IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 12, No. 3, September 2023, pp. 1386~1395
ISSN: 2252-8938, DOI: 10.11591/ijai.v12.i3.pp1386-1395 ❒ 1386

Energy-efficient deep learning model for fruit freshness detection

Febrian Valentino[1], Tjeng Wawan Cenggoro[2,3], Gregorius Natanael Elwirehardja[3], Bens Pardamean[1,3]
[1]Department of Computer Science, BINUS Graduate Program-Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia
[2]Department of Computer Science, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia
[3]Bioinformatics and Data Science Research Center, Bina Nusantara University, Jakarta, Indonesia

Article Info

Article history:

Received Apr 4, 2022
Revised Aug 25, 2022
Accepted Oct 11, 2022

Keywords:

Convolutional neural network
Deep learning
Detection
Fruit
Image classification

ABSTRACT

Fruits are usually used as complementary foods because they contain good nutrients such as protein and vitamins. In addition to having good content, it turns out that there are potentially harmful microorganisms contained in fruits caused by decay. Currently, many artificial intelligence (AI) techniques have been proposed in research related to fruit freshness. Deep learning is one of its most prominent types in similar studies. As deep learning typically requires a lot of computation power, it usually consumes a lot of electricity. This is an important concern, especially for agribusiness companies that require AI implementations. Based on these problems, we propose to build a convolutional neural network (CNN) model consisting of six layers to detect fruit freshness and save energy. The CNN model we built uses electrical power ranging from 55 to 73 Watts during the training process and 20 to 27 Watts during the testing process. For accuracy, the result is 98.64%. However, compared to previous studies with the MobileNetV2 model, our model only excels in several aspects, such as recall in fresh banana and fresh oranges, recall and F1-score in Rotten Banana.

Corresponding Author:

Febrian Valentino
Department of Computer Science, BINUS Graduate Program - Master of Computer Science
Bina Nusantara University
Jakarta, Indonesia, 11480
Email: febrian.valentino@binus.ac.id

1. INTRODUCTION

Technology and globalization have made significant changes to life on this earth [1]. This can be proven by the rampant era of digitization [2] and the fewer resources available at this time [3]. Humans living in the current digitalization era are accustomed to an instant lifestyle, including their dietary habits, which will impact health [4]. As time goes by, more and more people are aware that one factor affecting their health is the nutrition they get from their food [5]. The necessary nutrients can be obtained from consuming foods such as meat, fruits, and vegetables.

Fruits are usually used as complementary food because it has excellent nutritional contents such as protein, vitamins, and so on [6]. In addition to having good contents, it turns out that potentially harmful microorganisms such as germs and bacteria can be contained in fruits. The microorganisms appear because the fruit has undergone a decay process [7]. Currently, there are still many fruit supplier companies, both imported and local, that send fruits that are not fit for consumption due to the inaccuracy of the classification process carried out by company employees [8] and supermarkets that still sell fruits that are not fit for consumption.

Based on the problem described earlier, detecting food spoilage, especially fruits, is important, starting from production in plantations to consumption. Consequently, many techniques based on computer vision and artificial intelligence (AI) have been proposed in the last few decades, as discussed by Moeslund and Granum [9]. This technique is proposed because machine learning algorithms are successful in making computers that can learn by themselves [10], [11], and an independent learning method called deep learning emerged [12], [13] and became the current trend in AI research.

Deep learning is an algorithm that seeks to learn at different levels of abstraction [14]. The deep learning algorithm is inspired by the structure of the human brain, where each neural layer's structured nervous system is interconnected and conveys information. This reflects the function of the human brain, where every time we receive new knowledge, the brain tries to analyze it with previously known knowledge. Deep learning has excellent capabilities for computers in self-learning, from natural language processing to image processing [15]. This is because deep learning can process input from raw data, making it superior to machine learning [12]. As one of the latest trends in computer science research, especially machine learning and AI [16], deep learning is making significant breakthroughs around the world. One of them is digital image processing [17]. The type of deep learning most often applied to perform digital image processing is convolutional neural network (CNN) [18]. Not only CNN but there are also other types of neural networks, such as artificial neural networks (ANN) [19] and recurrent neural networks (RNN) [20]. CNN is generally used to perform image processing, face recognition, image classification, and object detection. This is because CNN is a type of deep learning that can receive input from images by determining what aspects or objects in an image can be studied and then being able to distinguish one image from another.

Many fields of work are helped by deep learning. For a crucial area of work, such as the health sector, deep learning can classify skin cancer [21] and breast cancer [22], which are diseases that cause many deaths in the world [23]. In addition to the health sector, there is also the agricultural and plantations sector. In the agricultural and plantations sector, deep learning is applied to classify oil palm fruit based on maturity [24], [25] and then developed as an automatic oil palm fruit picking machine that can harvest oil palm fruit according to the level of maturity [26].

Apart from the agricultural and plantation areas already mentioned, there are other areas such as detection of fruit freshness. The problem that gave rise to the idea of applying deep learning to detect fruit freshness is that conventional spoilage detection techniques are still slow and time-consuming [27]. Based on these problems, a method for detecting rotten fruit was developed based on digital image processing with machine learning [28]–[30], which has proven to give high potential in the agricultural and plantations industry [31].

From the high potential generated by the application of machine learning, Karakaya *et al*. conducted a comparative study of machine learning and deep learning feature extraction [27]. The results obtained are deep learning provides better accuracy than machine learning [27]. Then further, Chakraborty *et al*. implemented deep learning called MobileNetV2 to identify rotten fruit [32]. Based on previous research related to fruit freshness that has been mentioned, this research still aims to find the best accuracy by comparing models.

