

Research Article

Muskmelon Maturity Stage Classification Model Based on CNN

Huamin Zhao , Defang Xu , Olarewaju Lawal , and Shujuan Zhang 

College of Agricultural Engineering, Shanxi Agricultural University, Taigu County, Jinzhong City, Shanxi Province 030801, China

Correspondence should be addressed to Olarewaju Lawal; olarewajulawal@yahoo.com

Received 22 August 2020; Revised 26 January 2021; Accepted 7 August 2021; Published 17 August 2021

Academic Editor: Yaoyao Wang

Copyright © 2021 Huamin Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to quickly and accurately judge the maturity of muskmelon is very important to consumers and muskmelon sorting staff. This paper presents a novel approach to solve the difficulty of muskmelon maturity stage classification in greenhouse and other complex environments. The color characteristics of muskmelon were used as the main feature of maturity discrimination. A modified 29-layer ResNet was applied with the proposed two-way data augmentation methods for the maturity stages of muskmelon classification using indoor and outdoor datasets to create a robust classification model that can generalize better. The results showed that code data augmentation which is the first way caused more performance degradation than input image augmentation—the second way. This established the effectiveness of the code data augmentation compared to image augmentation. Nevertheless, the two-way data augmentations including the combination of outdoor and indoor datasets to create a classification model revealed an excellent performance of F_1 score $\sim 99\%$, and hence the model is applicable to computer-based platform for quick muskmelon stages of maturity classification.

1. Introduction

Muskmelon fruit offers essential nutrients and several health benefits to man. It matures more quickly in moist and warm weather than in cool conditions. The color of muskmelon fruit turns from green to yellow during maturity stages, making it very difficult for those visually impaired people, supermarket or grocery workers, farm workers, children, and so on to classify their maturity stages. Generally, the wide variety, irregular shape, color, and texture characteristics have always been a relatively complicated problem associated with fruit classification [1]. For example, fruits' visual inspection requires trained employers that are familiar with the unique characteristics of fruits for classification [2], which are labor-intensive and relatively expensive. Therefore, a classification model that is applicable to mobile phone and automatic sorting system is required to replace the traditional way of classification. This will promote self-inspection, speedy inspection, packaging, and transportation systems of muskmelon.

Deep learning with convolutional neural networks (CNNs) is a tool widely used for image analysis and image classification, allowing recognition and detection of objects wherever they are positioned in an image. Also, it can extract

complex features automatically and directly from the input images [3]. Ananya et al. [4] used CNN to study the maturity classification model of apple, mango, and other five fruits. The results of the classification accuracy of the retrieval of 50 images have a precision value of 88.93%. Saranya et al. [5] studied the classification method of banana maturity based on CNN. The CNN model was trained with overall validation accuracy of 96.14%. Tu et al. [6] studied tomato detection and maturity classification model based on Fast-RCNN and RGB-D images, which achieves 92.71% detection accuracy and 91.52% maturity classification accuracy. Liu et al. [7] also studied the maturity classification model of tomato. Yueju et al. [8] studied the detection of immature mango based on yolov2. In addition, Parvathi and Tamil Selvi [9] studied the maturity classification model of coconut, Huanghua pears [10], fresh tea shoot [11], and other agriculture products by using the method of deep learning and obtained good accuracy. The above research results show that deep learning has excellent performance in fruit maturity classification. The CNN architecture consists of convolutional layer, pooling layer, ReLU layer, fully connected layer, and loss layer. The core block of CNN is convolutional layer, which consists of filters (or kernels) to detect different types of features from the input image and

later pass forward. Pooling layer is placed in-between successive convolutional layers to reduce the number of parameters and computation in the network, while ReLU layer is the activation function that sets negative values to zero. Fully connected layer (flatten) forms the last few layers in the network by taking the output result from the convolutional or pooling layer to reach a classification decision. Finally, loss layer is applied to show the deviation between predicted (output) and true labels. The CNN image classification methods such as AlexNet [12], VGGNet [13], ResNet [14], and so on are mainly used for extracting features, thus achieving an accuracy sometimes exceeding human-level performance. These image classifiers classify a single object in the image, output a single category per image, and give the probability of matching a class [15]. Recently, Chung and Tai [16] reported 95% accuracy on Fruits 360 dataset produced based on EfficientNet [17], a pretrained CNN. Duong et al. [18] also used EfficientNet and Mixnet to establish fruit recognition system based on Fruits 360. On the same dataset, Muresan and Oltean [19] recorded an average performance of 99.31% accuracy on training set and 93.59% accuracy on test set of ten different CNN configurations. Nevertheless, deep learning exploratory studies on muskmelon stages of maturity are rarely available.

He et al. [14] developed the state-of-the-art ResNet, which introduces skip connection or shortcut connection as shown in Figure 1 to solve drop-off from saturated accuracy for deeper CNN. The 1×1 convolution layer is added to the start and end of the network. With this, the number of connections (parameters) can reduce, while the performance of the network is not so degrading. ResNet consists of convolutional and identity blocks and also uses batch normalization [20]. ResNet is also an excellent image classification network, semantic segmentation, and object detection compared to VGGNet, GoogLeNet [21] etc. Different ResNets include ResNet34, ResNet50 composed of 26 million parameters, ResNet101 with 44 million parameters of 101 layers, and ResNet152 with 152 layers. Meanwhile, ResNet50 and ResNet101 are used widely in object detection models. Wang et al. [22] used ResNet and ResNeXt to detect the internal mechanical damage of blueberries using hyperspectral transmittance data type and achieved F_1 score and average accuracy of 89.52%/89.05% and 88.44%/87.84%, respectively for ResNet/ResNeXt. Mahajan and Chaudhary [23] explored ResNet for image categorical classification and reported 93.57% accuracy for 18-layer CNN which was better than 34-layer and 50-layer CNNs at 91.58% and 92.67%, respectively. However, seldom studies provide two-way data augmentation on muskmelon stages of classification based on ResNet. The first way is a CNN code augmentation, while the second way is the input image augmentation.

In deep learning, an effective data augmentation method is required for more learning data in order to obtain a better and well-generalized model. This data augmentation method is based on possible ways of images captured by a camera. Though data augmentation methods may cause a decrease in the accuracy, they constituted an increase in the number of training data and reduction of overfitting. Mikołajczyk and

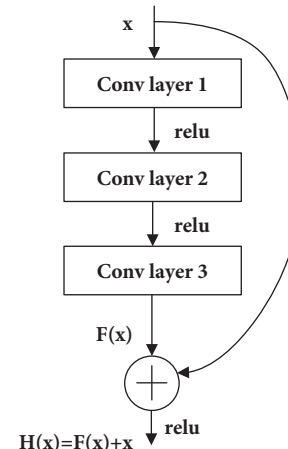


FIGURE 1: Shortcut connection structure in ResNet.

