Journal of Information Technology, computer science and Electrical Engineering (JITCSE)
Vol. 2, No. 1, February 2025, pp. 35~46
ISSN: 3046-4900

Implementation of Deep Learning CNN Algorithm for Classification of Gas Station Digitization Inventory Devices

Doli Sawaluddin[1], Zuhri Ramadhan[2]
[1,2] Faculty of Science and Technology, Universitas Pembangunan Panca Budi, Indonesia

ABSTRACT

The process of digitizing gas station devices requires careful data validation, especially on device images that are often subject to input errors. To overcome this, this research proposes the use of Convolutional Neural Network (CNN) algorithm with transfer learning technique. The pre-trained CNN model will be used to classify the device images into 13 classes. For the sake of development flexibility, the data is divided into 2 separate models.

Keyword : Convolutional Neural Network, Deep Learning, Classification, Gas Station Digitization, Device Inventory.

Corresponding Author:
Zuhri Ramadhan,
Faculty of Science and Technology
Universitas Pembangunan Panca Budi
Jl. Gatot Subroto KM. 4,5 Medan, 20122, Indonesia.
Email : ramadhanzoe@pancabudi.ac.id

Article history:
Received Jan 20, 2025
Revised Jan 27, 2025
Accepted Feb 02, 2025

INTRODUCTION

The rapid development and widespread use of information systems has changed the way we work, learn and interact. The positive impact is that we can enjoy higher quality and more efficient information services (Hermansyah et al., 2022) . On the other hand, in order to realize good governance, the government prioritizes improving and improving the quality of services provided to all levels of society (Tama et al., 2024) . One of the improvements in the quality of public services carried out by the current government is through the Gas Station Digitalization program. Digitalization of SPBU itself is an information system that in real time can monitor payment transactions and fuel distribution at spbu (Ulya et al., 2023) . As well as making it easier for people to make cashless payments.

In this Gas Station Digitalization, there are various types of devices used by the system to operate properly. Because the digitalization device operates for 24 hours, it requires data collection and periodic maintenance of these devices in order to keep the system working optimally and functioning properly. Currently, PT TELKOM AKSES has conducted routine maintenance and data collection (*inventory*) of Digitalization Station devices located in each working area, this action aims to maintain the quality and performance of the device to stay in the best condition. Inventory or inventory is the process of recording and managing all assets that have economic value, which applies to individuals, companies, and government agencies (Lestari et al., 2023) . In this Gas Station Digitalization, inventory is the work done to record the digitalization devices located at the gas station.

Information System is a series of components that work together to collect, process, and distribute information to support decision making and supervision in an organization (Syukron in Putra, 2018) . In the Gas Station Digitalization program, the information system used to collect and monitor device inventory data is called DigimonSPBU. DigimonSPBU is an information system used to monitor and collect data on spbu digitalization devices, every data inputted via telegram bot will be monitored on the DigimonSPBU web. The data that has been entered into DigimonSPBU will later be validated manually to ensure that the data inputted by the technician is correct or not. Checking is done starting from the suitability of the inputted image and serial number. This check requires accuracy and is quite time-consuming, on the other hand, inputting via telegram bot is very inefficient because technicians must input device photos, serial number photos, and retype the device serial number on the telegram bot which raises several problems such as incorrect input of device images and incorrect input of serial number characters.

Nationally, the number of gas stations that have been recorded in DigimonSPBU is 5,518 gas stations, while in the field access telkom area itself there are 166 gas stations spread from Langkat district to Serdang Bedagai district, and in each gas station there are more than 13 digitization devices with various brands and different types, this data can increase and decrease according to existing conditions and policies.