Behind the success of deep learning in making computers self-learning has negative impacts, such as large electricity consumption. This is a problem for companies that require AI implementations, especially agribusiness companies [33]. In 2019, researchers at the University of Massachusetts Amherst estimated that training a deep learning model could use 12,041.51 Watts of electrical power and generate up to 626,155 pounds of carbon dioxide (CO2) emissions [34]. It is also an important concern to create deep learning models with less energy consumption.

Regarding fruit freshness research, no one has aimed to build a deep learning model or compare deep learning models that are energy efficient but still provide good accuracy. Research conducted by previous researchers still aims to find the best accuracy by comparing models. This is certainly a gap in research related to fruit freshness. This proposed study is a continuation of our previous study [35] and is focused on building an accurate and energy-efficient CNN model specifically for detecting fruit freshness. This is because the application of AI technology must aim to optimize work efficiency in terms of time and cost.

2. RESEARCH METHOD

The method proposed in this research consists of data preprocessing and data augmentation, CNN model design, hyperparameter tuning, model building, testing, and evaluation. The diagram of the research methodology is shown in Figure 1. The dataset used in this study is a publicly accessible dataset called fruits fresh and rotten for classification. The dataset contains images of fresh and rotten fruit from a specific class consisting of two folders, the train folder, and the test folder. From the two folders, there are 10,901 images divided into six classes, namely Freshapples, Freshbanana, Freshoranges, Rottenapple, Rottenbanana, and Rottenoranges which can be seen in Figure 2. We split the dataset into training, validation, and testing in our experiment. The training data uses 80% of the data from the train folder, and the validation data uses the rest. For test data, use data that comes from the test folder. After the data separation is complete, the next step is to enter the data pre-processing and data augmentation stage.

Data preprocessing is done by resizing the data to a size of 224×224 and data augmentation, including geometric transformations (Zoom, width shifting, height shifting, horizontal flip, and shear intensity). Data augmentation aims to make the CNN model able to learn and recognize geometrically or photometrically transformed data. In most cases, the use of data augmentation has been successful in improving the performance of deep learning models. This increase occurred because the model could recognize more objects of various types and patterns. Data that has been pre-processed and augmented is used to train the designed CNN model and the comparison model, namely residual network 50 (ResNet-50) [36].

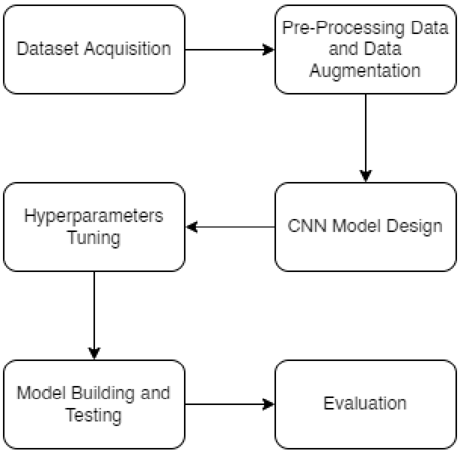

Figure 1. Research methodology

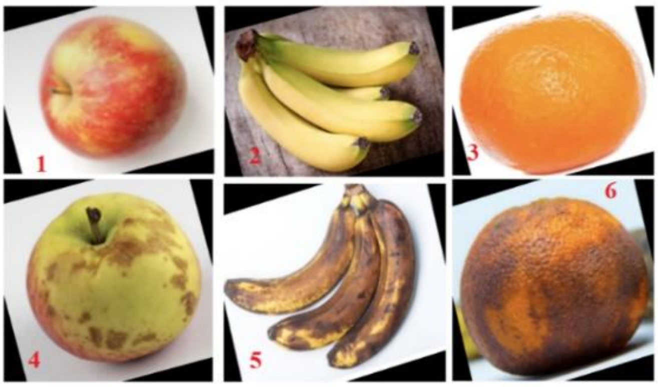

Figure 2. Examples of fruit classes in the datasets

2.1. Experimental setup and cnn model design and hyperparameter tuning

This study was conducted on a personal computer with the specifications of Processor Intel® Core™ i5-9400F CPU @ 2.90 GHz × 6, 32 GB DDR4 RAM, GPU NVIDIA GeForce RTX 2080Ti 11 GB. The device uses the Ubuntu 18.04 64 bits operating system with the python 3.6 programming language and the open-source TensorFlow library with Keras deep learning framework. The deep learning method that we apply is a six-layer CNN. Our model has MaxPooling2D with pool size (2.2), Conv2D with values 32, 64, 128, 256, 512, 1024. The hyperparameters we will compare to find the best model are learning rate 0.001 and 0.0001, batch size 20 and 32, 50 epochs, Adam as an optimizer, and rectified linear units (ReLU) as activation function. The CNN model we designed is illustrated in Figure 3.

