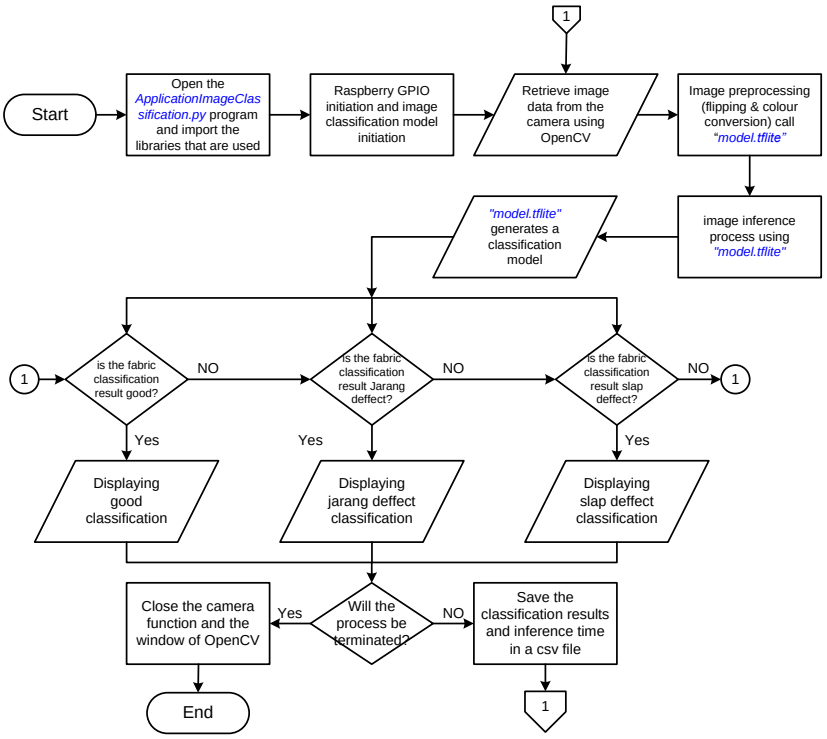

Fig. 7. Deployment algorithm of image classification programming with Raspberry Pi.

Fig. 7 explains in detail the stages of deploying the image classification program into the Raspberry Pi microprocessor using the python programming language. This process begins with system initialization, and configuration of GPIO pins on the Raspberry Pi. Next, the program takes an image from the camera source, and performs the necessary pre-processing operations such as flipping and color conversion. The image obtained from the camera capture is operated with a pre-initialized image classification model. The inference results of the model are used to perform classification of the image.

When the classification result is satisfied with the predefined conditions, the program will control the GPIO pins based on the classification result. This process continues continuously, returning to the step of capturing the image from the camera after completing the classification and hardware control steps. When the user presses the ESC key, the program stops and closes the program. The visualization of the Raspberry Pi working system flow chart is described in the hardware block diagram in Fig. 8.

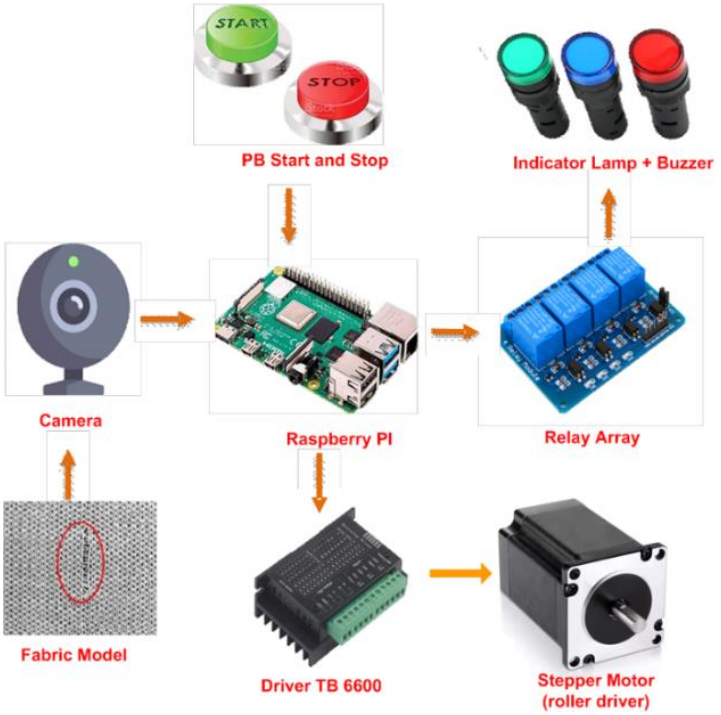

Fig. 8. Hardware visualization of image classification systems.