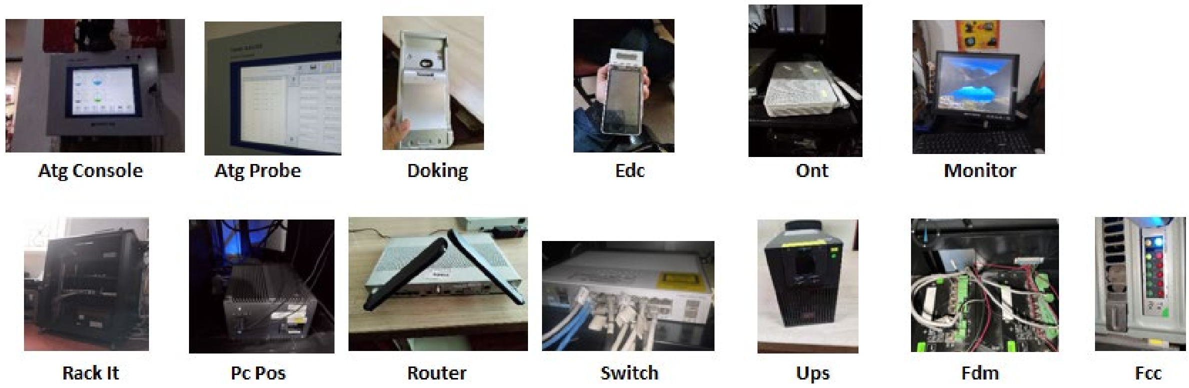


Figure 1. Gas station digitalization device

Productivity is the ability to produce maximum results using minimal resources (Andika et al., 2019) . Thus, to increase work productivity, researchers try to increase work efficiency by minimizing errors in data collection of gas station digitalization devices. To overcome device serial number input errors, currently an application has been developed to scan barcodes while taking serial number photos, so this research aims to build a device image classification model by utilizing the CNN (*Convolutional Neural Network*) algorithm to eliminate manual checking (data validation) of data entering DigimonSPBU.

1.1. Problem Formulation

1. How to build a CNN model for image classification of gas station digitization devices to eliminate manual validation?
2. How to build a model that is lightweight and easily extensible across platforms?
3. How to generate good model metrics and the model is able to predict new data well?

1.2. Problem Limitation

1. The model is limited to the classification of inventory device types shown in Figure 1. The model cannot be used for different combinations or additions of devices.
2. Model testing is limited to inference using a web browser by utilizing javascript programming language and TensorFlow.js library. Testing does not include saving to the database and the login process to the application because it is related to existing regulations.
3. Model testing is done on the web, the final result for development to server and mobile only up to the conversion of the model due to the limited resources of the device used.

1.3. Research Objectives

1. Build a CNN (*Convolutional Neural Network*) Model using python programming language and tensorflow 2.15.0 framework to classify types of gas station digitization devices.
2. Build an inference web using javascript programming language and TensorFlow.js library to test device image classification results.

1.4. Research Benefits

With this model, it is hoped that the Company can optimize employee work time, efficiency in the device data collection process, minimize errors during the data entry process and reduce human involvement in data validation so that in the end it can increase employee work productivity. This research is expected to help the development of image recognition in the field of machine learning.

RESEARCH METHODS

This research went through several stages as shown in the chart below.

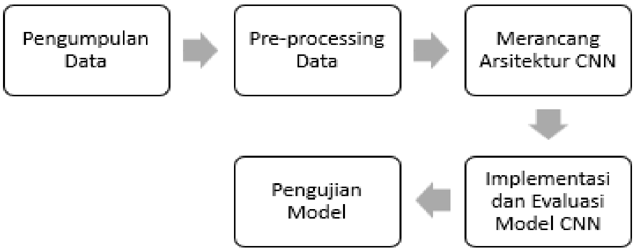


Figure 2. Research flow chart

2.1. Data Collection

In the process of collecting the image dataset, images were taken from various viewpoints and different backgrounds so that the images collected were not uniform and had a high resolution. Next, selection of unsuitable images is carried out, the aim is to obtain device images that can accurately represent each type of device. In the last stage, the collected images will be sorted and grouped according to the device label.

