Journal of Information Hiding and Multimedia Signal Processing ©2023 ISSN 2073-4212
Ubiquitous International Volume **14**, Number **1**, March **2023**

Proposing a fruit classification system using Tensorflow model

Phat Nguyen Huu* and Thuong Nguyen Thi Mai

Department of Electronics
School of Electronic and Electrical Engineering
Hanoi University of Science and Technology, Hanoi, Vietnam
Email: phat.nguyenhuu@hust.edu.vn; thuong.ntm164021@sis.hust.edu.vn

Received December 2020; revised February 2023

Abstract. *Computer vision is an important area of artificial intelligence that aims to help it gain the ability to activate similar to humans. In the past, we often classify fruit by hand. Today, it is performed by the development of image-processing technology. When the quantity of fruits is huge, we needs machine learning for classifying them. Therefore, we propose the fruit classification using Tensorflow and Keras model (high-level framework of Tensorflow) in the paper. This is a simple problem of computer vision since it solves the basis problems such as object detection or face recognition. In the paper, we focus on modifying the network architecture of Tensorflow model. As a result, accuracy of proposal model achieves 99% with only five epochs.*

Keywords: Tensorflow, deep learning, accuracy, fruit classification, convolutional neural network.

1. **Introduction.** Currently, the machine learning model is applied in many fields such as facial recognition, handwriting processing, or object tracking [1–5]. One of the growing interesting areas is object classification. Although it is only a small subset of computer vision, its range of applications is huge. It underpins a lot of important applications. Therefore, we propose to build a fruit classification system using Tensorflow model in the paper. The simulation results show that the model has high accuracy and can be applied for real environments.

In the paper, there are three points that we propose as follows. Firstly, we change the network architecture according to the requirements. Secondly, we propose the fruit classification system using Tensorflow model. Finally, we modify Kaggle dataset that are suitable for real applications. As a result, accuracy of proposal model is improved up to 99%.

The rest of the paper is presented as follows. In Section II, we will present related work. In Section III and IV, we present and evaluate the effectiveness of the proposed model, respectively. Finally, we give conclusion in Section V.

2. **Related work.** There are many studies using image processing techniques in general classification problems with the development of artificial intelligence (AI) and computer vision [1, 6–15]. In [6], the authors propose a fruit classification method based on convolutional neural network (CNN). The results show that the method achieved the average

*P. N. Huu is corresponding author at School of Electronic and Electrical Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam (email: phat.nguyenhuu@hust.edu.vn).

classification accuracy up to 99.8%. Otherwise, the authors [7] propose the fruit classification method based on fruits-360 dataset. However, the accuracy of method is only 98.60%. In [8], the authors propose a fruit recognition system using deep convolutional neural network (DCNN). The authors use 17,823 images from 25 different categories with accuracy of 99.79%.

Besides, using CNNs in agriculture is developing for many applications [1,9–16]. In [16], the authors compare 21 methods and show that using AlexNet model for obstacle detection has accuracy of 99.9%. In [9], the authors propose method using deep-learning techniques for feature extraction and classification. However, accuracy of method is only 75% for 43 different types of fruit. In [10], the authors propose the automation of fruit classification system based on CNN. The accuracy rate of system is 98% for 180 and 20 images using for training and testing. The authors [12] use a pure convolutional neural network (PCNN) with seven parameters and achieve a classification accuracy of 98.88%. The authors [1] propose to use image saliency for efficient fruit and vegetables classification system. The results show that the system achieves an accuracy rate of 95.6%.

However, the disadvantages of systems all use complex network configurations. This is difficult to apply for real applications. Therefore, we propose to use the Tensorflow model with the simplest configuration to apply for practice in the paper. The results show that the proposal system can achieve up to 99.9% accuracy.

3. **Proposal system.**

3.1. **Overview of system.** Our system is proposed based on [10] as shown in Fig. 1. In the model, we focus on modifying the structure of model as follows.

- **Filters**:
  If a convolution has more kernels, the more possible learn many features. If we choose multiple kernels, it can lead to over-fitting. Therefore, we choose the corresponding number of kernels as 16, 32, 64, and 128 that are suitable for our application.
- **Size of kernel**
  It has some of the following characteristics as follows:
  The larger the size, the larger the number of features can be detected. Otherwise, the number of detecting features will be smaller. If the kernel size is large, the training time will be long. Therefore, we choose equal to 2.
- **Padding parameter**
  It depends on choosing the *same* or *valid* parameter. If we choose *same*, it will keep the size of network after convolution. If we choose *valid*, the output of convolution will be calculated according to [17–19]. Our aim is to choose parameters that focus on reducing training time but still ensuring output. Therefore, we select the *same* parameter.