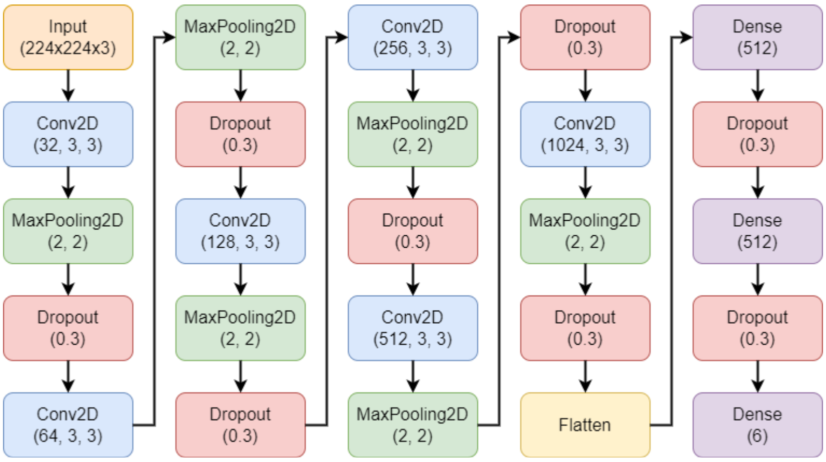

Figure 3. CNN model design

2.2. Model building and testing

After the CNN model is designed, the CNN model is built, and a training process is carried out with the prepared data and by comparing the previously mentioned hyperparameter sets to find the best model configuration. The dataset that is trained on our model produces 7,078,726 parameters that can be trained. While the ResNet-50 model gets 24,902,534 parameters, but only 10,246,150 parameters can be trained. A brief explanation of these parameters is shown in Figure 4.

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_6 (MaxPooling2	(None, 111, 111, 32)	0
dropout_8 (Dropout)	(None, 111, 111, 32)	0
conv2d_7 (Conv2D)	(None, 109, 109, 64)	18496
max_pooling2d_7 (MaxPooling2	(None, 54, 54, 64)	0
dropout_9 (Dropout)	(None, 54, 54, 64)	0
conv2d_8 (Conv2D)	(None, 52, 52, 128)	73856
max_pooling2d_8 (MaxPooling2	(None, 26, 26, 128)	0
dropout_10 (Dropout)	(None, 26, 26, 128)	0
conv2d_9 (Conv2D)	(None, 24, 24, 256)	295168
max_pooling2d_9 (MaxPooling2	(None, 12, 12, 256)	0
dropout_11 (Dropout)	(None, 12, 12, 256)	0
conv2d_10 (Conv2D)	(None, 10, 10, 512)	1180160
max_pooling2d_10 (MaxPooling	(None, 5, 5, 512)	0
dropout_12 (Dropout)	(None, 5, 5, 512)	0
conv2d_11 (Conv2D)	(None, 3, 3, 1024)	4719616
max_pooling2d_11 (MaxPooling	(None, 1, 1, 1024)	0
dropout_13 (Dropout)	(None, 1, 1, 1024)	0
flatten_1 (Flatten)	(None, 1024)	0
dense_3 (Dense)	(None, 512)	524800
dropout_14 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 512)	262656
dropout_15 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 6)	3078

Total params: 7,078,726
Trainable params: 7,078,726
Non-trainable params: 0

Our Model

Layer (type)	Output Shape	Param #
model (Model)	(None, 2048)	23587712
dense (Dense)	(None, 512)	1049088
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 512)	262656
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 6)	3078

Total params: 24,902,534
Trainable params: 10,246,150
Non-trainable params: 14,656,384

ResNet-50

Figure 4. Summary of the total parameter

If the data training process has been completed, it will produce an output file with the format (.pb). The output file is used in the testing process with the help of the Docker platform. The output file will be converted into docker images which can then be accessed using the application programming interface (API). The API is included in a testing script written in python.

3. RESULTS

3.1. Training evaluation

In this experimental scenario, we compare the hyperparameters with the hyperparameter set described in subsection 2.1. Then the hyperparameters are used when training the data to find the hyperparameters that give the best results for the CNN model that was built. The results of the comparison of hyperparameters are shown in Figure 5 and detailed in Table 1.

