Implementation of Convolutional Neural Network Method for Detecting Vegetables as Recommendation for Vegetarian Food Recipes

Kevin Aliffanova Ardisa, Wahyu Aji Eko Prabowo, Supriadi Rustad

***Abstract*—Vegetarianism is a lifestyle of not consuming food from animals. Even though vegetarianism has become a trend among young people in Indonesia, there is still very little information about vegetarian recipes even including the restaurants that serve them. This study implements the Convolutional Neural Network (CNN) method with MobileNetV2 transfer learning to recommend vegetarian recipes based on vegetable image data. The dataset includes 6 types of vegetable images downloaded from Kaggle. They are labeled, preprocessed, and trained to become a CNN model, converted into tesorflow.js, and implemented on a website-based system. With 1000 images for each vegetable, the model produces an excellent accuracy of 95.78%.**

***Index Terms* — Deep Learning, Convolutional Neural Network, Transfer Learning, MobileNetV2.**

I. Introduction

VEGETARIANISM is a lifestyle that avoids consuming food from animals [1, 2]. Vegetarianism is becoming a trend, and many young people have been following this trend recently. People's changing consumption pattern needs support from their environmental surroundings regarding food and nutrition security. Among ASEAN countries, Indonesia's global vegetarian index is lower than that of Thailand and Malaysia [3]. In this country, it is not an easy way for people to find information on vegetarian food recipes, even in the restaurant that provides them. To identify vegetable images as a vegetarian food recipes recommendation, needs digital image processing. This technology has developed very rapidly and implemented in various fields [4-8]. It can be developed using deep learning [9, 10], which integrates feature extraction and classification in one architecture so that there is no need for separating feature extraction on the data. Duangsuphasin *et al*, designed a decision support system for vegetarian food flavoring by using deep learning methods, which is a multi-layer perceptron neural network model for the aging society [11].

Manuscript received March 30, 2022; revised April 18, 2022.

Kevin Aliffanova Ardisa is with Department of Computer Science, Dian Nuswantoro University, Semarang, Central Java, Indonesia. (e-mail: kevinaliffanovaardisaa@gmail.com).

Wahyu Aji Eko Prabowo is an Assistant Professor at Department of Computer Science, Dian Nuswantoro University, Semarang, Central Java, Indonesia. (Corresponding author to provide phone: +6224 3517261; fax: +6224 3569684; e-mail: prabowo@dsn.dinus.ac.id).

Supriadi Rustad is a Professor at Department of Computer Science, Dian Nuswantoro University, Semarang, Central Java, Indonesia. (e-mail: srustad@dsn.dinus.ac.id).

The convolutional neural network is the development of multi-layer perceptron neural network, which is designed to process two-dimensional image data. Because of its high network depth, CNN enters the deep neural network, which causes CNN to be often applied to image datasets [12, 13]. CNN has the advantage of parameter sharing, which can help reduce the number of parameters in the entire system and reduce the computational load. CNN also has the advantage of spatial features. This refers to the arrangement of pixels and the relationship between pixels in an image. Many CNN models are used to classify VGG16, MobileNet, ResNet, and others [14-16]. This study implements a convolutional neural network method to encourage finding the right vegetarian food recipes easier for users.

II. Fundamentals and Related Studies

This section overviews states of the art in the major areas. The areas are convolutional neural network with transfer learning, MobileNetV2 architecture, and confusion matrix.

A. Convolutional Neural Network with Transfer Learning

According to O'Shea *et al*., [12], CNN is analogous to a traditional artificial neural network (ANN) because it consists of neurons that optimize themselves through the training stage. Each neuron receives input and performs the basic operations of countless ANN. The entire network expresses the weight score function from the raw image vector input to the final class score output. The last layer contains the loss function associated with the class. The most noticeable difference between CNN and ANN is that CNN is used in the pattern recognition field in the image. This allows researchers to create an architecture and suitable networks for image-focused tasks [17, 18]. The research conducted by Gopalakrishnan *et al*., [19] has proven that CNN is very effective in processing visual data, such as images and videos. CNN takes raw input data at the lowest level and transforms the data by processing them through a sequence of basic computing units for the highest layer. This method usually requires large and annotated image datasets to achieve high prediction accuracy. Huang *et al*., [10] show that transfer learning can solve the collecting training data problem by transferring knowledge from a large dataset which known as source domain to smaller dataset commonly known as target domain. Transfer learning using CNN is commonly used in various fields [21] by demonstrating trained layers on ImageNet.