Besides, we also perform pre-processing of input data by changing the distribution of layers to ensure more balance. Finally, we also change the ratio of train and test datasets to 5091 and 1689 to improve the accuracy of model, respectively.

In the proposal system, we use a fruits (360 datasets) collecting from Kaggle [20]. The collection of this dataset is described as follows:
Fruits and vegetables are grown in the shaft of low-speed motor and a short 20-second video clip is recorded. Logitech C920 camera is used to shoot them. We put a blank sheet of paper as the background of fruits.

However, the background is not uniform due to variations in lighting conditions. Therefore, we use a dedicating algorithm that extracts fruit from background. In the algorithm, we start at each edge of the image and we mark all pixels. We then highlight all pixels in

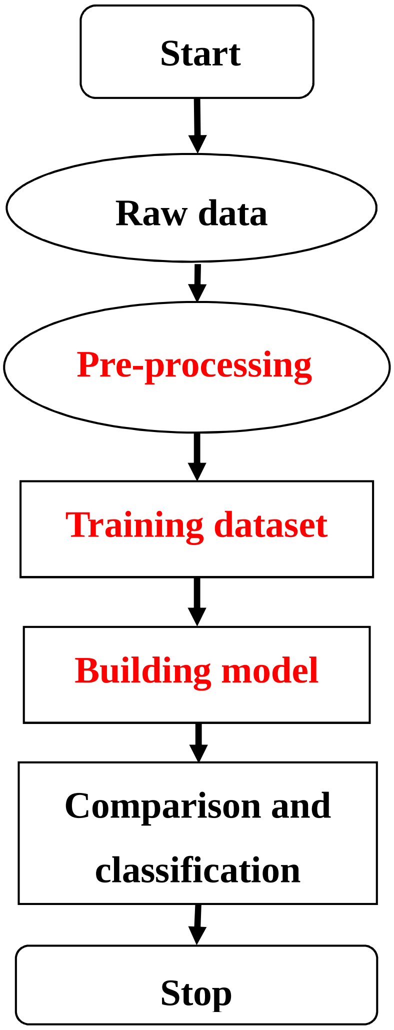

Figure 1. Proposal system diagram.

its vicinity whose colors is less than a specified value. We repeat the previous step until we are not able to highlight the extra pixels.

All highlighted pixels are considered background (filling with white) and remaining pixels are belonging to the object. The maximum value for distance between two neighboring pixels is a parameter of algorithm that is setup for each video. We use CNN to solve classification problem in the step.

Although classification is a simple computer vision problem, it is the backbone of large problems as object detection or face recognition. Therefore, it will be very feasible when applying for real applications.

3.2. **Building a model.**

3.2.1. *Preparing data.* In the step, we will use Google Colab to train through the google drive account. We compress dataset into zip file to reduce the processing time.

After uploading them to Google drive, we connect to Colab. We then unzip "file.zip" to perform the processing data step.

Besides, we use Apple libraries for training model. In the model, we use 6780 images for training and testing process.

To improve the processing time and accuracy, we use Tensorflow and Keras (this is a high-level framework by Tensorflow). Details of processing_data are as follows:

TABLE 1. Result of total number of labels and names for Tensorflow and Keras.

No.	Name	Label class
0	Class 0	Apple Braeburn
1	Class 1	Banana
2	Class 2	Cherry 2
3	Class 3	Kaki
4	Class 4	Kiwi
5	Class 5	Lemon
6	Class 6	Potato Red
7	Class 7	Tomato 3
8	Class 8	Walnut

In the pre-processing step, we use the load_files command in sklearn to load the path to the dataset containing image. It returns data frame containing three columns (filename, target, target_names) where filename contains absolute path to image, target contains the actual labels of image files, and target_names are names of all labels.

We use the "np.array" command to convert the list into an array to be processed. In the step, we perform with the number of images to train and test as 5091 and 1689.

In Tab. 1, the result shows the total number of labels and names for nine classes to use for training and testing model.

Besides, we use the seaborn and pandas libraries to estimate the balance of classes. The results are as shown in Fig. 2.