2.2. Data Pre-processing

At this stage, augmentation will be performed on the training data to increase the variety of data and prevent overfitting. Augmentation is applied such as rotation, flipping and changing the background color of the image with a different color.

a. *Changing the background:* At this stage, augmentation is carried out using the open cv library to reproduce image data by changing the background color of the image with different color variations.

b. *Rotation*: Performs rotation, shifting the edges of the image while maintaining the original shape.

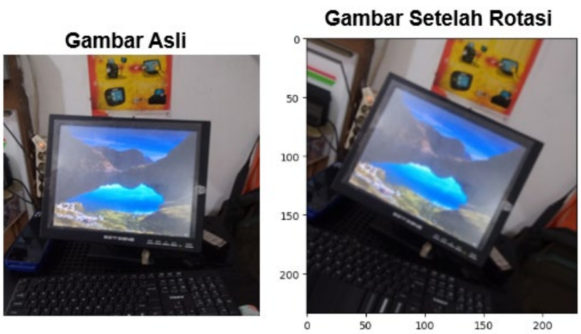
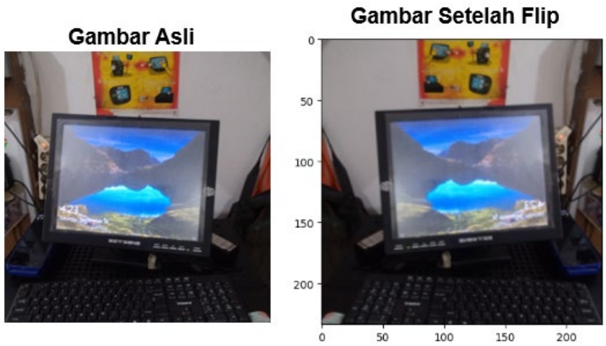


Image Rotation

c. *Flipping*: Reverses the image both vertically and horizontally.



Picture Flipping

2.3. Designing CNN Architecture

This research utilizes the pre-trained weights of the transfer learning model to improve the accuracy of the model, speed up the training process and improve the model's ability to extract relevant features. These pre-trained models have been previously trained on very large datasets, so we only need to adapt them to the model to handle more specific tasks. These pre-trained weights will be integrated in the new model, starting from the 2D Convolutional layer, 2D MaxPooling, Batch Normalization, Flatten Layer, to the last Dense layer and produce output in the form of class probabilities. In general, the CNN architecture can be seen in Figure 5.

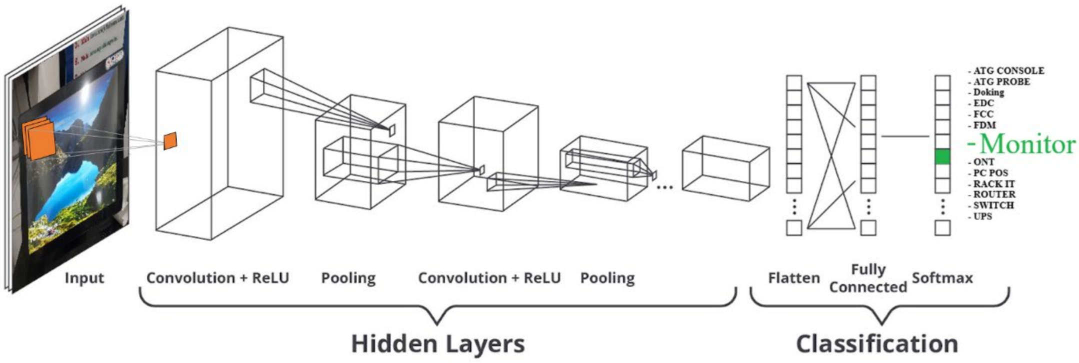
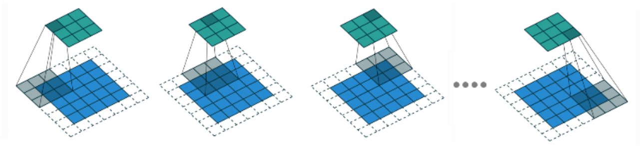