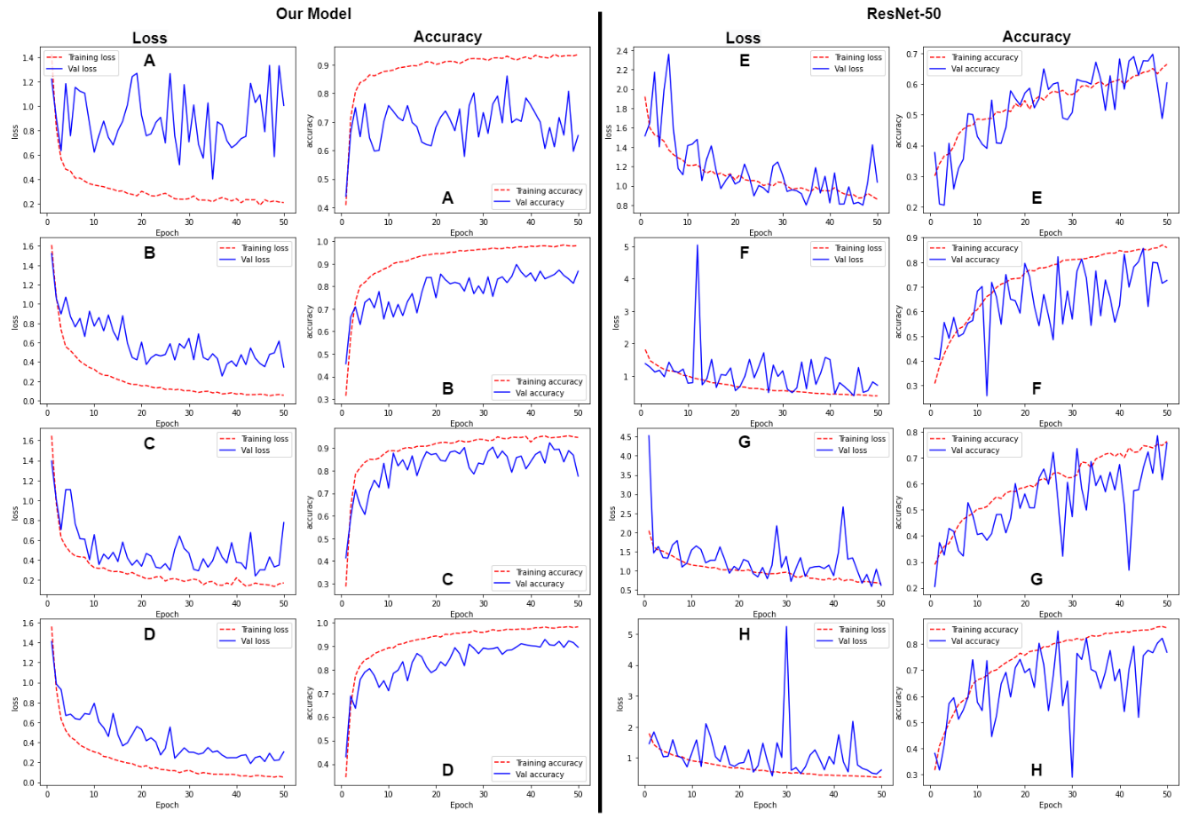

Figure 5. Graphical accuracy and loss of both models

Table 1. Accuracy and loss scores on our model and ResNet-50

Configuration	Our Model				ResNet-50			
	Train Acc	Valid Acc	Train Loss	Valid Loss	Train Acc	Valid Acc	Train Loss	Valid Loss
Bs = 20, Lr = 0.001	0.9353 (Figure 5(a))	0.6521 (Figure 5(a))	0.2077 (Figure 5(a))	1.0042 (Figure 5(a))	0.6647 (Figure 5(e))	0.6041 (Figure 5(e))	0.8628 (Figure 5(e))	1.0383 (Figure 5(e))
Bs = 20, Lr = 0.0001	0.9806 (Figure 5(b))	0.8671 (Figure 5(b))	0.0539 (Figure 5(b))	0.3453 (Figure 5(b))	0.8595 (Figure 5-F)	0.7265 (Figure 5-F)	0.3815 (Figure 5-F)	0.7071 (Figure 5-F)
Bs = 32, Lr = 0.001	0.9474 (Figure 5(c))	0.7763 (Figure 5(c))	0.1720 (Figure 5(c))	0.7765 (Figure 5(c))	0.7628 (Figure 5(g))	0.7566 (Figure 5(g))	0.6587 (Figure 5(g))	0.6166 (Figure 5(g))
Bs = 32, Lr = 0.0001	0.9819 (Figure 5(d))	0.8950 (Figure 5(d))	0.0509 (Figure 5(d))	0.3053 (Figure 5(d))	0.8624 (Figure 5(h))	0.7685 (Figure 5(h))	0.3765 (Figure 5(h))	0.6057 (Figure 5(h))

From the results of the training process, our model with hyperparameter set batch size=32, activation function = ReLU, optimizer = Adam and learning rate=0.0001 shows the result is an accuracy of 0.8950 (89.50%). Meanwhile, ResNet-50 as a comparison model, gets the result with an accuracy of 0.7685 (76.85%) using the same set of hyperparameters. The performance of each model is illustrated graphically in Figure 5.

When the training process of our model is executed, the GPU memory used is 10,011 MB and consumes electricity ranging from 55 to 73 Watts. The ResNet-50 model uses less memory at 9863 MB. However, the electricity consumed is greater, which is 59 to 171 Watts. GPU performance when training both models is graphically illustrated in Figure 6.