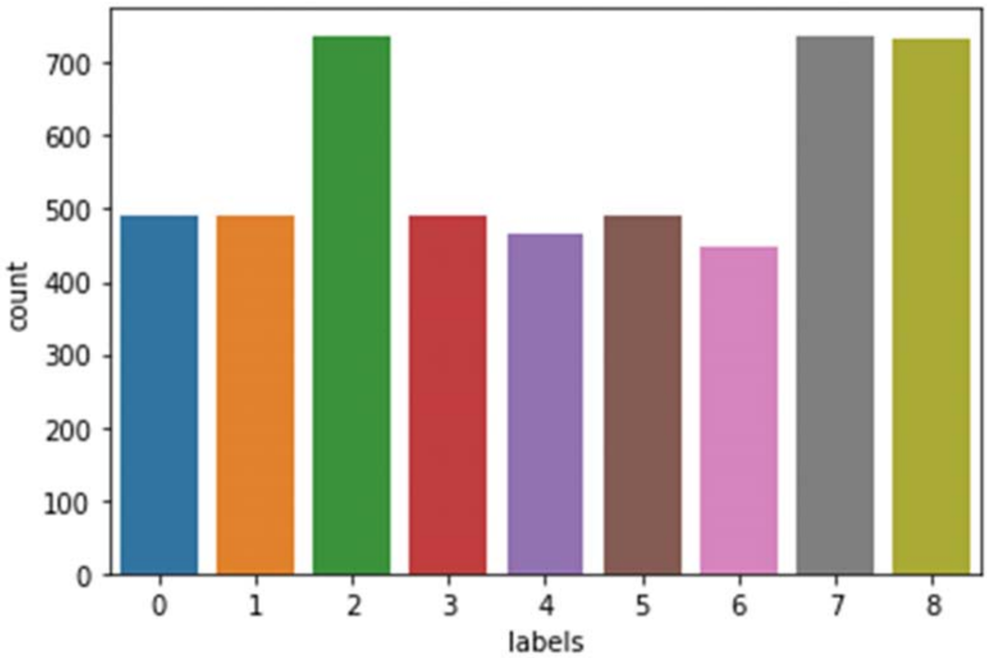

FIGURE 2. Estimating the balance of classes.

The results show that the classes are not out of balance.

Next, we will read the data into an array and divide the train and validation datasets using sklearn of train_test_split to manipulate the hyparameter (batch_size, lr, etc.) while training the model. In the model, we modify the parameters as $test_{size} = 0.2$ and $random_{state} = 320$.

TABLE 2. Input and output parameters for layers.

No.	Layer (type)	Output Shape	Number of parameters
1	Conv2D_1	(None, 50, 50, 32)	2080
2	Max_Pooling2d_1	(None,25,25,32)	0
3	Conv2D_2	(None,25,25,64)	8256
4	Mish_2	(None,25,25,64)	0
5	Max_Pooling2d_2	(None,12,12,64)	0
6	Conv2D_3	(None,12,12,128)	32896
7	Mish_3	(None,12,12,128)	0
8	Max_pooling2d_3	(None,6,6,128)	0
9	Dropout	(None,6,6,128)	0
10	Flatten	(None,4608)	0
11	Dense	(None,150)	691350
12	Mish_4	(None,150)	0
13	Dropout_1	(None,150)	0
14	Dense_1	(None,9)	1359

We also perform to preprocess the images before training. The results are shown in Fig. 3.

FIGURE 3. Preprocessing input data before training.

We next have to normalize the data since the training will not be affected by the histogram. Finally, we use activation as *mish* function (nonlinear function synthesis provides higher performance than ReLU function) and Tensor board to track model during training. Results are store in log file of program.

3.2.2. *Model building and training.* In the model, we use filters 16, 32, 64, 128 and activation function is $softmax$. The number of of parameters using in the model is 736,149. The parameters for models are shown in Tab. 2.

In our model, we use the optimal algorithm Adam and the cost function as $bacth_{size} = 32$ and $epochs = 20$. Results are shown in Tab. 3.

4. **Result.** To evaluate the accuracy of model, we experiment with the dataset above. The results of training accuracy are shown in Fig. 4. As shown in Fig. 4, we see that the accuracy of model reaches 100% when performing to the 7^{th} epoch.

Loss function results for training are shown in Fig. 5. In the Fig. 5, we see that it reaches 0 at the first epoch.