Grochowski [24] proposed data augmentation for improving deep learning in image classification problem that was validated on three medical case studies, while Tsuchiya et al.'s [25] experimental results tested on road damage classification and detection show that data augmentation method can increase accuracy efficiently and effectively. However, research studies involving CNN-ResNet application on two-way data augmentation of muskmelon classification are limited. Therefore, exploring deep learning on the muskmelon stages of maturity classification is of great significance to the automatic muskmelon sorting system.

The main objective of this paper is to create a robust classification model that can better generalize indoor and outdoor muskmelon stages of maturity and that can be applicable to mobile phone, automatic sorting system, and so on. This paper proposed two-way data augmentation methods for the maturity stages of muskmelon classification using indoor and outdoor datasets for the classification model. The color of muskmelon was used as a distinguishing feature of maturity stages. The data augmentation methods were independently applied to each class of the dataset including the CNN code augmentation in order to demonstrate the classification effectiveness through modified ResNet29 architecture application. The obtained experimental results show some interesting phenomena about the muskmelon maturity stage classification.

2. Materials and Methods

2.1. Dataset Preparation. The images of muskmelon were identified with respect to their maturity stages and captured using a digital camera with a resolution of 3968×2976 from Wanghaizhuang village, Houcheng township, Taigu county, Shanxi Province, China. The samples of “xingtian-24” muskmelon in different maturity stages of the same variety in the greenhouse had no defects. The maturity stages of muskmelon fruit that turns from green to yellow were classified as 0: green, 1: white, and 2: yellow. The images collected from the greenhouse termed as outdoor dataset were resized to 416×416 pixels and divided into 75% training set, 20% valid set, and 5% test set according to each

class. After the completion of outdoor image capture, the muskmelon fruits were harvested and sent to the laboratory for indoor image capture. The Shengyue SY8031 autofocus camera and 5500K LED lamp were used for the indoor image acquisition as shown in Figure 2. In order to prevent color deviation while taking images, the camera was adjusted to the white balance of the LED lamp. Meanwhile, a black cloth was spread on the surface of an adjustable lifting table in the photo-studio chamber, and then the muskmelon was placed on the table, where the front and back side images were captured. The reason for capturing the indoor image was to have a true image of muskmelon that is separated from its background for proper classification. Just like the outdoor dataset, the indoor dataset was also divided into 75% training set, 20% valid set, and 5% test set. Table 1 provides details of the outdoor and indoor datasets, and their corresponding sample image is shown in Figure 3. For the completion of Table 1 dataset, the training set for outdoor dataset was added to the indoor dataset in their corresponding classes. The same process was also applied to valid set and test set. This is to enhance the model usage for indoor and outdoor.

This study applied two-way data augmentation methods to increase the generality of the classification model so as to learn more robust features. The first way used CNN code augmentation displayed in Table 2 from Keras—ImageDataGenerator [26]. Meanwhile, the second way augmented randomly the input images from the dataset by thirteen image augmentation methods [27] of three tiers as stated in Table 3 and displayed in Figure 4. The image augmentation method augments 75% training set of both the outdoor and indoor datasets. The image augmentation methods were selected based on possible ways of images from a camera and the various environmental conditions that the classification model could operate. The main idea is to create a robust classification model that can better generalize and that can be applicable in computer-based platform. The dataset for all image augmentations is presented in Table 4 including the combined dataset from augmented indoor and outdoor datasets. The addition of A-indoor and A-outdoor datasets was carried out in the same way as shown in Table 1. Finally, the out-indoor dataset (Table 1) was added to their corresponding one in A-out-indoor dataset (Table 4) in order to create All dataset shown in Table 5.

2.2. ResNet Architecture. The idea of ResNet was adopted based on its skip connections between layers that add the outputs from previous layers to the outputs of stacked layers. According to [13], training of much deeper networks constitute an increase in classification performance. 29-layer ResNet was used in this paper to solve the degradation problem of networks, accelerate the training speed, and promote the faster network convergence. Figure 5 shows the modified 29-layer ResNet in detail. ResNet 29-layer was chosen based on the input image resolution 128×128 pixels feed into the network and its convergence of loss and accuracy toward achieving an excellent result [28]. Notably,

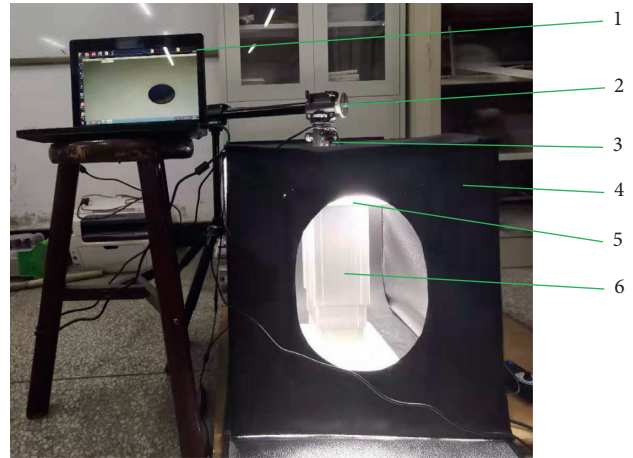


FIGURE 2: Muskmelon image acquisition setup for indoor dataset. 1: computer; 2: tripod; 3: camera; 4: photo studio; 5: LED light bar; 6: lifting platform.

TABLE 1: Dataset details without data augmentations.

Dataset	Class	Training set	Valid set	Test set	Total
Outdoor	Green	310	70	20	400
	White	658	173	44	875
	Yellow	907	257	61	1225
Indoor	Green	216	69	15	300
	White	841	233	56	1130
	Yellow	1261	316	83	1660
Out-Indoor	Green	507	158	35	700
	White	1491	414	100	2005
	Yellow	2195	546	144	2885

the last fully connected layer played a role as a classifier, which calculated and output the scores of muskmelons.