Figure 5. CNN Architecture

a. Input Layer
The input layer is a layer that receives input in the form of images, its main task is to pass data to the convolutional layer based on a predetermined size.

b. Hidden Layer
Hidden layer is a hidden layer in a neural network, where the layers in it extract important features in an image. In this research, the hidden layers in the model design are Convolutional 2D, Maxpooling, Flatten Layer, Batch Normalization, Dropout and Fully Connected (Dense).

c. Convolutional Layer
This layer will extract features in an image, such as corners and edges. This layer will move based on the filter or kernel that has been set, then through all image pixels starting from the top corner of the image to the bottom corner of the image, each filter and kernel will produce a matrix-shaped feature map.



Convolutional Layer

d. Pooling Layer
Pooling layer is responsible for reducing the dimension of the feature map, where the pooling that will be used is maxpooling layer, maxpooling will take the maximum value of the feature map based on the pooling size and then form a new feature map. The most commonly used sizes are 2x2 or 3x3.

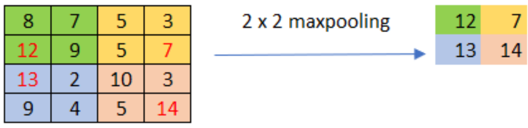


Figure 7. Pooling Layer

e. Flatten Layer
The Flatten layer will convert the multidimensional tensor generated by the convolutional, maxpoling, or previous layer results into a one-dimensional vector.

Figure 8. Flatten Layer

f. Batch Normalization and Dropout
Batch normalization is a technique to normalize inputs and make training faster and more stable while dropout is a regulation technique that can prevent overfitting by randomly deactivating neurons in the neural network.

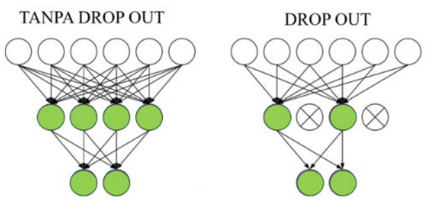


Figure 9. Dropout

g. Fully Connected Layer (Dense Layer)
This layer is connected to all neurons in the previous layer, where each input will be multiplied by a unique weight or unique parameter learned during the training process then the results will be summed before being fed to the softmax activation function, in this activation function the output of the previous layer will be converted into the probability of the device class with the highest value of 1 and 0 as the lowest value.

Furthermore, to build a model that is lightweight and easy to deploy to various platforms and facilitate development, data will be divided into 2 models, namely the model for devices in the server rack and the device model outside the server rack, with each data including 7 classes for devices in the server rack and 6 classes for devices outside the server rack. Data division can be seen in Table 1.

Table 1. Model Data Sharing

Device Server Rack (Model 1)	Device Outside the Server Rack (Model 2)
Rack It	Atg Console
Pc Pos	Atg Probe
Router	Edc
Switch	Doking
Oops	Ont
Fcc	Monitor
Fdm	

2.4. CNN Model Implementation and Evaluation

At this stage, the prepared dataset will be processed through the CNN model implemented using the Python programming language with the TensorFlow 2.15.0 framework.

The model evaluation used is the Accuracy metric, where this evaluation metric will show the overall percentage of correct predictions by dividing the correct predictions by the total predicted data.

$$Accuracy = \frac{Prediksi\ Benar}{Total\ Prediksi} \tag{1}$$

2.5. Model Testing

For model testing, a test will be carried out to predict new data, testing aims to determine the ability of the model to predict the latest data and test whether the model can run on other platforms properly.

RESULTS AND DISCUSSION

3.1 DATA SET

The real-world data collected amounted to 6500 raw data, then a selection of these images was made to obtain 5080 clean data with the following details:

Table 2: Total Raw Data

Device Name	Raw Data	Clean Data
Rack It	500	330
Pc Pos	500	219
Router	500	326
Switch	500	500
Oops	500	459
Fcc	500	260
Fdm	500	315
Atg Console	500	332
Atg Probe	500	400

Edc	500	500
Doking	500	500
Ont	500	500
Monitor	500	439
Total	**6500**	**5080**

Due to the inconsistent background of the device image, augmentation of the clean data is performed by utilizing the open cv library. The image is converted into HSV format to facilitate recognition of the basic background color, then the area outside the background color range will be isolated before the background color range is changed to another color. With this, it is expected that the model will better recognize the object and ignore the background of the image, this is very suitable for field conditions where the background of the device will always change. The process of changing the image background can be seen in Figure 10.