To check the details of model operation, we set $verbose = 1$. The results show that the model will reach saturation with 53 epochs. The result show that the accuracy of model is up to 99% as shown in Tab. 4.

To evaluate the resulting accuracy and coverage of the predictions for each class we use the confusion matrix. A confusion matrix of four indices (TP (True Positive),TN (True

TABLE 3. Result of Adam optimal algorithm.

Epoch	Time for step (second)	Loss training	Accuracy training	Loss testing	Accuracy testing
1	19	1.6069	0.4887	0.4814	0.9068
2	11	0.3503	0.8934	0.1353	0.9706
3	11	0.1511	0.9563	0.0383	0.9951
4	11	0.0800	0.9799	0.0184	0.9961
5	11	0.0462	0.9904	0.0111	0.9971
6	11	0.0323	0.9931	0.0079	0.9980
7	11	0.0234	0.9948	0.0056	0.9980
8	11	0.0180	0.9958	0.0033	1.0000

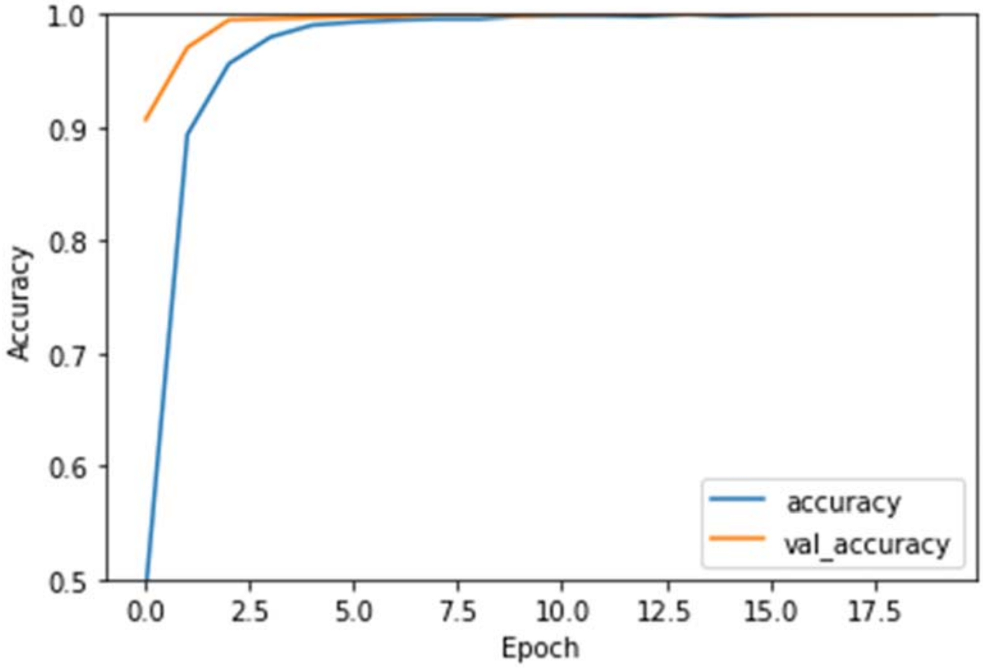

FIGURE 4. Result of accuracy training.

TABLE 4. Accuaracy of model at the 53^{th} epoch.

Epoch	Time for step (second)	Loss training	Accuracy training	Loss testing	Accuracy testing
53	6	0.0500	0.9858	0.0499	0.9858

Negative), FP (False Positive), and N (False Negative)) for each classifier [21,22]. Results that are shown in Fig. 6 indicate a high degree of accuracy between the predicting and actual labels.

Besides, we also save and predict several images to perform for the next time. They are stored and predicted in data frame. Besides, we evaluate score and predict class output. Results are shown in Tab. 5.

We perform to compare with other model [10] with the similar dataset. In [10], the authors use 200 fruits for one type where ratio of training and testing is 9:1. The results are shown in Figs. 7 and 8.

Based on Figs. 7 and 8, it shows that our model is better than that of [10] in terms of accuracy and loss valuable.