B. *MobileNetV2 Architecture*

MobileNetV2 is a CNN architecture performed on mobile devices, introduced by Sandler *et al.*, [22]. The MobileNetV2 comes from an inverted residual structure where the residual connections are between the bottleneck layers. Sandler *et al.*, show that MobileNetV2 improves the performance of the mobile model on multiple tasks. The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. The architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers. In contrast to MobileNetV1, the pointwise convolutional layer in MobileNetV2, known as the projection layer of data with a high number of channels, is placed into a tensor with a much smaller number of channels [23].

C. *Confusion Matrix*

Confusion Matrix [24] is a machine learning approach that accommodates information about the actual and predicted classification. The confusion matrix is indexed by the actual class of an object and the class predicted by the classifier [25]. Figure 1 shows the confusion matrix, where the classification results are divided into two types, namely positive and negative tuples. It consists of 4 matrix elements, namely True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN).

		Predicted	
		Positive	**Negative**
Actual	**Positive**	TP	FN
	Negative	FP	TN

Fig. 1. The Confusion Matrix

The classification performance measures such as precision, recall, accuracy, and F1-score can be determined based on the confusion matrix. Some general calculations are given as follows. Equation (1) calculates the average precision value based on the number of TPs and FPs.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

Equation (2) calculates the average recall value based on the number of TPs and FNs.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

Equation (3) calculates the average accuracy value to show the level of effectiveness of the classification model. Accuracy is a good reference when the output data is symmetrical.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

Equation (4) calculates the average value of the F1-score, using precision and recall values.

$$\text{F1-score} = \frac{2 \times PRECISION \times RECALL}{PRECISION + RECALL} \tag{4}$$

III. Research Methodology

A. *Data Preparation*

The dataset used is images of 6 classes of vegetables downloaded from Kaggle. They are Spinach, Corn, Long Beans, Kale, Eggplant, and Cucumber. Each class consists of 1000 images then the total dataset is 6000 vegetable images. Table 1 shows data separation scenario where 70%, 15%, and 15% of the dataset respectively were used for training, testing, and validation.

TABLE I
Data Separation Scenario

Training Data	Testing Data	Validation Data
70%	15%	15%
700	150	150

The initial step of data preparation is resizing all downloaded images into 900 × 900 format. The next step is to label images based on its class in the Google Colaboratory and then split them by classes. The input size is determined to be (224, 224), as the one often used in ImageNet. Figure 2 shows the results of the image visualization that has been done by data preparation.

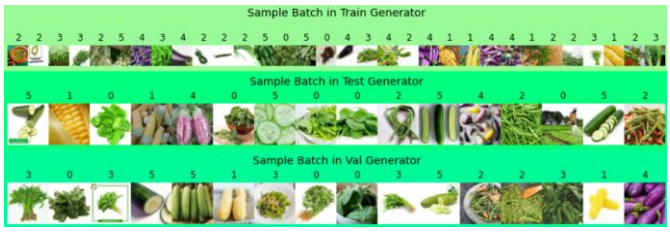

Fig. 2. Image visualization after data preparation

After determining the input size, each image is rescaled to 1./255, rotated to 40 degrees, sheared to 0.2 degrees, zoomed to 0.2, and flipped in a horizontal image.

B. *Implementation and Training*

The MobileNetV2 with a transfer learning architecture is used to train the data. Figure 3 shows the feature extraction process where the dense layer uses a softmax activation function to classify the image into 6 classes. The input size is 224 × 224 × 3, where 224 refers to the length and width of the image, while 3 indicates three color channels, namely Red, Green, and Blue (RGB).

ISBN: 978-988-14049-3-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

	Input Layer	Input	224, 224, 3
		Output	224, 224, 3
Weights = imagnet, trainable = true	MobileNet V2	Input	224, 224, 3
		Output	7, 7, 1280
	GlobalAverage Pooling	Input	7, 7, 1280
		Output	1280
6 Neuron (according to 6 vegetables class)	Dense + Softmax	Input	1280
		Output	6

Fig. 3. The feature extraction process is carried out by compiling the model, GlobalAveragePooling2D and dense into sequential.

This study uses 50 epochs and the Adam Optimizer [26] with a learning rate of 0.00001. The training stops at the 18th epoch as it has obtained an accuracy of more than 0.95. Figure 4 presents the training-validation accuracy (top) and loss values (bottom). It is shown that the testing runs well because for greater epochs, the accuracy value continues to increase while the loss value decreases.