Fig. 8 explains the communication process between the hardware used in the image classification application for fabric defect detection. The camera takes a sample image of the fabric that passes within its range, then the image is sent to the Raspberry Pi via the USB port to be processed with the CNN algorithm. Image data processing uses the "Application of Image Classification" program with the output in the form of fabric classification data. Based on the classification result data, the program will control the Raspberry Pi GPIO pin to carry out the program execution function, namely stopping the rotation of the rolling motor turning on the fabric defect type indicator and activating the buzzer. The start button is used to reactivate the fabric rolling process by activating the stepper motor and the classification system is active again. The stop button is to temporarily stop the process that is running.

3 Results and Discussion

3.1 Application Image Classification Result

In this study, there are three main variables observed, which are Frame Per Second (FPS) data, inference time, and accuracy for each deployment process with the ApplicationImageClassification.py application.

Fig. 9 is ApplicationImageClassification.py. view. In this view, there is some important information presented:

1. Frames Per Second (FPS) data, states the speed at which the camera device is able to capture image frames in one second.
2. Inference data states the time required to perform the inference process or the time required to predict or classify the data it reads.
3. Classification result is the result of the camera reading that states the type of classification of the object model it reads.

Classification value is the confidence level or accuracy level of the system against the given classification result.

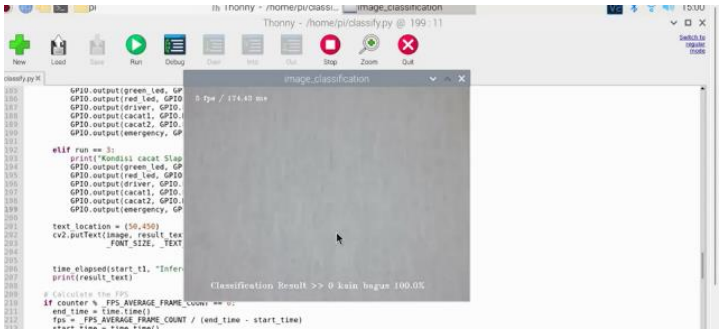

Fig. 9. ApplicationImageClassification.py.

3.2 Image Classification Application Performance Testing

Image classification application performance testing includes measuring the speed of the classification process in the inference times parameter, the speed of the camera reading each frame of the Frame Per Second (FPS) unit, and the accuracy of reading the classification results. The values of the inference times and fps parameters for each category are shown in Table 2.

Table 2. Classification process speed test data

Category	FPS average	Inference average (ms)	Number of frames/minute
Good fabric	7.44	126.69	351
Rare defects	5.98	150.87	365
Slap defect	5.93	152.93	365

Table 2 states the FPS speed and inference speed of the classification process for 3 types of models, namely good fabrics, rare defective fabrics, and slap defective fabrics. Based on Table 2, the performance graph of the ApplicationImageClassification.py program can also be made.

Fig. 10 presents the inference time and Frames Per Second (FPS) graph for the good fabric category, based on a total of 352 data captures. The analysis reveals that the average inference time for good fabrics is approximately 126.69 milliseconds, which translates to a processing rate of around 7.44 frames per second.

Fig. 10. Graph of application image classification performance for good fabrics.

Fig. 11 illustrates the inference time and Frames Per Second (FPS) graphs for fabrics with sparse defects, using data from 365 captures. On average, the inference time for these fabrics is approximately 150.87 milliseconds, corresponding to a processing speed of about 5.98 frames per second.

Fig. 12 presents the inference time and Frames Per Second (FPS) graphs for fabrics with slap defects, based on 365 data captures. The average inference time for these fabrics is 152.93 milliseconds, yielding a processing rate of approximately 5.93

frames per second. From these findings, it can be concluded that the image classification application achieves an overall average inference time of 143.5 milliseconds across all categories, with an average processing speed of 6.45 FPS.

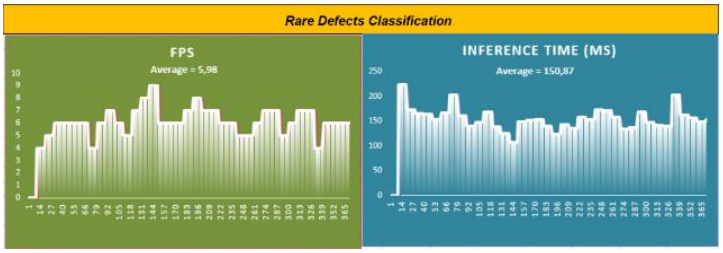

Fig. 11. Graph of application image classification performance for rare defects.