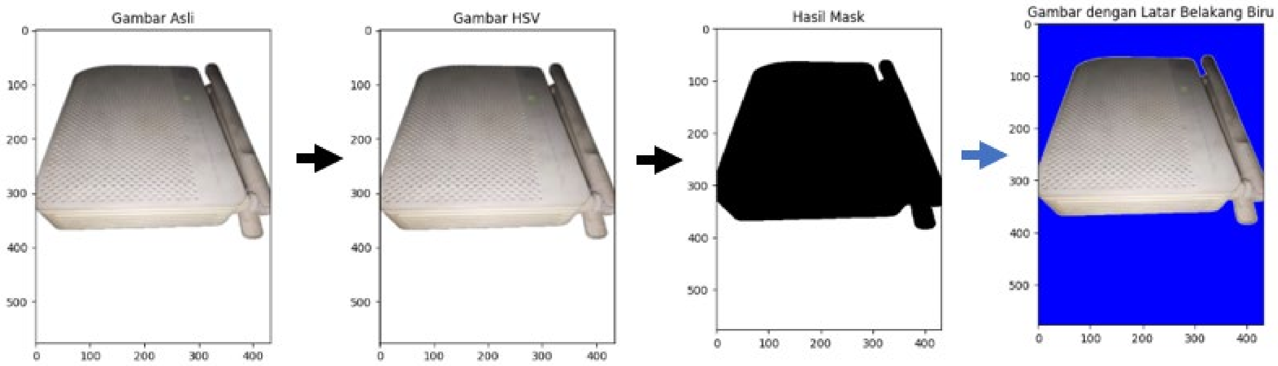


Figure 10: The process of changing the background

After performing augmentation to change the image background, rotation, and flipping on the training data, 10900 new data are obtained with the following details:

Table 3. Total Net Data

Device Name	Training	Validation	Total Data
Rack It	595	105	700
Pc Pos	595	105	700
Router	595	105	700
Switch	595	105	700
Oops	595	105	700
Fcc	595	105	700
Fdm	595	105	700
Atg Console	800	200	1000
Atg Probe	800	200	1000
Edc	800	200	1000
Doking	800	200	1000
Ont	800	200	1000
Monitor	800	200	1000
Total	**8965**	**1935**	**10900**

3.2 MODEL

To facilitate future development, the model is divided into 2 parts, the layer arrangement in each model can be seen in Table 4.

a. Transfer learning with MobileNetV2
MobileNetV2 is one of the convolutional neural network (CNN) architectures that is very suitable for devices with limited resources such as mobile devices. By freezing most of the layers in mobilenet we can save computation time during the training process and prevent overfitting. Similar to a typical CNN architecture, however, mobilnetv2 has been trained to recognize features on very large datasets. After freezing the layers in mobilnet and only opening the last few layers and connecting them to the custom model that has been designed, the following model design is obtained:

Table 4. Model 1 and Model 2

Layer (type)	Output Shape (Model 1)	Output Shape (Model 2)
mobilenetv2_1.00_224	None, 7, 7, 1280	None, 7, 7, 1280
conv2d	None, 7, 7, 96	None, 7, 7, 32
max_pooling2d	None, 4, 4, 96	None, 4, 4, 32
conv2d_1	None, 4, 4, 96	None, 4, 4, 96
max_pooling2d_1	None, 2, 2, 96	None, 2, 2, 96
conv2d_2	None, 2, 2, 128	None, 2, 2, 64
max_pooling2d_2	None, 1, 1, 128	None, 1, 1, 64
conv2d_3	None, 1, 1, 512	None, 1, 1, 384
max_pooling2d_3	None, 1, 1, 512	None, 1, 1, 384
flatten	None, 512	None, 384
batch_normalization	None, 512	None, 384
dropout	None, 512	None, 384
dense	None, 512	None, 512
dense_1	None, 512	None, 1024
dense_2	None, 7	None, 6