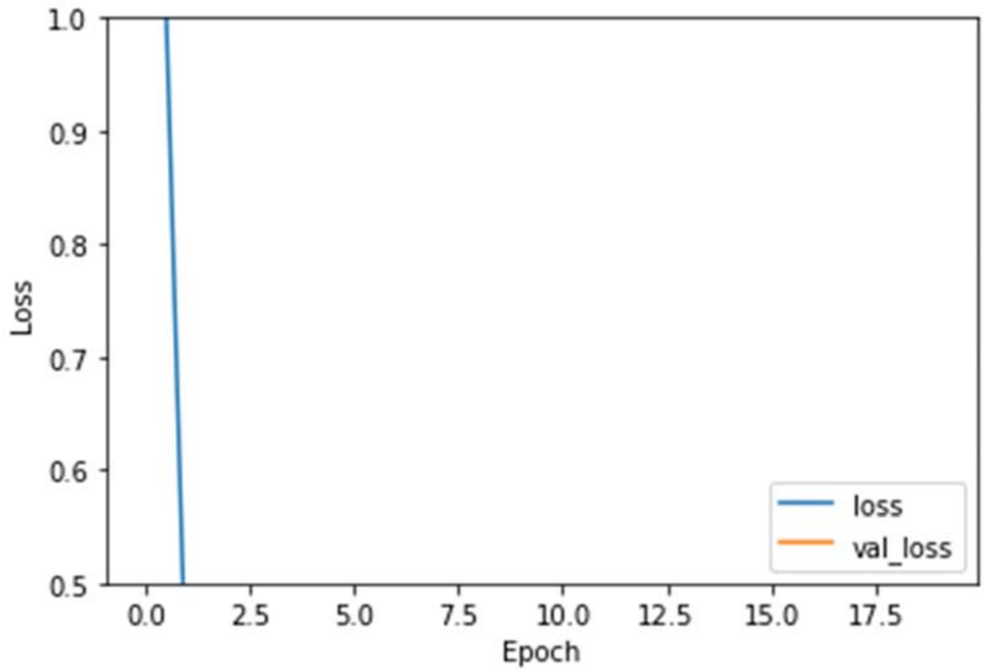

FIGURE 5. Result of loss training.

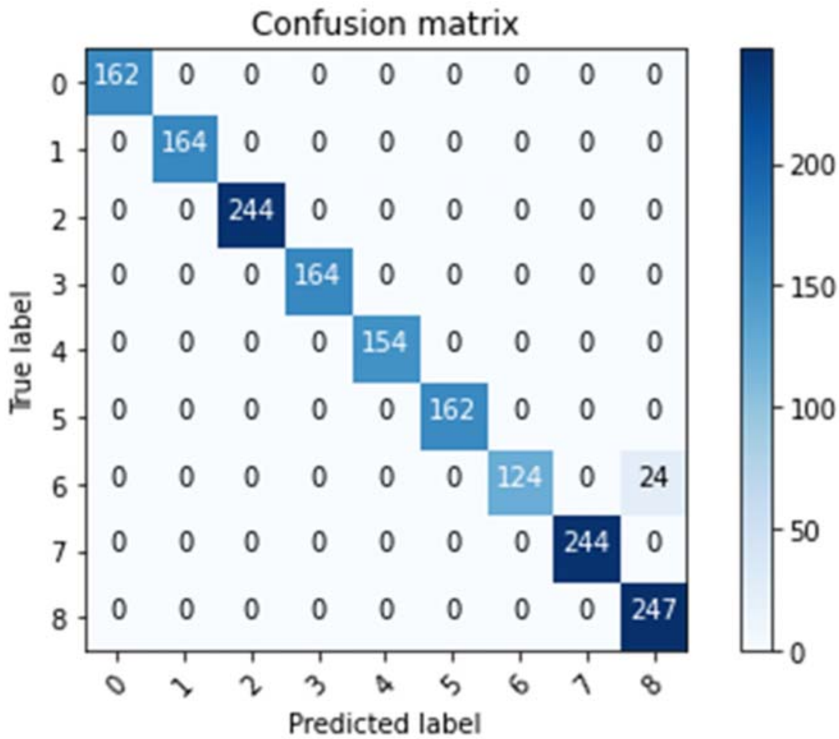

FIGURE 6. Evaluating accuracy between the prediction and actual label.

Table 5. Evaluating score and predict class of output.