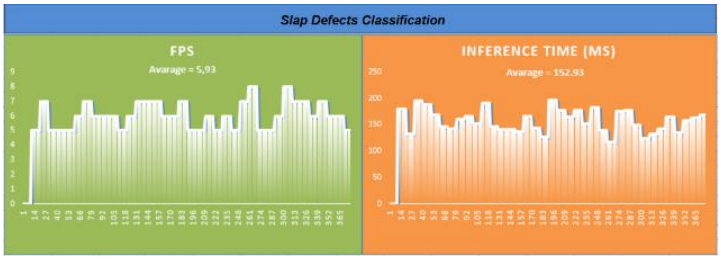

Fig. 12. Graph of application image classification performance for slap defects.

3.3 Model Classification Accuracy Testing

Accuracy testing to evaluate how well the model or system classifies images into the right class or label. Accuracy test data is obtained from the file 'results_classification.csv'. Table 3 shows the system accuracy prediction data that has been taken from the csv file.

Table 3. Data accuracy result of fabric classification system

Predicted	Inference time	Actual		
		Good fabric	Rare defects	Slap defects
Good fabric	126.69	318	0	0
Rare defects	150.87	7	325	7
Slap defects	152.93	0	0	318

Based on Table 3, it explains that in the good fabric test, namely for 126.69 ms, 325 inference results are read with 318 stating as good fabrics and 7 stating as rare defects. In the rare defect test, namely for 150.87 ms, 325 inference results were read with 325 stating as rare defects. In the slap defect test, namely for 152.93 ms, 325 inference results were read with 318 stating as slap defects and 7 stating as rare defects. Visualization of the confusion matrix in percentage form based on the data contained in Table 3 is shown in Fig. 13.

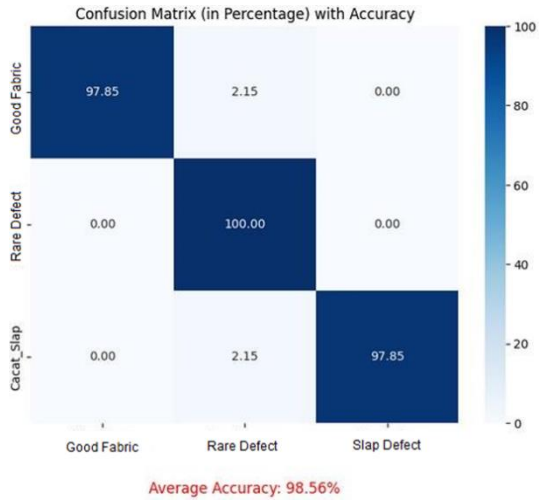

Fig. 13. Confusion matrix fabric classification accuracy.

Based on the results in Fig. 13, the average accuracy of all model classification test results is 98.56%. This is better than the previous research that resulted in an accuracy of 98.44% [25].

4 Conclusion

This research developed a fabric defect detection system with the implementation of the image classification method demonstrated high performance, with an average inference speed of 143.49 milliseconds and a frame rate of 6.45 fps. The system achieved an accuracy rate of 98.56% in detecting fabric defects, demonstrating its effectiveness in real-world applications. Analysis revealed a correlation between inference time and frame rate, where an increase in inference time resulted in a lower frame rate. Therefore, it is promising to create a practical early-warning inspection tool for fabric defects. The system's ability to classify defects accurately and operate efficiently makes it a valuable asset for quality control in the textile industry. Future improvements could focus on enhancing the system's adaptability to various lighting conditions and further optimizing the processing speed.

References

[1] S. Kasus, P. T. Iskandar, I. Printing, D. F. Dewanti, and D. Pujotomo, “Analisis Penyebab Cacat Produk Kain Dengan Menggunakan Metode Failure Mode And Effect Analysis (FMEA)”.

[2] S. I. Kampezidou, A. T. Ray, A. P. Bhat, O. J. P. Fischer, and D. N. Mavris, “Fundamental Components and Principles of Supervised Machine Learning Workflows with Numerical and Categorical Data,” pp. 384–416, 2024.

[3] I. D. Id, “Machine Learning : Teori , Studi Kasus dan Implementasi Menggunakan Python,” no. July, 2021, doi: 10.5281/zenodo.5113507.

[4] M. Nurhadi and J. Purnomo, “Implementation of Image Classification Using Convolutional Neural Network (Cnn) Algorithm on Vehicles Images,” *ASEAN J. Syst. Eng.*, vol. 6, no. 1, pp. 1–5, 2022, doi: 10.22146/ajse.v6i1.72411.

[5] 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, doi: 10.52072/unitek.v16i1.504.

[6] C. R. Kotta, D. Paseru, and M. Sumampouw, “Implementasi Metode Convolutional Neural Network untuk Mendeteksi Penyakit Pada Citra Daun Tomat,” *J. Pekommas*, vol. 7, no. 2, pp. 123–132, 2022, doi: 10.56873/jpkm.v7i2.4961.