3.3 MODEL IMPLEMENTATION AND EVALUATION

The dataset is divided into 2 parts, namely training data and validation data. The ratio between training and validation data is 85:15 for model 1 and 80:20 for model 2. This division is made based on the most suitable data division experiment for the dataset. Furthermore, after the model design is implemented using the python language and model training is carried out, the training result plot is obtained as follows:

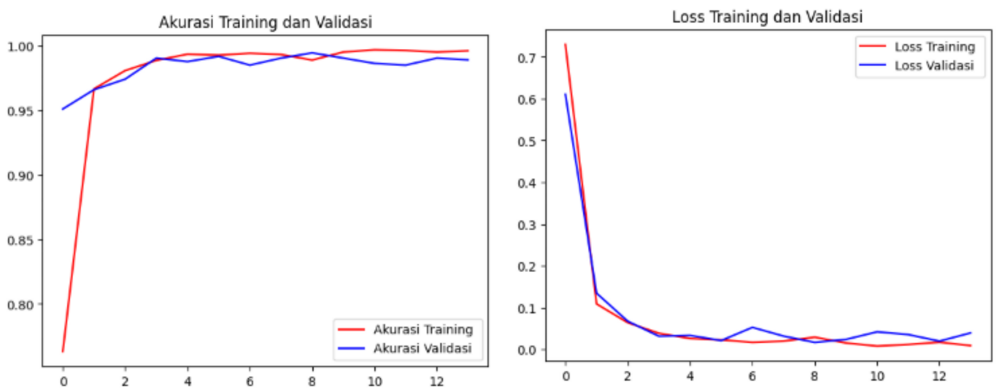


Figure 11: Model 1 Evaluation Plot

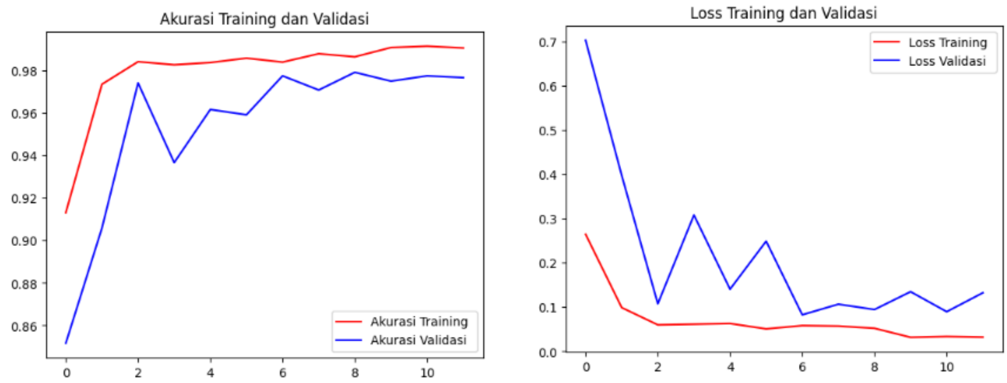


Figure 12: Model 2 Evaluation Plot

Based on the results of the evaluation plot above, we can see that the distance between accuracy and loss on training and validation data is not too far away, where the training data for model 1 has an accuracy of 0.99 and a loss of 0.0094, while for model 2 itself the accuracy is 0.99 with a loss of 0.03 so it can be said that the model is quite good at recognizing data during the training process. However, the training time is quite time consuming because the data is too large, so a large enough computation is needed to speed up the training process.

3.4 TEST RESULTS

Testing is done using validation data as many as 1935 images. The test results can be seen in the table below.