2.3. Model Training and Evaluation. The classification model was trained and tested on Windows 10 with the following specification: Intel® Core™ i7-8700 CPU @ 64 bit 3.20 GHz, 16 GB RAM, NVIDIA Quadro M4000 GPU, CUDA v10.2, cuDNN v7.6.5. The ResNet received an input image size of $128 \times 128 \times 3$. The stochastic gradient descent (SDG) optimizer was employed in this study because it requires less memory with computation taking only one point at once. The learning rate of SDG optimizer was set to a constant of 1×10^{-2} , momentum of 0.9 and categorical cross-entropy was used as a cost function. In addition, the code augmentation methods mentioned in Table 2 were included in the model network. The models were trained for 50 epochs according to the constructed datasets and structured as shown in Figure 6. Figure 6 shows the 14 datasets trained in this paper. With code means that the network used to train the dataset contains Table 2 information, while the dataset that contains A prefix (A-) means with image augmentation (Table 3). The performance of classifiers was evaluated using equation (1). Precision is simply the ratio of the number of correctly predicted muskmelons to the total number of predicted muskmelons, Recall is the ratio of the number of correctly predicted muskmelons to the total number of

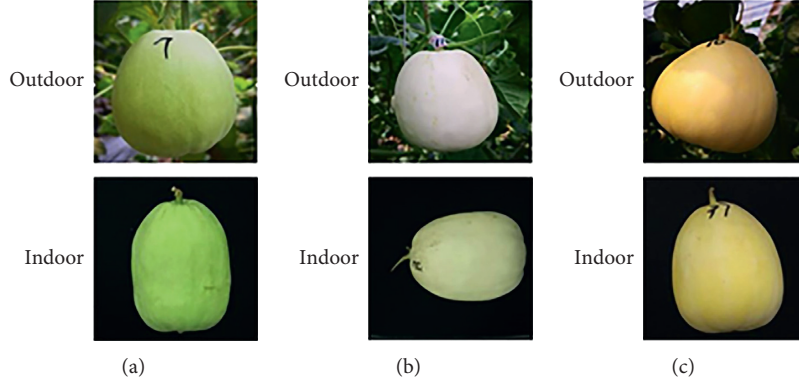


FIGURE 3: Sample of image in outdoor and indoor datasets. (a) Green. (b) White. (c) Yellow.

TABLE 2: Details of code augmentation method.

No.	Details of method ¹
1	Rotation range 20
2	Zoom range 0.15
3	Width shift range 0.2
4	Height shift 0.2
5	Shear range 0.15
6	Horizontal flip
7	Vertical flip

¹Source: ImageDataGenerator [26].

muskmelons in the dataset, and F_1 score is the trade-off between recall and precision to show the comprehensive performance of the trained models.

$$\left\{ \begin{array}{l}
 \text{Precision} = \frac{TP}{TP + FP}, \\
 \text{Recall} = \frac{TP}{TP + FN}, \\
 \text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \\
 F_1 \text{ score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
 \end{array} \right. \quad (1)$$

where TP is true positive, FP is false positive, TN is true negative, and FN is false negative.

3. Results and Discussion

3.1. Evaluation Metrics. The obtained loss rate curve of the valid set was similar to the training set, with lower error at 50 epochs, indicating the high stability of outdoor and indoor muskmelon. Outdoor of Figure 7(a) and indoor of Figure 7(c), that is, without code augmentation, showed an excellent performance of almost 100% valid accuracy with minimum valid loss. The confusion matrix of classification results also showed an excellent performance of 100% prediction in both Figures 7(a) and 7(c), where 0 is green, 1 is

TABLE 3: Details of image augmentation method.

No.	Details of method ¹
1	imgcorruptlike.spatter (severity = 3)
2	imgcorruptlike.GaussianNoise (severity = 2)
3	imgcorruptlike.GaussianBlur (severity = 2)
4	imgcorruptlike.motionBlur (severity = 3)
5	imgcorruptlike.contrast (severity = 2)
6	imgcorruptlike.brightness (severity = 2)
7	imgcorruptlike.saturnate (severity = 2)
8	HorizontalFlip (0.5)
9	Affine (rotate = (-45, 45))
10	ShearY ((-30, 20))
11	Rain (speed = (0.1, 0.3))
12	Clouds()
13	Rain (drop_size = (0.15, 0.20))

¹Source: imgaug [27].

white, and 2 is yellow. However, the performance of Figure 7(c) is more stable than that of Figure 7(a) due to their image feature background being different. The incorporation of code augmentation into the network constituted the low stability found in Figures 7(b) and 7(d) with shown missed prediction in their corresponding confusion matrix. Nevertheless, Figure 7(d) is more stable than Figure 7(b). This is as a result of difference in their background images. The obtained findings of Figure 8 trended similarly as Figure 7, except that Figure 8 contained image augmentation. The number of missed predictions in the confusion matrix for indoor is more than outdoor, where Figure 8(c)>Figure 8(a) for without code augmentation and Figure 8(d)>Figure 8(b) for with code augmentation. This is an indication of outdoor images adjusting to the external features or background images compared to indoor images. In other words, two-way augmentations of outdoor performed better than indoor.

Similarly, the obtained results of combined outdoor and indoor dataset without image augmentation showed that it can better perform in terms of confusion matrix than combined outdoor and indoor datasets with image augmentation. Apart from Figure 9(a) having without code and image augmentation that showed 100% prediction, the number of missed predictions in Figure 9(b)<Figure 9(c)<Figure 9(d). This further confirmed the two-way augmentations constituting performance degradation. With reference to the