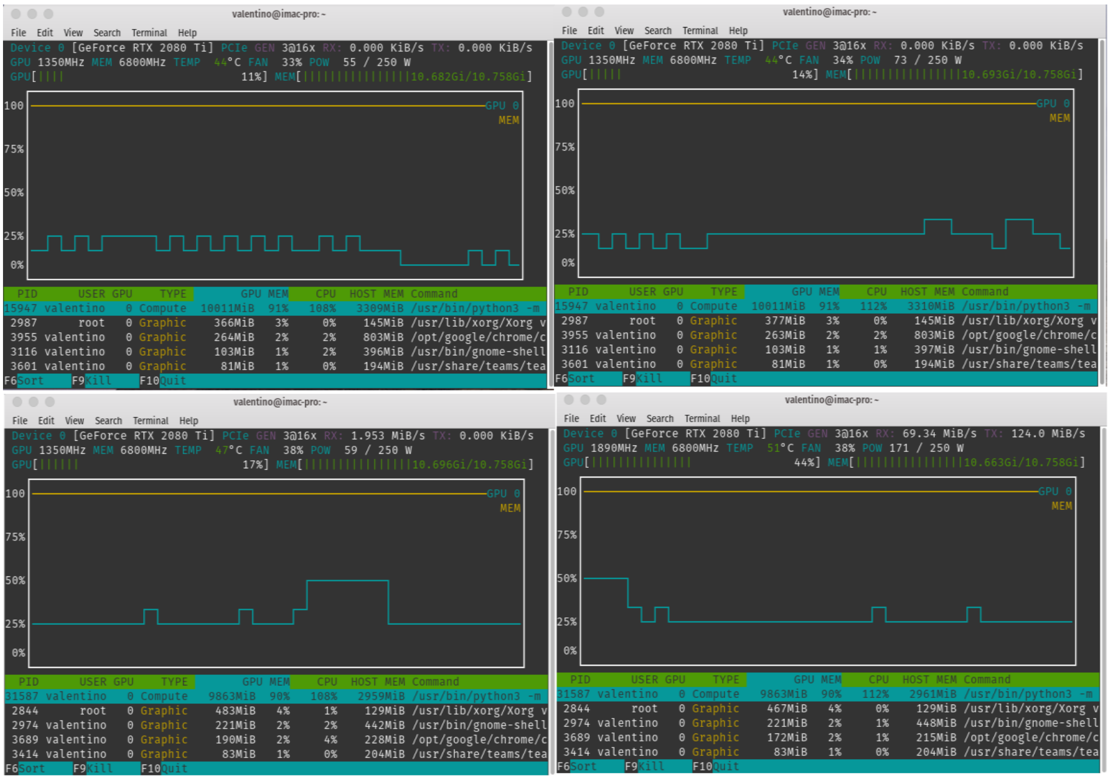

Figure 6. GPU performance while training data both models

ResNet-50 is a neural network model with a large architecture. This can be seen from the number of parameters described in subsection 2.2. So that when training data with the ResNet-50 model, the GPU load becomes larger from 17% to 44%, as illustrated in Figure 6. The GPU also requires a large power consumption when processing with a large load. This is why the ResNet-50 model consumes more power than our model.

3.2. Testing evaluation

The test results are obtained after running a script to predict images written in Python3. Based on the test results, our trained model can predict all images in the test folder in 411,226 seconds or about 6 minutes 51 seconds. Meanwhile, the ResNet-50 trained model takes 456,351 seconds or about 7 minutes 36 seconds. The length of time for testing the two models can be seen in Figure 7.

```
test/rottenoranges/rotated_by_15_Screen Shot 2018-06-12 at 11.21.22 PM.png  > > > SUCCESS
test/rottenoranges/rotated_by_75_Screen Shot 2018-06-12 at 11.18.28 PM.png  > > > SUCCESS
test/rottenoranges/vertical_flip_Screen Shot 2018-06-12 at 11.32.33 PM.png  > > > SUCCESS
test/rottenoranges/saltandpepper_Screen Shot 2018-06-12 at 11.32.13 PM.png  > > > SUCCESS
test/rottenoranges/rotated_by_30_Screen Shot 2018-06-12 at 11.40.23 PM.png  > > > SUCCESS
test/rottenoranges/rotated_by_60_Screen Shot 2018-06-12 at 11.23.09 PM.png  > > > SUCCESS
test/rottenoranges/Screen Shot 2018-06-12 at 11.44.48 PM.png  > > > SUCCESS
--- 456.3511438369751 seconds ---
```

ResNet-50

```
test/rottenoranges/rotated_by_15_Screen Shot 2018-06-12 at 11.21.22 PM.png  > > > SUCCESS
test/rottenoranges/rotated_by_75_Screen Shot 2018-06-12 at 11.18.28 PM.png  > > > SUCCESS
test/rottenoranges/vertical_flip_Screen Shot 2018-06-12 at 11.32.33 PM.png  > > > SUCCESS
test/rottenoranges/saltandpepper_Screen Shot 2018-06-12 at 11.32.13 PM.png  > > > SUCCESS
test/rottenoranges/rotated_by_30_Screen Shot 2018-06-12 at 11.40.23 PM.png  > > > SUCCESS
test/rottenoranges/rotated_by_60_Screen Shot 2018-06-12 at 11.23.09 PM.png  > > > SUCCESS
test/rottenoranges/Screen Shot 2018-06-12 at 11.44.48 PM.png  > > > SUCCESS
--- 411.22669982910156 seconds ---
```

Our Model

Figure 7. Comparison of the duration testing process

After the script to predict the image is complete, it will generate a prediction report in comma separated values (CSV) format. The report results will be manually mapped into the confusion matrix. The report can be seen in Figure 8 to determine the accuracy, Micro-Precision, Micro-Recall, and Micro-F1 values.

Our Model

	Fresh apples	Fresh banana	Fresh oranges	Rotten apples	Rotten banana	Rotten oranges
Freshapples	384	0	0	9	0	2
Freshbanana	3	366	2	3	1	6
Freshoranges	0	0	337	2	0	49
Rottenapples	6	0	0	579	0	16
Rottenbanana	1	0	0	1	526	2
Rottenoranges	0	0	4	3	0	396

ResNet-50

	Fresh apples	Fresh banana	Fresh oranges	Rotten apples	Rotten banana	Rotten oranges
Freshapples	268	0	37	89	0	1
Freshbanana	1	328	11	37	4	0
Freshoranges	2	3	358	19	2	4
Rottenapples	39	0	58	471	0	33
Rottenbanana	1	2	1	23	499	4
Rottenoranges	2	0	71	71	1	257

Figure 8. Confusion matrix of both models

From the mapped confusion matrix results, our model with the best set of hyperparameters described in Subsection 3.1 gets better accuracy per class when compared to the ResNet-50 model with the same parameters and Karakaya *et al.* ResNet-50 [27]. The results of the accuracy comparison for each class are presented in Table 2. From Table 2, the model we built got an average accuracy of 98.64%, followed by Karakaya *et al.* ResNet-50 of 97.61%, and the last one is ResNet-50 with the same hyperparameters as our model of 93.62%. In addition to comparing with research conducted by Karakaya *et al.*, we also compare with research conducted by Chakraborty *et al*. The evaluation scores for each class are presented in Table 3.