Table 5. Model Prediction Results

DEVICE NAME	CORRECT PREDICTION	FALSE PREDICTION
FCC	105	0
FDM	104	1
PC POS	105	0
RACK IT	105	0
ROUTER	105	0
SWITCH	104	1
UPS	103	2
ATG CONSOLE	195	5
ATG PROBE	192	8
DOCKING	197	3
EDC	198	2
MONITOR	191	9
ONT	200	0

Based on the table above, the accuracy metric is obtained as follows:

$$Accuracy = \frac{Prediksi\ Benar}{Total\ Prediksi}$$

$$Accuracy\ Model1\ \frac{731}{735} = 0{,}99$$

$Accuracy\ Model2\ \frac{1173}{1200} = 0{,}98$

Based on this data, the accuracy obtained is 99% for model 1 and 98% for model 2. Thus the model that has been built can predict the type of device very accurately.

To test whether the model can run on platforms other than notebooks, an integration test between the model and an application that can run on a mobile browser was conducted. The following is how inference works along with the display of device predictions on the platform:

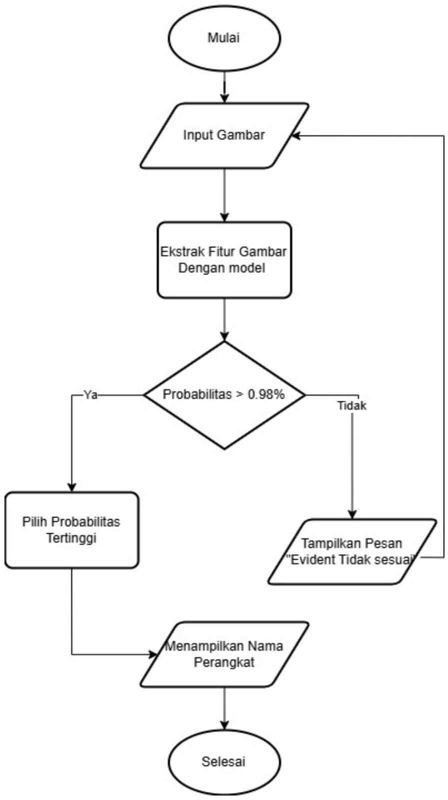
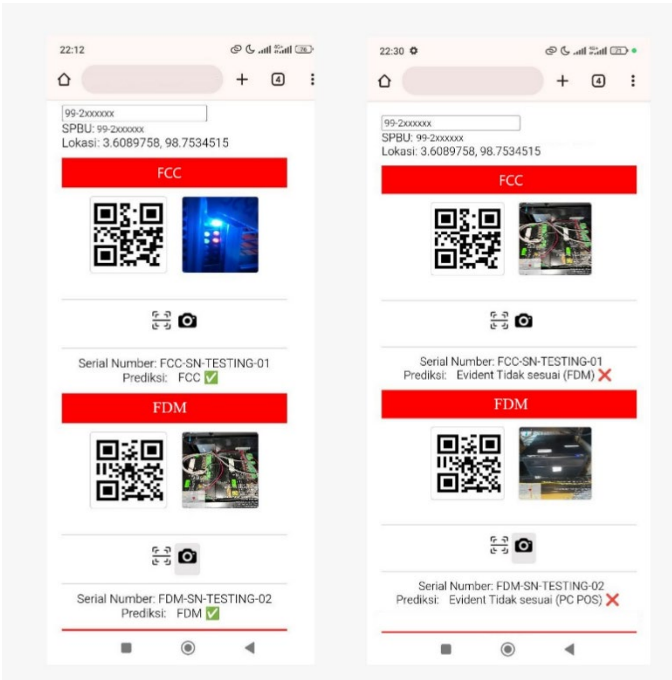


Figure 13. How the Inference Application Works



GAMBAR INPUT SESUAI PROBABILITAS > 0.98

GAMBAR INPUT TIDAK SESUAI PROBABILITAS < 0.98

Figure 14. Prediction with Application

Based on these tests, when taking pictures through a cellphone camera, the model can run smoothly and provide good predictions.