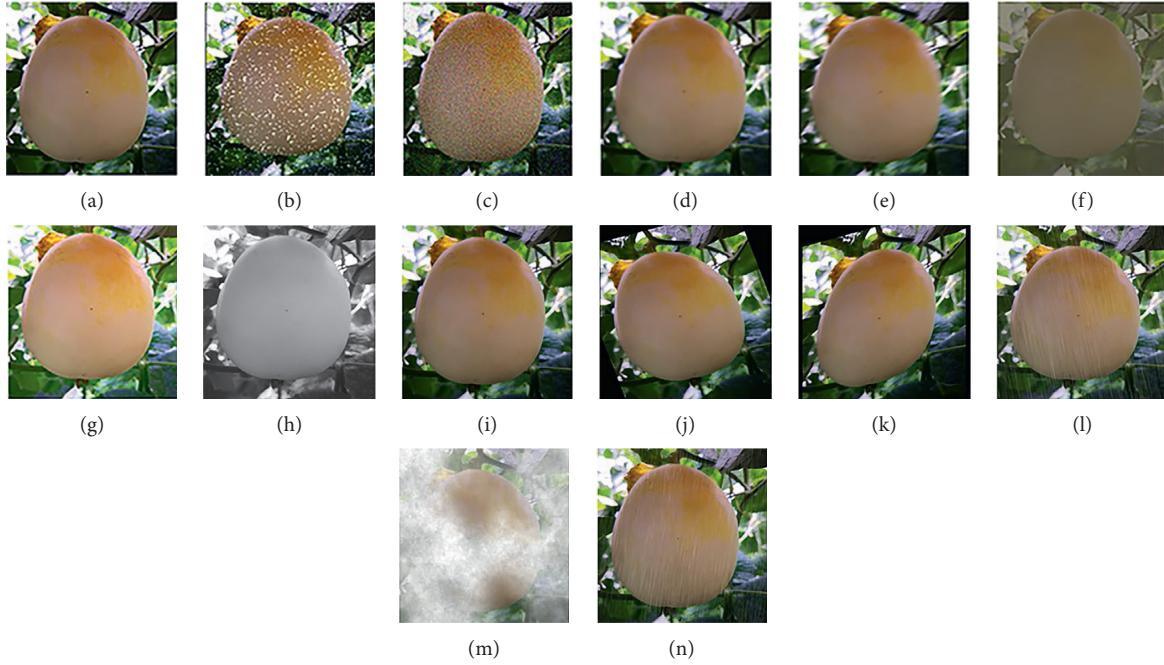


FIGURE 4: Sample of image augmentation in outdoor and indoor datasets. (a) Original. (b) Spatter. (c) Noise. (d) Blur. (e) Motion. (f) Contrast. (g) Brightness. (h) Saturate. (i) Flip. (j) Affine. (k) Shear. (l) Rain. (m) Clouds. (n) Rain drops.

TABLE 4: Dataset details for data augmentations.

Dataset	Class	Training set	Valid set	Test set	Total
A-Outdoor	Green	1560	416	104	2080
	White	3375	947	228	4550
	Yellow	4814	1237	319	6370
A-Indoor	Green	1182	300	78	1560
	White	4418	1164	294	5876
	Yellow	6450	1750	432	8632
A-Out-Indoor	Green	2765	693	182	3640
	White	7790	2115	521	10426
	Yellow	11246	3006	750	15002

TABLE 5: All dataset with data augmentations.

Dataset	Class	Training set	Valid set	Test set	Total
All	Green	3248	875	217	4340
	White	9326	2484	621	12431
	Yellow	13422	3571	894	17887

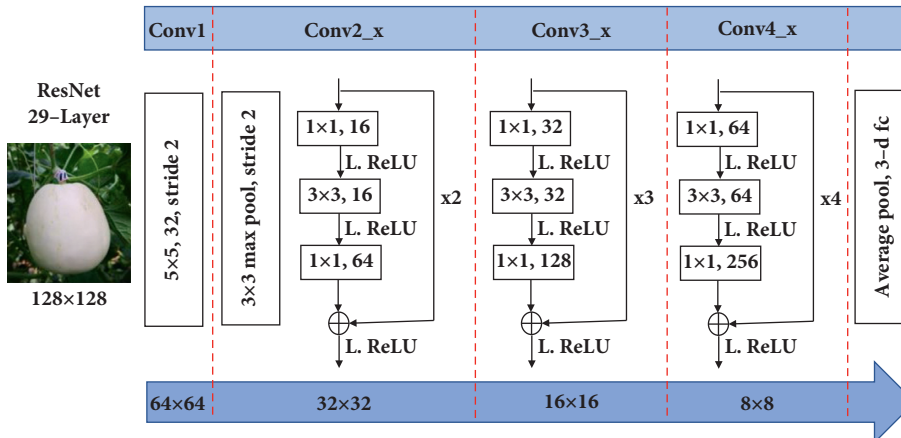


FIGURE 5: Modified 29-layer ResNet.

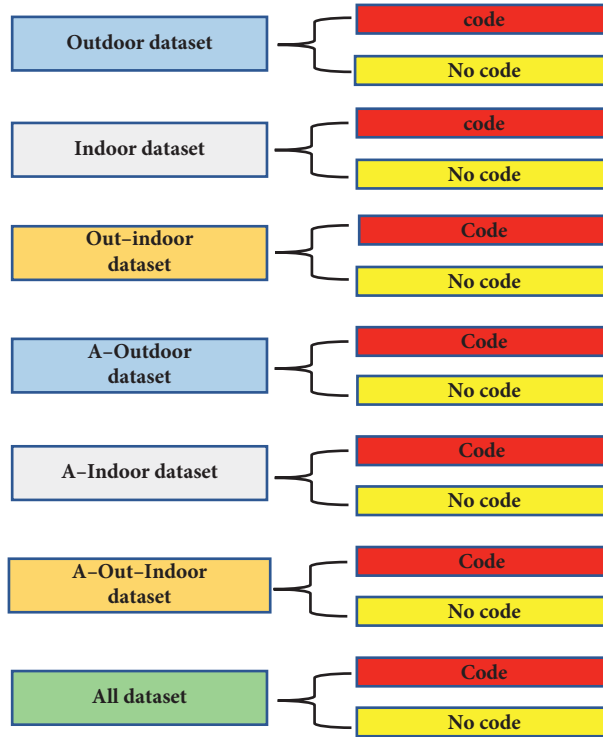


FIGURE 6: Model training structure.

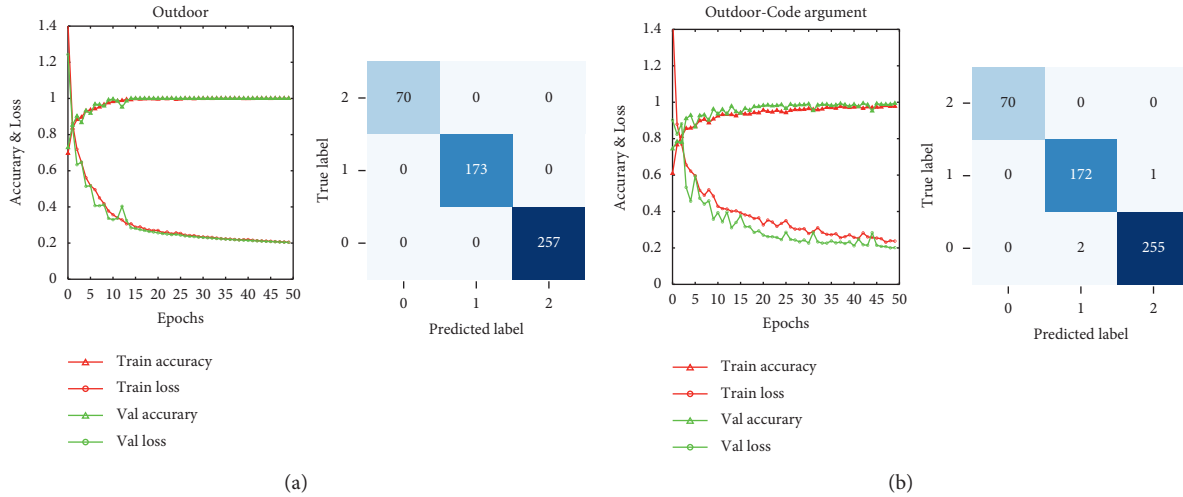


FIGURE 7: Continued.