Table 2. Each class accuracy score

Class	Our Model Accuracy	ResNet-50 Accuracy	ResNet-50 [27] Accuracy
Freshapples	99.22%	93.62%	98.67%
Freshbanana	99.44%	97.85%	99.33%
Freshoranges	97.89%	92.29%	96.50%
Rottenapples	98.52%	86.32%	96.67%
Rottenbanana	99.81%	98.59%	99.67%
Rottenoranges	96.96%	93.07%	94.83%
Average	98.64%	93.62%	97.61%

Table 3. Each class evaluation score

Class	Our Model			ResNet-50			MobileNetV2 [32]		
	Preci-sion	Recall	F1-Score	Preci-sion	Recall	F1-Score	Preci-sion	Recall	F1-Score
Freshapples	0.97	0.97	0.97	0.68	0.86	0.76	0.98	0.99	0.97
Freshbanana	0.96	1.00	0.98	0.86	0.98	0.92	0.99	0.99	0.99
Freshoranges	0.87	0.98	0.92	0.92	0.67	0.77	0.99	0.98	0.97
Rottenapples	0.96	0.97	0.97	0.78	0.66	0.72	0.98	0.99	0.98
Rottenbanana	0.99	1.00	1.00	0.94	0.99	0.96	0.99	0.99	0.99
Rottenoranges	0.98	0.84	0.91	0.64	0.86	0.73	0.99	0.98	0.98

From Table 3, our model is only superior in several aspects when compared to the research conducted by Chakraborty *et al.* with the MobileNetV2 model. This is because the MobileNetV2 model has a linear bottleneck feature and shortcut connections between bottlenecks. At the bottleneck, there are inputs and outputs between the models, while the inner layer or layer encapsulates the model's ability to convert inputs from lower-level concepts (i.e., pixels) to higher-level descriptors (i.e., image categories). In the end, as with the residual connections on CNNs in general, shortcuts between bottlenecks allow for faster training and better accuracy [37].

When the data test process with our model is performed, the GPU memory used is 356 to 367 MB and consumes electricity from 20 to 27 Watts. In contrast to the training process, the ResNet-50 model uses more memory than ours, which is 390 to 404 MB. The electricity consumed is even greater, from 22 to 28 Watts. GPU performance when testing both models is graphically illustrated in Figure 9.

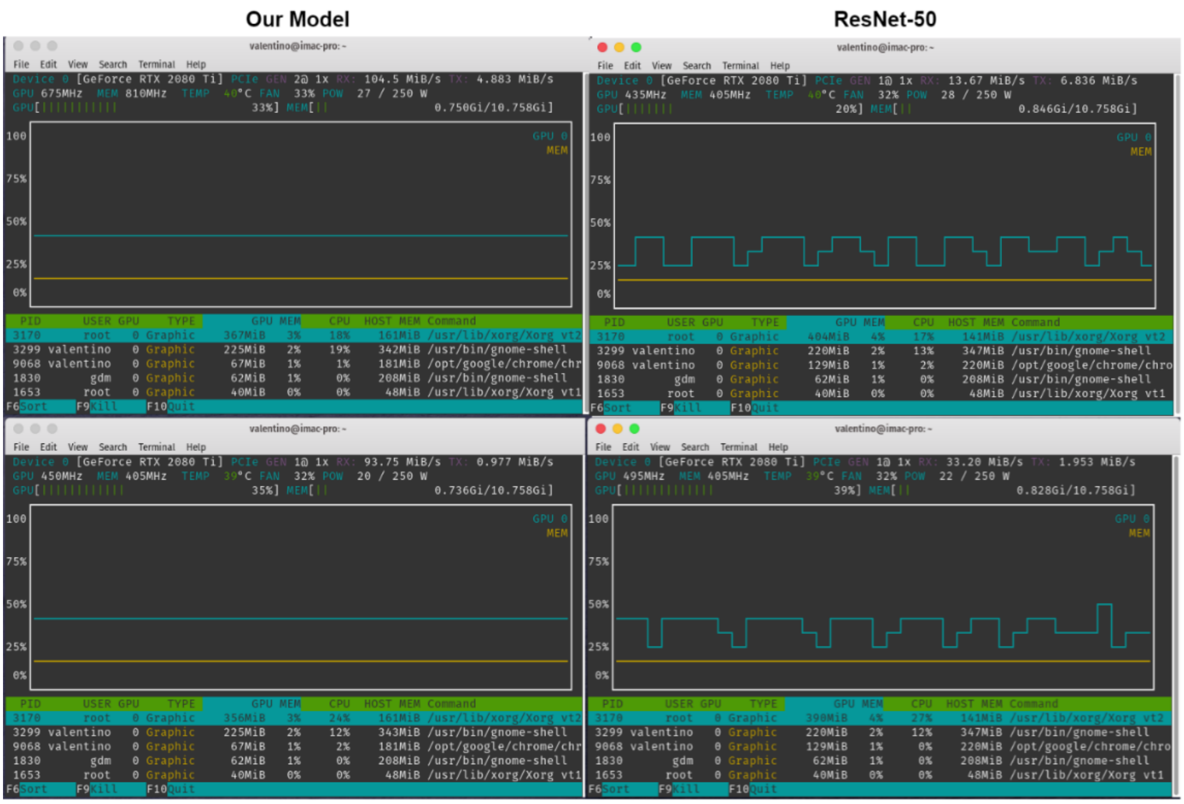

Figure 9. GPU performance while testing of both models

4. CONCLUSION

We built an energy-efficient and accurate CNN model from the experiments in this study. Based on the results of accuracy testing and testing time, our model gives better performance results than ResNet-50 by using the same hyperparameters, with 98.64% accuracy, and takes a shorter time to predict all the given images, which is 411.226 seconds. In addition to comparing with other models using the same hyperparameters, we compared the accuracy results obtained with previous studies. The average accuracy of our model is 1.03% higher than the ResNet-50 used used in the previous study. However, our model is only superior in several aspects, such as Recall in the Freshbanana and Freshoranges class and Recall and F1-Score in the Rottenbanana class when compared to other similar studies. with the MobileNetV2 model. This is because the MobileNetV2 model features linear bottlenecks and shortcut connections between bottlenecks. This feature allows for faster training and better accuracy. Not only in terms of accuracy and test time, but we also evaluated GPU memory usage and GPU power usage during the training and testing process. This is necessary for our success in building an energy-efficient CNN model. Even though during the training process, it uses more memory by a difference of 142 MB, it uses less power than the ResNet-50 model. The power used ranges from 55 to 73 Watts. During the testing process, the ResNet-50 model uses more memory with a difference of about 33 to 48 MB and more extensive power of about 2 to 8 Watts.