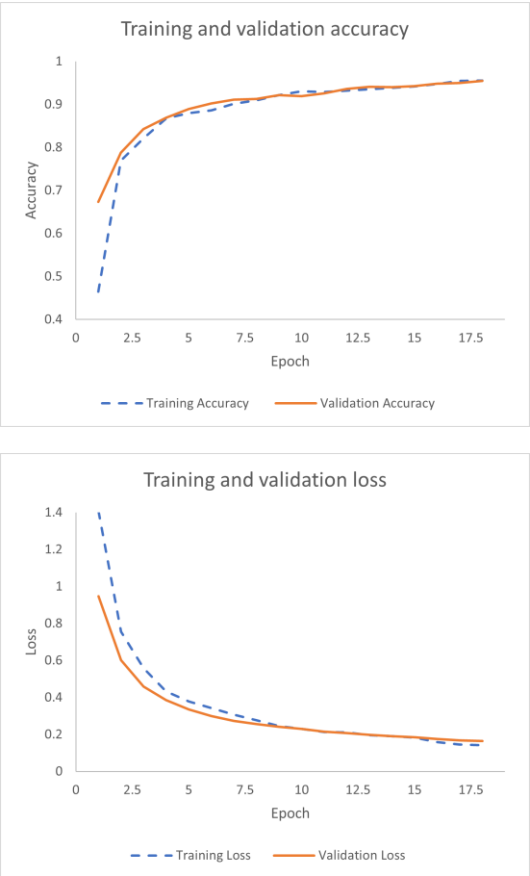

Fig. 4. Training-validation accuracy (top) and loss (bottom)

IV. Results and Discussion

A. Testing Accuracy with Confusion Matrix

The accuracy testing is done to get the percentage of accuracy obtained from classifying the correct vegetable image data. Figure 5 shows the results of the model classification of six classes of vegetable image data.

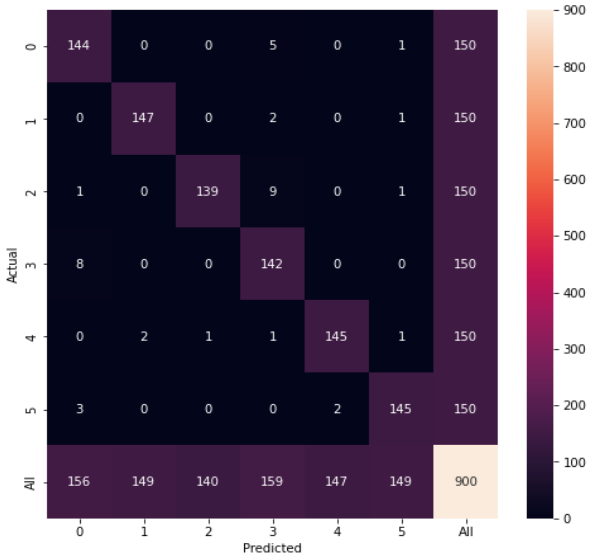

Fig. 5. The result of model classification of 6 classes of vegetable image data. Numbers 0, 1, 2, 3, 4, 5 are labels for Spinach, Corn, Long Beans, Kale, Eggplant, and Cucumber, respectively.

In general, the classification works well. From 150 images for each vegetable, 144, 147, 139, 142, 145, and 145 images are successfully classified for Spinach, Corn, Long Beans, Kale, Eggplant, and Cucumber, respectively. The maximum value of False Negative (FN) is about 7%.

From the confusion matrix, one can calculate precision, recall, and F1-score according to equation 1, 2, and 4. Table 2 shows the precision, recall, and F1-score values for each vegetable image class. It is shown that almost all precision, recall, and F1-score values are above 90%, except the precision of Kale. The classification accuracy for all classes of vegetables is calculated based on equation 3 and the result is 95.78%.

TABLE II
Classification Performance

Num	Vegetable Type	Precision	Recall	F1-score
1	Spinach	0.92	0.96	0.94
2	Corn	0.99	0.98	0.98
3	Long Beans	0.99	0.93	0.96
4	Kale	0.89	0.95	0.92
5	Eggplant	0.99	0.97	0.98
6	Cucumber	0.97	0.97	0.97

B. Implementation on Website

The results of the CNN model are stored in the .h5 model. This model then converted into TensorFlow js (.tfjs). The TensorFlow Js is implemented in the "veggiediaries" website (https://the-veggiediaries.000webhostapp.com/).

ISBN: 978-988-14049-3-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The website has a main menu to display information about the website, food recipes, and vegetable detection. Several menus are provided on the vegetable detection page, namely main, upload, and camera for capturing images in real-time.

C. Website Testing

The Convolutional Neural Network method is tested on a website system. This test shows that the Convolutional Neural Network (CNN) method and the system have been running well and can recommend vegetarian food recipes. Table 3 shows the validity of website system in recognizing vegetable images from smartphone camera with 12 megapixels in .jpg format with a size of 3024 × 3024. It is clear that the system successfully recognizes all classes of images and may recommend vegetarian food recipes.