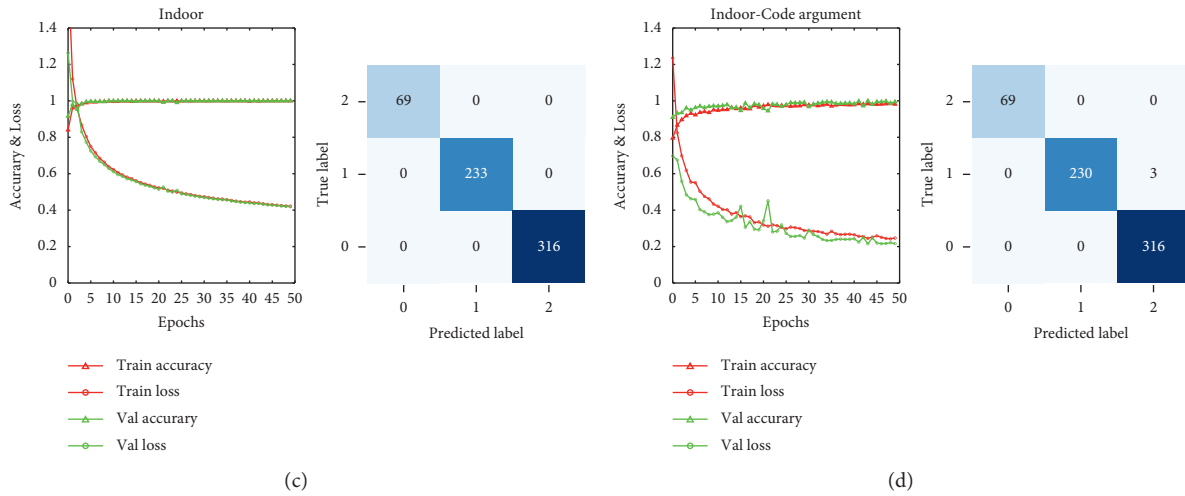


FIGURE 7: No image argumentation results for (a) outdoor without code, (b) outdoor with code, (c) indoor without code, and (d) indoor with code.

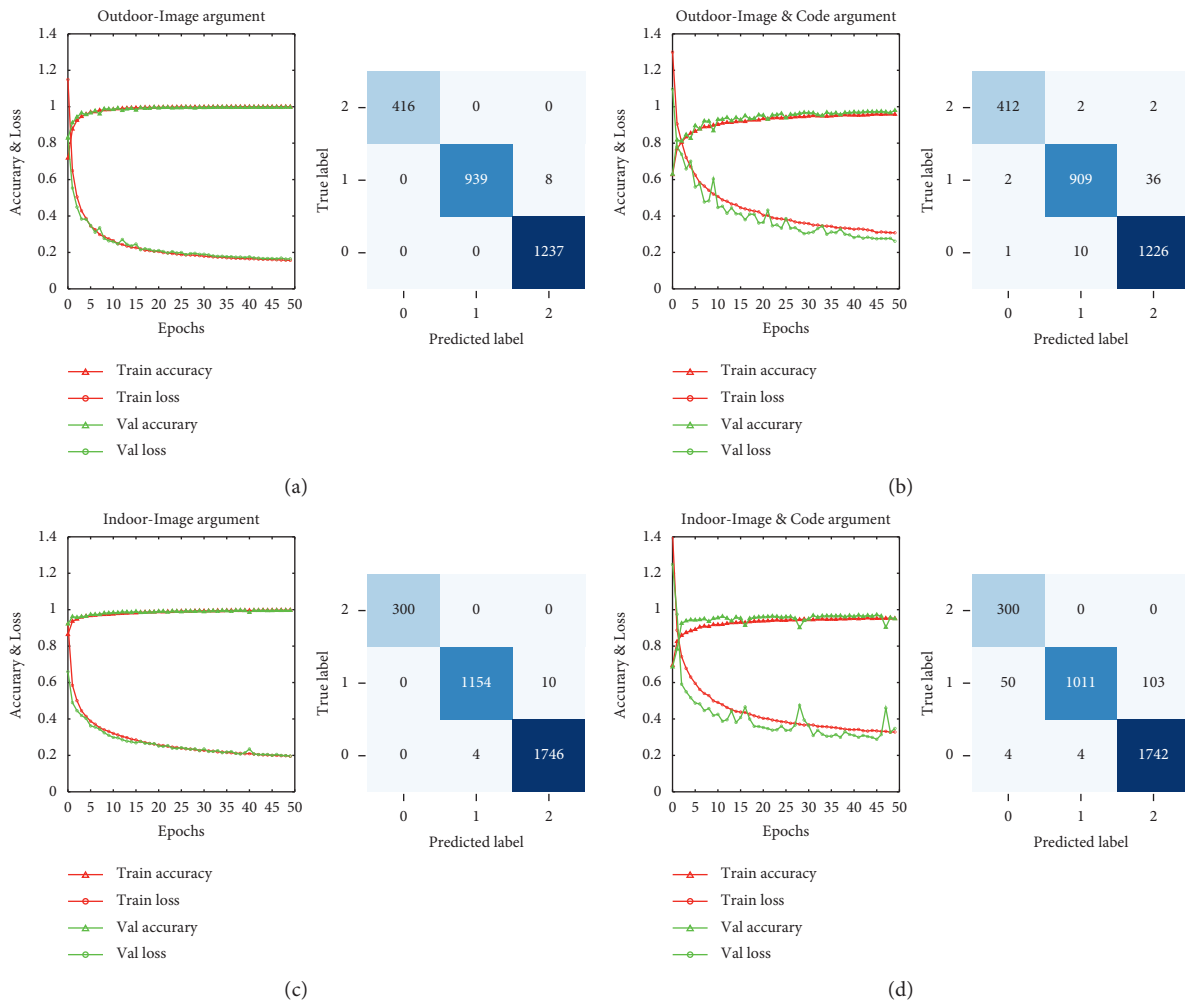


FIGURE 8: Image argumentation results for (a) outdoor without code, (b) outdoor with code, (c) indoor without code, and (d) indoor with code.