No.	Filename	Score	Predicting label
0	100_100.jpg	0.999996	Walnut
1	103_100.jpg	0.999998	Walnut
2	183_100.jpg	1.000000	Potato Red
3	232_100.jpg	0.999994	Cherry 2
4	233_100.jpg	0.999993	Cherry 2
5	250_100.jpg	1.000000	Kaki
6	252_100.jpg	1.000000	Kaki
7	292_100.jpg	0.999998	Tomato 3
8	293_100.jpg	0.999998	Tomato 3
9	320_100.jpg	0.999483	Banana
10	321_100.jpg	0.999817	Banana
11	327_100.jpg	0.999995	Lemon
12	61_100.jpg	1.000000	Apple Braeburn
13	62_100.jpg	1.000000	Apple Braeburn
14	63_100.jpg	0.999952	Kiwi
15	64_100.jpg	0.999948	Kiwi
16	r_0_100.jpg	0.999878	Potato Red
17	r_2_100.jpg	0.998811	Lemon

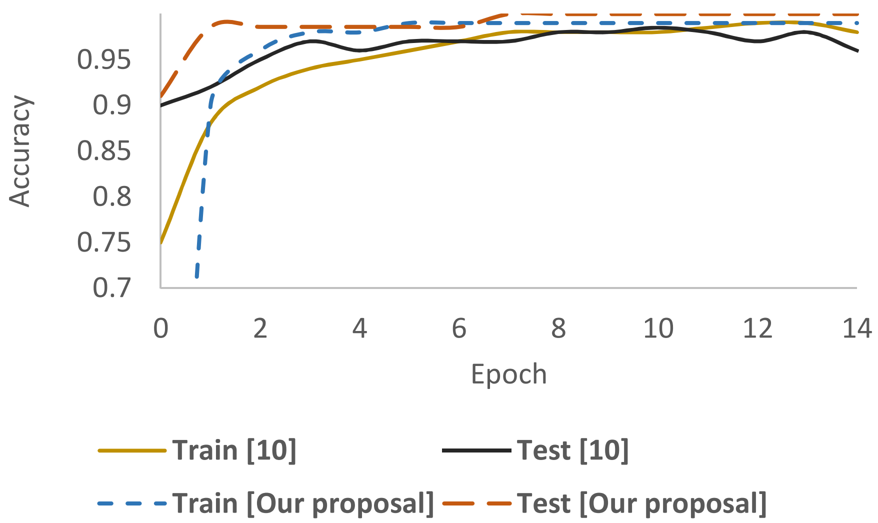

Figure 7. Comparing accuracy with [10].

5. **Conclusions.** The paper proposed the fruit classification model using the Tensorflow model. The results show that the accuracy of the model increases up to 99% when the number of epochs increases. However, the processing time of the model is still large at 5 milliseconds per step. We have tackled the multi-layered classification problem. This is a premise to solve more difficult problems such as gesture and face recognition as well as classification of different actions.

In the future, we therefore will improve the model and combine with our model [23] to increase the accuracy and processing time.

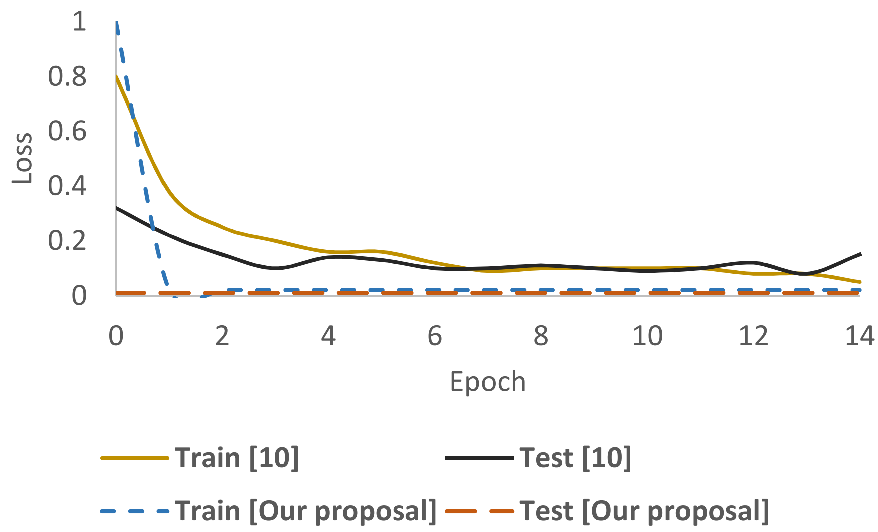

FIGURE 8. Comparing loss with [10].