TABLE III
SYSTEM TEST RESULT

Num	Vegetable Type	Input image	Results	Description
1	Spinach			Valid
2	Corn			Valid
3	Long Beans			Valid
4	Kale			Valid
5	Eggplant			Valid
6	Cucumber			Valid

V. Conclusion

The study concludes that the convolutional neural network method with MobileNetV2 transfer learning has excellent results in detecting all vegetable image. The level of accuracy reaches a value of 95.78%. This study also produces a website-based application called "veggiediaries" to detect vegetable images and recommend vegetarian food recipes to users.

References

[1] Matthew B. Ruby, "Vegetarianism. A Blossoming field of study", *Appetite*, pp. 141-150, 2012.
[2] Daniel L. Rosenfeld, "The psychology of vegetarianism: Recent advances and future directions", *Appetite*, pp. 125-138, 2018.
[3] Wowshack Team, "Indonesia is the 16th most vegetarian-friendly country in the world", 2021. Available: https://www.wowshack.com/indonesia-vegetarian-friendly/
[4] R. Maini and H. Aggarwal, "A comprehensive review of image enhancement techniques", *Journal of Computing*, pp. 8-13, 2010.
[5] Ashraf A. Aly, Safaai B. Deris, Nazar Zaki, "Research review for digital image segmentation techniques", *International Journal of Computer Science & Information Technology*, pp. 99-106, 2011.
[6] S. Dutta, S. K. Pal, S. Mukhopadhyay, R. Sen, "Application of digital image processing in tool condition monitoring: A review", *CIRP Journal of Manufacturing Science and technology*, pp. 212-232, 2013.
[7] J.G. Arnal Barbedo, "Digital image processing techniques for detecting, quantifying and classifying plant diseases", *SpringerPlus*, pp. 1-12, 2013.
[8] Ajay K. Boyat and Brijendra K. Joshi, "A review paper: Noise models in digital image processing", *Signal and Image Processing: An International Journal*, pp. 63-75, 2015.
[9] Y. Le Cun, Bengio, Y., Hinton, G., "Deep Learning", *Nature*, pp. 436-444, 2015.
[10] Y. Guo, Y. Liu, A. Oerlemans, S. Leo, S. Wu, Michael S. Lew, "Deep learning for visual understanding: A review", *Neurocomputing*, pp. 27-48, 2016.
[11] A. Duangsuphasin, A. Kengpol, and R. M. Lima, "Design of a decision support system for vegetarian food flavoring by using deep learning for the ageing society," *2021 Research, Invention, and Innovation Congress: Innovation Electricals and Electronics*, pp. 54-59, 2021.
[12] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," pp. 1–11, 2015. Available: http://arxiv.org/abs/1511.08458.
[13] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, pp. 1-6, 2017.
[14] T. Agarwal and H. Mittal, "Performance Comparison of Deep Neural Networks on Image Datasets," *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pp. 1-6, 2019.
[15] L. Liu *et al*., "Deep-learning and depth-map based approach for detection and 3-D localization of small traffic signs," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 2096-2111, 2020.
[16] Soumya R. Nayak *et al*., "Application of deep learning techniques for detection of COVID-19 cases using X-ray images: A comprehensive study", *Biomedical Signal Processing and Control*, pp. 1-12, 2020.
[17] Sahu, M., Dash, R. "A Survey on Deep Learning: Convolution Neural Network (CNN)", *Intelligent and Cloud Computing. Smart Innovation, Systems and Technologies*, Vol. 153. *Springer*, 2021.
[18] Nath, S., Naskar, R. "Automated image splicing detection using deep CNN-learned features and ANN-based classifier", *Signal, Image and Video Processing (SIViP)* pp.1601–1608, 2021.
[19] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, "Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection," *Constr. Build. Mater.*, pp. 322–330, 2017.
[20] Z. Huang, Z. Pan, and B. Lei, "Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data," *Remote Sens.*, pp. 1–21, 2017.
[21] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1717–1724, 2014.
[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018.
[23] A. Michele, V. Colin, and D. D. Santika, "Mobilenet convolutional neural networks and support vector machines for palmprint recognition", *Procedia Computer Science*, pp. 110–117, 2019.
[24] C. Sammut and G.I. Webb, "Encyclopedia of Machine Learning", *Springer*, 2011.
[25] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Information Sciences*, pp. 250–261, 2016.
[26] Kingma, D.P. and Ba, J., "Adam: a Method for Stochastic Optimization", *The 3rd International Conference for Learning Representations*, pp. 1-15, 2015.