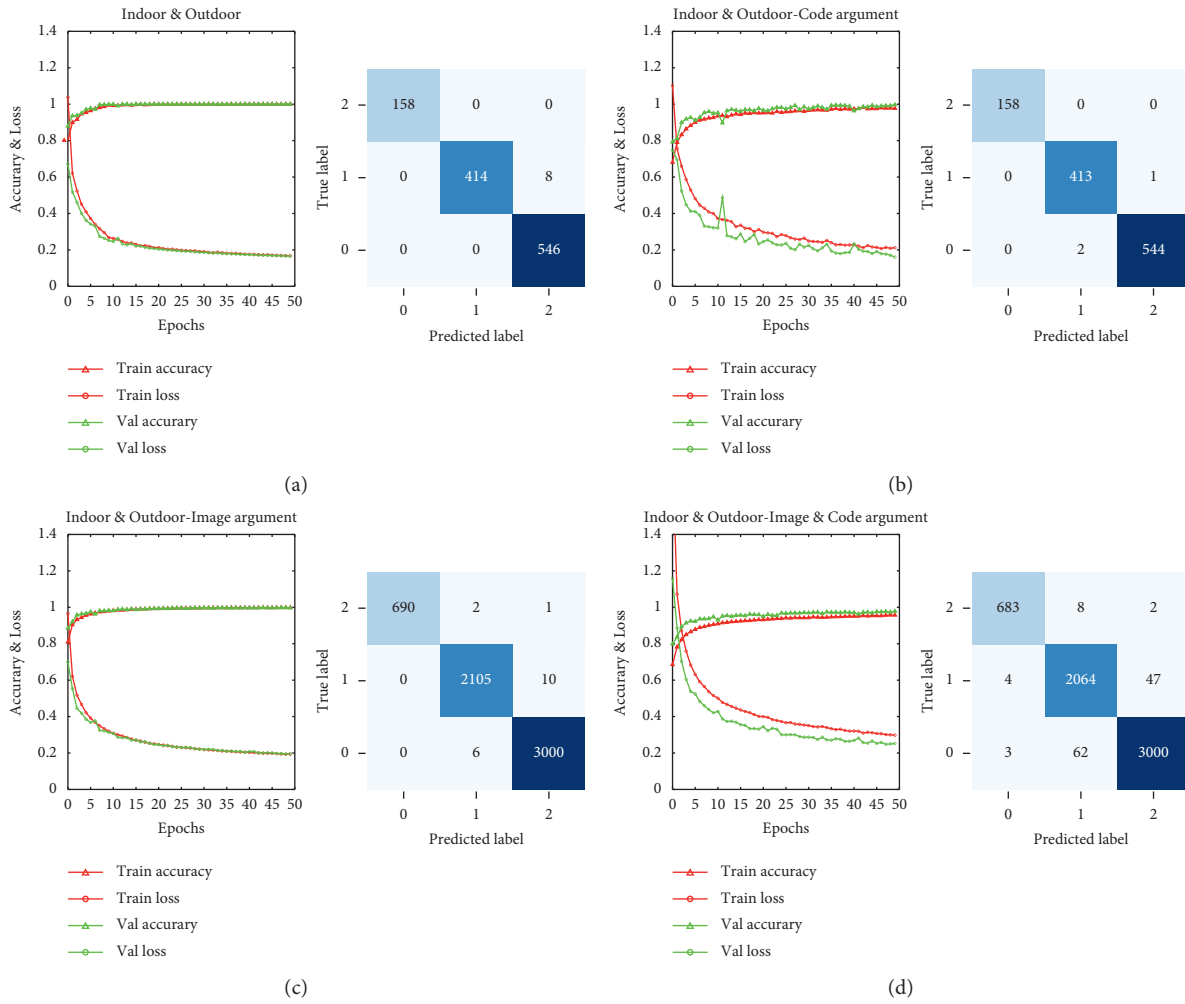


FIGURE 9: Combined outdoor and indoor results for (a) out-indoor without code, (b) out-indoor with code, (c) out-indoor image augmentation without code, and (d) out-indoor image augmentation with code.

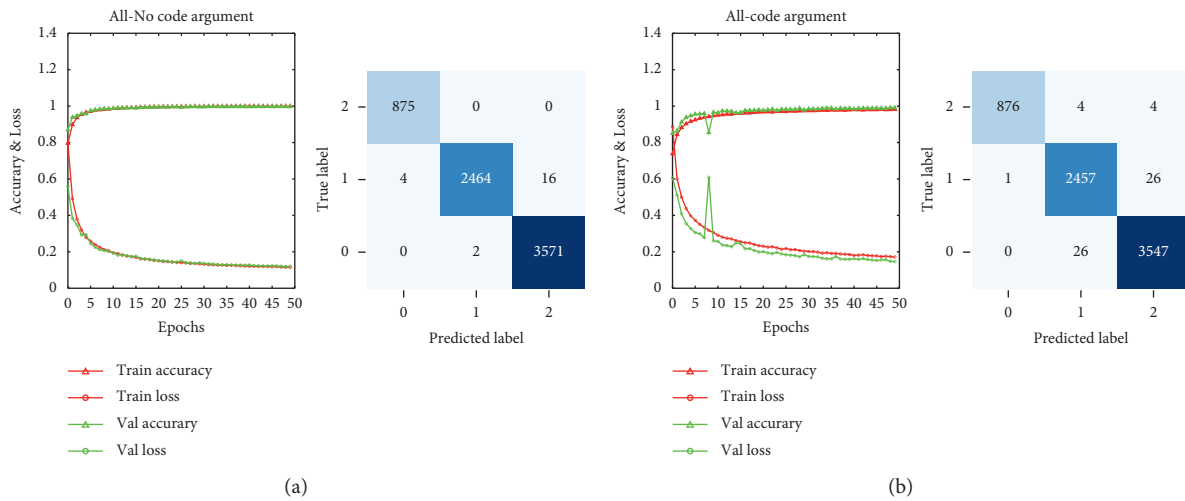


FIGURE 10: All combined results for (a) image augmentations without code and (b) image augmentations with code.