CONCLUSIONS

Based on the research that has been done, the models that have been designed can predict well where the accuracy rate is very high, namely 0.99 for model 1 and 0.98 for model 2, and both models can run well on platforms other than notebooks. By increasing the amount of data, maintaining data quality and selecting the right model layer, this research has succeeded in building a CNN model to perform classification while making the model run on different platforms, and producing a fairly good accuracy metric.

Finally, we can add this model as a feature to make predictions on the DigimonSPBU web or on other mobile applications without having to change the existing application much.

LITERATURE

Andika, R., Widjarnako, D. B., & Ahmad, R. (2019). THE EFFECT OF WORK MOTIVATION AND WORK COMPETITION ON WORK PRODUCTIVITY THROUGH JOB SATISFACTION AS AN INTERVENING VARIABLE IN EMPLOYEES OF PANCA BUDI DEVELOPMENT UNIVERSITY MEDAN. *Journal of Management Tools,11* (1), 189-205. Retrieved from https://jurnal.pancabudi.ac.id/index.php/JUMANT/article/view/513

Hermansyah, H., Wahyuni, S., & Akbar, A. (2022). Designing Web-Based Information Media Facilities for Klambir Lima Village Using the Waterfall Method. *JURIKOM (Journal of Computer Research),9* (2), 515-521. https://doi.org/10.30865/jurikom.v9i2.3803

Lestari, A., Hariyanto, E., & Hardinata, R. S. (2023). ANALYSIS OF INVENTORY INFORMATION SYSTEMS USING WEB ENGINEERING METHODS AT THE PRIMARY COOPERATIVE KARTIKA SEJAHTERA. *Journal of Information Technology and Computer Science,6* (2), 1156-1162. https://doi.org/10.31539/intecoms.v6i2.5346

Putra, E. (2018). IMPLEMENTATION OF A WEB-BASED POPULATION DATA PROCESSING INFORMATION SYSTEM AT THE OFFICE OF THE VILLAGE HEAD OF SUMUR JAUH, KELING LAKE SUB-DISTRICT, KERINCI DISTRICT, JAMBI PROVINCE. *JOURNAL OF ENGINEERING AND INFORMATICS,5* (1), 14-17. Retrieved from https://garuda.kemdikbud.go.id/documents/detail/1054632

Tama, A., Amin, M., & Ramadhan, Z. (2024). Information System Criticism and Suggestions for Optimizing Services at the Padang Tualang Subdistrict Office, Web-Based Using the CodeIgniter Framework. *Install: Computer Journal,16* (3), 331-339. https://doi.org/10.54209/jurnalinstall.v16i03.238

Ulya, N., Rachmawati, & Suandi, I. (2023). FIBER OPTIC ATTENUATION ANALYSIS ON PERTAMINA GAS STATION DIGITALIZATION SYSTEM IN LHOKSEUMAWE CITY. *TEKTRO JOURNAL,7* (1), 51-57. https://doi.org/10.30811/tektro.v7i1.3861

Adam, S., Smith, P., & Harris, M. (2020). *Chatbots in Customer Service: Enhancing User Satisfaction Through Interactivity and Responsiveness.*

Araujo, A. (2024). *Challenges and Opportunities in AI-Assisted Education: A Study on Student Perceptions of ChatGPT.* Journal of Educational Technology, 38(1), 57-70.

Bravo, D. (2024). *The Impact of ChatGPT and AI Chatbots on Technical Education: Advancing Language Processing and Machine Learning Techniques.*

Chiong, C., & Jovanovic, J. (2022). "The Role of Artificial Intelligence in Supporting Educational Engagement." *Journal of Educational Technology Systems, 50*(4), 409-426

Hidayat, I. (2021). *Artificial Intelligence in Education: Opportunities and Challenges for the Future*. Jakarta: Andi Publisher.

Hidayat, I. (2021). *Artificial Intelligence in Education: Opportunities and Challenges for the Future*. Jakarta: Andi Publisher.