Acknowledgment. This research is funded by the Hanoi University of Science and Technology (HUST) under project number T2022-PC-013. The authors would like to thank HUST for the financial support.

REFERENCES

[1] G. Zeng, *Fruit and vegetables classification system using image saliency and convolutional neural network*, Oct. 2017, pp. 613–617.

[2] G. Zeng, *Fruit and vegetables classification system using image saliency and convolutional neural network*, in *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2017, pp. 613–617.

[3] Seema, A. Kumar, and G. S. Gill, *Automatic fruit grading and classification system using computer vision: A review*, in *2015 Second International Conference on Advances in Computing and Communication Engineering*, 2015, pp. 598–603.

[4] H. Dang, J. Song, and Q. Guo, *A fruit size detecting and grading system based on image processing*, in *2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2, 2010, pp. 83–86.

[5] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, *Automatic fruit classification using deep learning for industrial applications*, *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1027–1034, 2019.

[6] L. Wu, H. Zhang, R. Chen, and J. Yi, *Fruit classification using convolutional neural network via adjust parameter and data enhancement*, in *2020 12th International Conference on Advanced Computational Intelligence (ICACI)*, 2020, pp. 294–301.

[7] D. Zhu, M. Wang, Q. Zou, D. Shen, and J. Luo, *Research on fruit category classification based on convolution neural network and data augmentation*, in *2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, 2019, pp. 46–50.

[8] S. Sakib, Z. Ashrafi, and M. A. B. Siddique, *Implementation of fruits recognition classifier using convolutional neural network algorithm for observation of accuracies for various hidden layers*, 2020.

[9] M. Akbari Fard, H. Hadadi, and A. Tavakoli Targhi, *Fruits and vegetables calorie counter using convolutional neural networks*, in *Proceedings of the 6th International Conference on Digital Health Conference*, ser. DH '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 121–122.

[10] M. Khatun, M. F. Ali, N. Turzo, J. Nine, and P. Sarker, *Fruits classification using convolutional neural network*, Ph.D. dissertation, 07 2020.

[11] L. Hou, Q. Wu, Q. Sun, H. Yang, and P. Li, *Fruit recognition based on convolution neural network*, in *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2016, pp. 18–22.

[12] A. Kausar, M. Sharif, J. Park, and D. R. Shin, *Pure-cnn: A framework for fruit images classification*, in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2018, pp. 404–408.

[13] Z. M. Khaing, Y. Naung, and P. H. Htut, *Development of control system for fruit classification based on convolutional neural network*, in *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 2018, pp. 1805–1807.

[14] S. Lu, Z. Lu, S. Aok, and L. Graham, *Fruit classification based on six layer convolutional neural network*, in *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, 2018, pp. 1–5.

[15] R. Yamparala, R. Challa, V. Kantharao, and P. S. R. Krishna, *Computerized classification of fruits using convolution neural network*, in *2020 7th International Conference on Smart Structures and Systems (ICSSS)*, 2020, pp. 1–4.

[16] A. Kamilaris and F. Prenafeta Boldú, *A review of the use of convolutional neural networks in agriculture*, *The Journal of Agricultural Science*, vol. 156, pp. 1–11, June 2018.

[17] A. Shrestha and A. Mahmood, *Review of deep learning algorithms and architectures*, *IEEE Access*, vol. 7, pp. 53 040–53 065, 2019.

[18] Q. Guan, P. W. Wheeler, Q. Guan, and P. Yang, *Common-mode voltage reduction for matrix converters using all valid switch states*, *IEEE Transactions on Power Electronics*, vol. 31, no. 12, pp. 8247–8259, 2016.

[19] D. Nix and A. Weigend, *Estimating the mean and variance of the target probability distribution*, in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 1, 1994, pp. 55–60 vol.1.

[20] H. Mureşan and M. Oltean, *Fruit recognition from images using deep learning*, 2021.

[21] M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto, *Evaluating trust prediction and confusion matrix measures for web services ranking*, *IEEE Access*, vol. 8, pp. 90 847–90 861, 2020.

[22] N. D. Marom, L. Rokach, and A. Shmilovici, *Using the confusion matrix for improving ensemble classifiers*, in *2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel*, 2010, pp. 000 555–000 559.

[23] P. N. Huu and H. L. The, *Proposing recognition algorithms for hand gestures based on machine learning model*, in *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*, 2019, pp. 496–501.