TABLE 6: Average performance evaluation results.

Dataset	Class	Precision (%)	Recall (%)	F_1 score (%)
Outdoor	Code	99	99	99
	No code	100	100	100
Indoor	Code	100	100	100
	No code	100	100	100
Out-Indoor	Code	100	100	100
	No code	100	100	100
A-Outdoor	Code	98	98	98
	No code	100	100	100
A-Indoor	Code	95	95	95
	No code	100	100	100
A-Out-Indoor	Code	98	98	98
	No code	100	100	100
All	Code	99	99	99
	No code	100	100	100



FIGURE 11: Visualization results of the tested model created from All dataset with code augmentation.

obtained confusion matrix results, the code augmentation caused more performance degradation than the image augmentation. Figure 10 for all combined datasets proved the effect of code augmentation, where the number of missed predictions observed in Figure 10(b) is more than that of Figure 10(a). In addition, the code augmentation is responsible for more instability compared to image augmentation. Nevertheless, the findings of All dataset can generalize very well to muskmelon stages of maturity classification.

The average performance results for precision, recall, and F_1 score are shown in Table 6. The F_1 score of indoor and out-indoor datasets with and without code augmentation was noted to be 100% compared to other datasets. This explains that the extracted image feature without background found in indoor, while outdoor and indoor combination are complimenting each for the obtained results in out-indoor dataset. The without code augmentation of F_1 score for all the dataset is 100%, while with code augmentation of F_1 score varied from dataset to dataset.

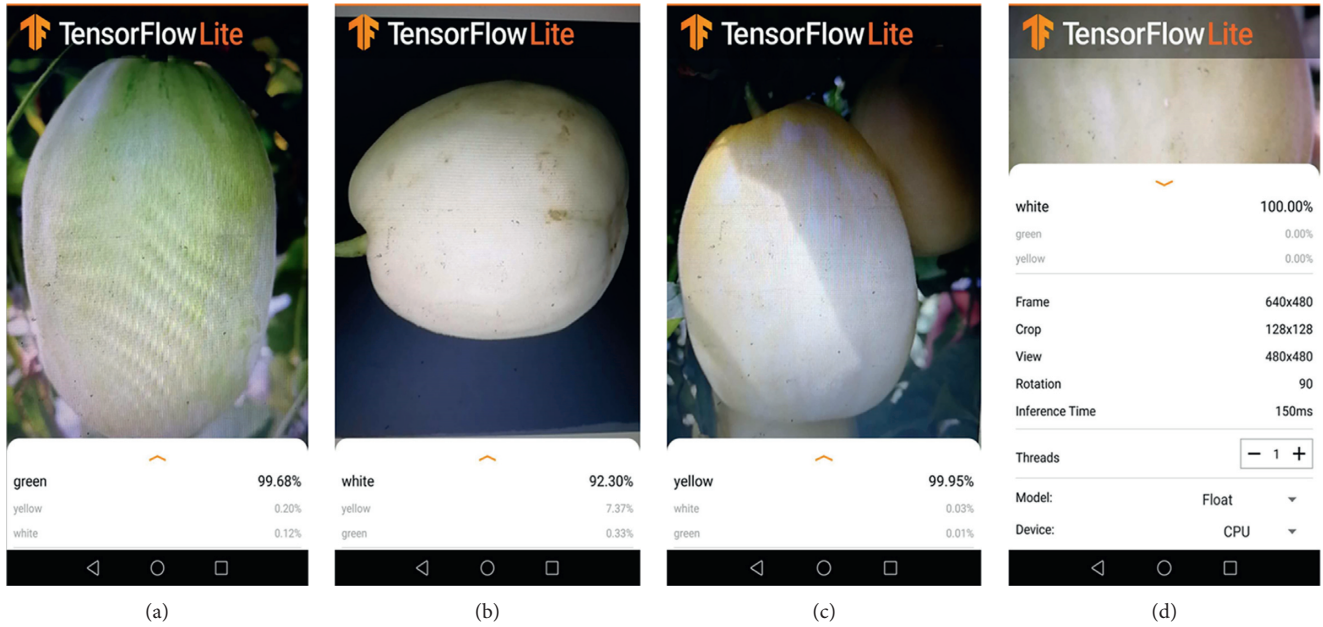


FIGURE 12: Muskmelon stages of maturity classification based on converted model through mobile application. (a) Green class. (b) White class. (c) Yellow class. (d) Inference time.

TABLE 7: Comparison of classification results of different models.

Model	Weight size (MB)	Precision (%)	Recall (%)	F_1 score (%)
ResNet29	3.5	99	99	99
VGGNet16	56.3	98	98	98
ResNet18	45.2	95	95	95
ResNet50	90.2	98	98	98
MobileNet-v2	2.1	97	97	97
ResNeXt50	90	99	99	99
DenseNet121	28.5	99	99	99
InceptionV4	161	99	99	99

The least F_1 score of 95% was noticed in A-indoor with code augmentation, similar to Figure 8(d) results. Just like Figure 10, the All dataset displays an excellent average performance.

3.2. Computer-Based Application. The created models from all the datasets were tested on their corresponding test set for visualization purpose. Figure 11 shows the visualization findings of model created from All dataset with code augmentation. The figure justified the performance with good predictions. Meanwhile, the created models were converted to model.tflite and model_unquant.tflite using TensorFlow and Keras package for mobile application. Figure 12 is the outcome of converted model created from All dataset with code augmentation. Figure 12(a) shows green muskmelon, Figure 12(b) shows white muskmelon, Figure 12(c) shows yellow muskmelon, and Figure 12(d) shows the inference time of classification from the tested converted model. It is an indication that our models are applicable to any computer-based applications. Apart from the mobile application, the computer-based application can be interfaced with electrical, mechanical system, and so on in order to perform other task such as the automatic sorting system.

3.3. Comparison with Other Models. Table 7 shows the results of tested models created from All dataset in Table 6. The compared results shows that our ResNet29 model outperformed other models. The F_1 score of ResNeXt50, Densenet121, and InceptionV4 is similar to ResNet29 at 99%, but with a larger model size. If the model size is large, it will consume more computing resources, and the speed of model is slow. Meanwhile, the size of MobileNet-v2 model is smaller than that of ResNet29 model, but its test result is not as good as our ResNet29. Therefore, our modified ResNet29 model has more advantages than other models for muskmelon maturity stage classification.