REFERENCES

[1] J. Tica, T. Globan, and V. Arčabić, “Managing the impact of globalization and technology on inequality,” *Economic Research-Ekonomska Istrazivanja* , vol. 35, no. 1, pp. 1035–1060, 2022, doi: 10.1080/1331677X.2021.1952466.

[2] Y. Kayikci, “Sustainability impact of digitization in logistics,” *Procedia Manufacturing*, vol. 21, pp. 782–789, 2018, doi: 10.1016/j.promfg.2018.02.184.
[3] D. S. Grebenkov, “Depletion of resources by a population of diffusing species,” *Physical Review E*, vol. 105, no. 5, 2022, doi: 10.1103/PhysRevE.105.054402.
[4] Y. B. Zhang *et al.*, “Combined lifestyle factors, all-cause mortality and cardiovascular disease: A systematic review and meta-analysis of prospective cohort studies,” *Journal of Epidemiology and Community Health*, vol. 75, no. 1, pp. 92–99, 2021, doi: 10.1136/jech-2020-214050.
[5] R. A. Wahab, “Food label use and awareness of nutritional information among consumers in Bahrain: An exploratory study,” *KnE Life Sciences*, vol. 4, no. 6, p. 26, 2018, doi: 10.18502/kls.v4i6.3088.
[6] M. Mathiazhagan, B. Chidambara, L. R. Hunashikatti, and K. V. Ravishankar, “Genomic approaches for improvement of tropical fruits: Fruit quality, shelf life and nutrient content,” *Genes*, vol. 12, no. 12, 2021, doi: 10.3390/genes12121881.
[7] P. Pétriacq, A. López, and E. Luna, “Fruit decay to diseases: Can induced resistance and priming help?,” *Plants*, vol. 7, no. 4, 2018, doi: 10.3390/plants7040077.
[8] Y. P. Tsang, K. L. Choy, C. H. Wu, G. T. S. Ho, H. Y. Lam, and V. Tang, “An intelligent model for assuring food quality in managing a multi-temperature food distribution centre,” *Food Control*, vol. 90, pp. 81–97, 2018, doi: 10.1016/j.foodcont.2018.02.030.
[9] T. B. Moeslund and E. Granum, “A survey of computer vision-based human motion capture,” *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001, doi: 10.1006/cviu.2000.0897.
[10] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, “Machine learning,” *Machine Learning*, vol. 45, no. 13, pp. 40–48, 2017, [Online]. Available: https://books.google.ca/books?id=EoYBngEACAAJ&dq=mitchell+machine+learning+1997&hl=en&sa=X&ved=0ahUKEwiomdqfj8TkAhWGslkKHRCbAtoQ6AEIKjAA.
[11] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, “Machine learning in agriculture: A review,” *Sensors (Switzerland)*, vol. 18, no. 8, 2018, doi: 10.3390/s18082674.
[12] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
[13] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018, doi: 10.1016/j.compag.2018.02.016.
[14] L. Deng and D. Yu, “Deep learning: Methods and applications,” *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2013, doi: 10.1561/2000000039.
[15] C. Lammie, A. Olsen, T. Carrick, and M. Rahimi Azghadi, “Low-power and high-speed deep FPGA inference engines for weed classification at the edge,” *IEEE Access*, vol. 7, pp. 51171–51184, 2019, doi: 10.1109/ACCESS.2019.2911709.
[16] M. R. Minar and J. Naher, “Recent advances in deep learning: An overview,” 2018, doi: 10.13140/RG.2.2.24831.10403.
[17] N. O’Mahony *et al.*, “Deep learning VS traditional computer vision,” *Advances in Intelligent Systems and Computing*, vol. 943, pp. 128–144, 2020, doi: 10.1007/978-3-030-17795-9_10.
[18] Y. L. and Y. Bengio, “Convolutional networks for images, speech, and time-series,” *The handbook of brain theory and neural networks*, pp. 255–258, 1998.
[19] X. Yao, “Evolving artificial neural networks,” *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999, doi: 10.1109/5.784219.
[20] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur, “Extensions of recurrent neural network language model,” *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 5528–5531, 2011, doi: 10.1109/ICASSP.2011.5947611.
[21] K. M. Hosny, M. A. Kassem, and M. M. Foaud, “Skin cancer classification using deep learning and transfer learning,” *2018 9th Cairo International Biomedical Engineering Conference, CIBEC 2018 - Proceedings*, pp. 90–93, 2019, doi: 10.1109/CIBEC.2018.8641762.
[22] B. Pardamean, T. W. Cenggoro, R. Rahutomo, A. Budiarto, and E. K. Karuppiah, “Transfer learning from chest X-ray pre-trained convolutional neural network for learning mammogram data,” *Procedia Computer Science*, vol. 135, pp. 400–407, 2018, doi: 10.1016/j.procs.2018.08.190.
[23] R. L. Siegel, K. D. Miller, H. E. Fuchs, and A. Jemal, “Cancer statistics, 2022,” *CA: A Cancer Journal for Clinicians*, vol. 72, no. 1, pp. 7–33, 2022, doi: 10.3322/caac.21708.
[24] H. Herman, A. Susanto, T. W. Cenggoro, S. Suharjito, and B. Pardamean, “Oil palm fruit image ripeness classification with computer vision using deep learning and visual attention,” *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 12, no. 2, pp. 21–27, 2020, [Online]. Available: https://jtec.utem.edu.my/jtec/article/view/5543.
[25] H. Herman, T. W. Cenggoro, A. Susanto, and B. Pardamean, “Deep learning for oil palm fruit ripeness classification with densenet,” *Proceedings of 2021 International Conference on Information Management and Technology, ICIMTech 2021*, pp. 116–119, 2021, doi: 10.1109/ICIMTech53080.2021.9534988.
[26] Harsawardana *et al.*, “AI-based ripeness grading for oil palm fresh fruit bunch in Smart Crane Grabber,” *IOP Conference Series: Earth and Environmental Science*, vol. 426, no. 1, p. 12147, 2020, doi: 10.1088/1755-1315/426/1/012147.
[27] D. Karakaya, O. Ulucan, and M. Turkan, “A Comparative Analysis on Fruit Freshness Classification,” *Proceedings - 2019 Innovations in Intelligent Systems and Applications Conference, ASYU 2019*, 2019, doi: 10.1109/ASYU48272.2019.8946385.
[28] S. Singh and N. P. Singh, “Machine learning-based classification of good and rotten apple,” *Lecture Notes in Electrical Engineering*, vol. 524, pp. 377–386, 2019, doi: 10.1007/978-981-13-2685-1_36.
[29] K. Roy, A. Ghosh, D. Saha, J. Chatterjee, S. Sarkar, and S. S. Chaudhuri, “Masking based segmentation of rotten fruits,” *2019 International Conference on Opto-Electronics and Applied Optics, Optronix 2019*, 2019, doi: 10.1109/OPTRONIX.2019.8862396.
[30] A. Nosseir and S. E. Ashraf Ahmed, “Automatic classification for fruits’ types and identification of rotten ones using k-NN and SVM,” *International journal of online and biomedical engineering*, vol. 15, no. 3, pp. 47–61, 2019, doi: 10.3991/ijoe.v15i03.9832.
[31] M. Teena, A. Manickavasagan, A. Mothershaw, S. El Hadi, and D. S. Jayas, “Potential of Machine Vision Techniques for Detecting Fecal and Microbial Contamination of Food Products: A Review,” *Food and Bioprocess Technology*, vol. 6, no. 7, pp. 1621–1634, 2013, doi: 10.1007/s11947-013-1079-7.
[32] S. Chakraborty, F. M. J. M. Shamrat, M. M. Billah, M. Al Jubair, M. Alauddin, and R. Ranjan, “Implementation of deep learning methods to identify rotten fruits,” *Proceedings of the 5th International Conference on Trends in Electronics and Informatics, ICOEI 2021*, pp. 1207–1212, 2021, doi: 10.1109/ICOEI51242.2021.9453004.
[33] R. Ben Ayed and M. Hanana, “Artificial intelligence to improve the food and agriculture sector,” *Journal of Food Quality*,