[7] Buyut Khoirul Umri and V. Delica, “Penerapan transfer learning pada convolutional neural networks dalam deteksi covid-19.,” *Jnanaloka*, pp. 9–17, 2021, doi: 10.36802/jnanaloka.2021.v2-no2-9-17.

[8] J. Sanjaya and M. Ayub, “Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup,” *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 2, pp. 311–323, 2020, doi: 10.28932/jutisi.v6i2.2688.

[9] Y. Religia, “Feature Extraction Untuk Klasifikasi Pengenalan Wajah Menggunakan Support Vector Machine Dan K-Nearest Neighbor,” *Pelita Teknol. J. Ilm. Inform. Arsit. dan Lingkung.*, vol. 14, no. 2, pp. 85–92, 2019.

[10] N. Handa, Y. Kaushik, N. Sharma, M. Dixit, and M. Garg, “Image Classification Using Convolutional Neural Networks,” *Commun. Comput. Inf. Sci.*, vol. 1393, no. December 2022, pp. 510–517, 2021, doi: 10.1007/978-981-16-3660-8_48.

[11] D. Jaswal, S. V, and K. P. Soman, “Image Classification Using Convolutional Neural Networks,” *Int. J. Sci. Eng. Res.*, vol. 5, no. 6, pp. 1661–1668, 2014, doi: 10.14299/ijser.2014.06.002.

[12] K. Diantoro, B. Adriasyah, C. Integral, and P. C. Awal, “Sistem Identifikasi Jenis Burung Dengan Image,” vol. 20, no. 1, pp. 96–105, 2019.

[13] Y. Boussemart, M. L. Cummings, J. Las Fargeas, and N. Roy, “Supervised vs unsupervised learning for operator state modeling in unmanned vehicle settings,” *J. Aerosp. Comput. Inf. Commun.*, vol. 8, no. 3, pp. 71–85, 2011, doi: 10.2514/1.46767.
[14] P. Cheng, B. Chien, and W. Yang, “Medical Image Classification by Supervised Machine Learning,” no. June, 2014.
[15] A. Lindholm, N. Wahlström, F. Lindsten, and T. B. Schön, “Supervised Machine Learning: Statistical Machine Learning course,” p. 112, 2019, [Online]. Available: http://www.it.uu.se/edu/course/homepage/sml/literature/lecture_notes.pdf
[16] A. N. Angelopoulos *et al.*, “Image-to-Image Regression with Distribution-Free Uncertainty Quantification and Applications in Imaging,” *Proc. Mach. Learn. Res.*, vol. 162, pp. 717–730, 2022.
[17] 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, doi: 10.11591/ijeecs.v31.i3.pp1551-1558.
[18] A. Olaode, G. Naghdy, and C. Todd, “Unsupervised classification of images: a review,” *Int. J. Image Process.*, no. 8.5, pp. 325–342, 2014, [Online]. Available: https://www.researchgate.net/profile/Abass-Olaode/publication/265729668_Unsupervised_Classification_of_Images_A_Review/links/541a74be0cf203f155ae295a/Unsupervised-Classification-of-Images-A-Review.pdf
[19] J. Goldberger, S. Gordon, and H. Greenspan, “Unsupervised image-set clustering using an information theoretic framework,” *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 449–458, 2006, doi: 10.1109/TIP.2005.860593.
[20] M. Dash, H. Liu, and J. Yao, “Dimensionality reduction of unsupervised data,” *Proc. Int. Conf. Tools with Artif. Intell.*, no. April, pp. 532–539, 1997, doi: 10.1109/tai.1997.632300.
[21] S. Afaq and S. Rao, “Significance Of Epochs On Training A Neural Network,” *Int. J. Sci. Technol. Res.*, vol. 9, no. 06, pp. 1–4, 2020, [Online]. Available: www.ijstr.org
[22] A. Ali, R. Pinciroli, F. Yan, and E. Smirni, “Batch: Machine learning inference serving on serverless platforms with adaptive batching,” *Int. Conf. High Perform. Comput. Networking, Storage Anal. SC*, vol. 2020-Novem, 2020, doi: 10.1109/SC41405.2020.00073.
[23] 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, doi: 10.1515/itms-2017-0003.
[24] 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, doi: 10.26740/jieet.v5n2.p44-48.
[25] E. A. Nugroho, J. D. Setiawan, Munadi, and Diki, “Design of image classification system for fabric inspection process using Raspberry Pi,” *J. Mechatronics, Electr. Power, Veh. Technol.*, vol. 15, no. 1, pp. 57–67, 2024, doi: 10.55981/j.mev.2024.863.