4. Conclusions

The color variation of muskmelon during maturity stages makes it more difficult for visually impaired people, supermarket or grocery workers, and even farm workers to classify. This paper proposed two-way data augmentation methods for the maturity stages of muskmelon classification using indoor and outdoor datasets to create a robust classification model that can better generalize. A modified ResNet of 29 layers was adopted to create the classification

model. The results showed that code augmentation (first way) caused more performance degradation than the image augmentation (second way). This is to say that code augmentation is more reliable in building a robust model than image augmentation which calls for future investigation of classification method. However, the two-way data augmentations including the combination of outdoor and indoor datasets showed an excellent performance of F_1 score ~99%, which is applicable for computer-based platform for quick muskmelon stages of maturity classification, such as mobile phone application and so on.

In addition, although the model proposed in this paper can achieve better classification effect of muskmelon maturity, it needs to cooperate with the object detection algorithm to be better used in automatic sorting/picking equipment. In this paper, the muskmelon object detection algorithm is not mentioned, which needs to be studied in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was sponsored by the University Science and Technology Innovation Project of Shanxi, China (project no. 2019L0402), Doctoral Research Project of Shanxi Agricultural University (project no. 2018YJ43), and Excellent Doctor of Shanxi Province to Jin Work Award Fund Research Projects (project no. SXYBK2018030).

References

- [1] F. García-Lamont, J. Cervantes, S. Ruiz, and A. López-Chau, "Color characterization comparison for machine vision-based fruit recognition," *Intelligent Computing Theories and Methodologies*, vol. 92, pp. 258–270, 2015.
- [2] E. O. Olaniyi, O. K. Oyedotun, and A. Khashman, "Intelligent grading system for banana fruit using neural network arbitration," *Journal of Food Process Engineering*, vol. 40, no. 1, p. 9, 2017.
- [3] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, "Deep learning - method overview and review of use for fruit detection and yield estimation," *Computers and Electronics in Agriculture*, vol. 162, pp. 219–234, 2019.
- [4] S. Anaty, V. C. Mawardi, J. Hendryli et al., "Fruit maturity classification using convolutional neural networks method," *IOP Conference Series: Materials Science*, vol. 1007, no. 1, Article ID 012149, 2020.
- [5] N. Saranya, K. Srinivasan, and S. Kumar, "Banana ripeness stage identification: a deep learning approach," *Journal of Ambient Intelligence Humanized Computing*, vol. 1, 2021.
- [6] S. Tu, Y. Xue, Y. Qi, H. Wan, and L. Mao, "Detection of passion fruits and maturity classification using red-green-blue depth images," *Biosystems Engineering*, vol. 175, pp. 156–167, 2018.
- [7] J. Liu, J. Pi, and L. R. Xia, "A novel and high precision tomato maturity recognition algorithm based on multi-level deep residual network," *Multimedia Tools and Applications*, vol. 79, no. 13–14, pp. 9403–9417, 2020.
- [8] X. Yueju, H. Ning, T. Shuqin et al., "Immature mango detection based on improved YOLOv2," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 34, no. 07, pp. 173–179, 2018.
- [9] S. Parvathi and S. Tamil Selvi, "Detection of maturity stages of coconuts in complex background using Faster R-CNN model," *Biosystems Engineering*, vol. 202, no. 6, pp. 119–132, 2021.
- [10] H. J. Liu, C. Y. Wei, W. Han et al., "Determination of Huanghua Pear's harvest time based on convolutional neural networks by visible-near infrared spectroscopy," *Spectroscopy and Spectral Analysis*, vol. 40, no. 9, pp. 2932–2936, 2020.
- [11] B. Chen and J. Yan, "Fresh Tea Shoot maturity estimation via multispectral imaging and deep label distribution learning," *IEICE - Transactions on Info and Systems*, vol. 103, no. 9, pp. 2019–2022, 2020.
- [12] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. 2, 2012.
- [13] K. Simonyan and A. Zisserman, "A very deep convolutional networks for large-scale image recognition," 2014, <https://arxiv.org/abs/1409.1556>.
- [14] K. He, X. Zhang, S. Ren et al., "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.
- [15] A. B. Amjoud and M. Amrouch, "Convolutional neural networks backbones for object detection, springer, image and signal processing. ICISP 2020," *Lecture Notes in Computer Science*, vol. 2020, Article ID 4152816, 22 pages, 2020.
- [16] T. P. C. Dang and D. V. Tai, "A fruits recognition system based on a modern deep learning technique," *Journal of Physics Conference*, vol. 1327, Article ID 012050, 2019.
- [17] M. Tan and Q. V. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," 2019, <https://arxiv.org/abs/1905.11946>.
- [18] L. T. Duong, P. T. Nguyen, C. Di Sipio et al., "Automated fruit recognition using EfficientNet and MixNet," *Computers and Electronics in Agriculture*, vol. 171, p. 10, 2020.
- [19] H. Murean and M. Oltean, "Fruit recognition from images using deep learning," *Acta Universitatis Sapientiae, Informatica*, vol. 10, no. 1, pp. 26–42, 2017.
- [20] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," 2015, <https://arxiv.org/abs/1502.03167>.
- [21] C. Szegedy, V. Vanhoucke, S. Ioffe et al., "Rethinking the inception architecture for computer vision," 2015, <https://arxiv.org/abs/1512.00567>.
- [22] Z. Wang, M. Hu, and G. Zhai, "Application of deep learning architectures for accurate and rapid detection of internal mechanical damage of blueberry using hyperspectral transmittance data," *Sensors*, vol. 18, no. 4, p. 14, 2018.
- [23] A. Mahajan and S. Chaudhary, "Categorical image classification based on representational deep network (RESNET)," in *Proceedings of the 2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, June 2019.
- [24] A. Mikolajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem,"

- International Interdisciplinary Phd Workshop*, vol. 99, pp. 117–122, 2018.
- [25] H. Tsuchiya, S. Fukui, Y. Iwahori, Y. Hayashi, W. Achariyaviriya, and B. Kijirikul, “A method of data augmentation for classifying road damage considering influence on classification accuracy,” *Procedia Computer Science*, vol. 159, pp. 1449–1458, 2019.
- [26] 2020 Keras-cn, available at: <https://keras-cn.readthedocs.io/en/latest/preprocessing/image/>.
- [27] A. B. Jung, *Imgaug*, available at: <https://github.com/aleju/imgaug>, 2020.
- [28] H. AY and N. AY, “Rectifier nonlinearities improve neural network acoustic models,” in *Proceedings of the The International Conference on Machine Learning*, Atlanta, Georgia, USA, May 2013.