vol. 2021, 2021, doi: 10.1155/2021/5584754.
[34] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research," *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pp. 1393–13696, 2020, doi: 10.1609/aaai.v34i09.7123.
[35] F. Valentino, T. W. Cenggoro, and B. Pardamean, "A design of deep learning experimentation for fruit freshness detection," *in IOP Conference Series: Earth and Environmental Science*, 2021, vol. 794, p. 12110. doi: 10.1088/1755-1315/794/1/012110.
[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
[37] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.

BIOGRAPHIES OF AUTHORS

Febrian Valentino currently holds a Bachelor of Computer Science degree obtained from Mercu Buana University in 2019. He started his IT career in the poultry industry in 2019, in the same year he was trusted to be the Head of the IT Department at a finance company to head the Digital Platforms in the Umrah financing division. Currently he is a graduate student at Master of Computer Science in Bina Nusantara University. His research is in the field of artificial intelligence. He can be contacted via email: febrian.valentino@binus.ac.id.

Tjeng Wawan Cenggoro received the bachelor's degree in information technology from STMIK Widya Cipta Dharma, and the master's degree in information technology from Bina Nusantara University. He is currently an AI Researcher whose focus is in the development of deep learning algorithms for application in computer vision, natural language processing, and bioinformatics. He is also a Certified Instructor with the NVIDIA Deep Learning Institute. He led several research projects that utilize deep learning for computer vision, which is applied to indoor video analytics and plant phenotyping. He has published over 20 peer-reviewed publications and reviewed for prestigious journals, such as Scientific Reports and IEEE ACCESS. He also holds two copyrights for AI-based video analytics software. He can be contacted at email: wcenggoro@binus.edu.

Gregorius Natanael Elwirehardja is a research assistant from Bioinformatics & Data Science Research Center of Bina Nusantara University, and a certified instructor at NVIDIA Deep Learning Institute. He completed his bachelor level education in the faculty of Computer Science of Bina Nusantara University. His research interests include applied machine learning in various fields including, but are not limited to, computer vision, mental health, and Natural Language Processing. He can be contacted at email: gregorius.e@binus.edu.

Bens Pardamean received a bachelor's degree in computer science and a master's degree in computer education from California State University, Los Angeles, and a Doctoral degree in informative research from the University of Southern California (USC). He has over 30 years of global experience in information technology, bioinformatics, and education. After successfully leading the Bioinformatics Research Interest Group, he currently holds a dual appointment as the Director of the Bioinformatics and Data Science Research Center (BDSRC), and a Professor of computer science with Bina Nusantara University, Jakarta, Indonesia. He can be contacted at email: bpardamean@binus.edu.