An Improved Deep Learning Model of Chili Disease Recognition with Small Dataset

Nuramin Fitri Aminuddin, Zarina Tukiran, Ariffuddin Joret, Razali Tomari, Marlia Morsin
Department of Electrical and Electronic Engineering
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, Johor, Malaysia

***Abstract*—Due to its tasty and spicy fruit with nutritional qualities, chili is a demanding crop widely farmed around the world. Hence, it is essential to accurately determine the health status of chili for agricultural productivity. Recent years have seen impressive results in recognition fields due to deep learning approaches. However, deep learning models' networks need an abundant data to perform well and collecting enormous data for the networks is time-consuming and resource-intensive. A data augmentation method is proposed to overcome this problem. It was applied to a small dataset of healthy and diseased chili leaf by utilizing geometric transformation method. Eventually, two deep learning models of CNN and ResNet-18 were evaluated using augmented and original datasets. From a series of experiment, it can be concluded that the trained deep learning models using original and augmented datasets perform better with an average accuracy performance of 97%.**

Keywords—Chili leaf; deep learning; data augmentation; geometric transformation

I. Introduction

Chili (Capsicum sp.) is an important spice from the family Solanaceae that originates from South and Central America [1]. It is a demanding crop and extensively cultivated in tropical Asia and equatorial America with a high genetic diversity due to its edible and pungent fruit with nutritional values [2]. Vitamin C, potassium, phosphorus fibre, antioxidants like vitamin A, and flavonoids like β-carotene, α-carotene, lutein, zeaxanthin, and cryptoxanthin are among the nutritional values contained in a chili fruit which have the ability to suppress several human cancers [3]. However, owing to the impact of fungus, bacteria, viruses, pests, and climate on the chili cultivation process, the chili itself is susceptible to a variety of diseases. These diseases make it difficult for chili to thrive, reducing the production and quality of the fruit. It is estimated that 60-70 percent of diseases and early disease symptoms are detected just on leaves [4]. Hence, it is necessary to identify chili diseases precisely and implement early preventive and treatment measures.

Since the advent of deep learning, deep learning models have made significant advances in disease recognition [5]. Good performance from deep learning models normally needs a large number of parameters and enormous data to make these parameters operate properly. In order to do so, manual data collection and labelling [6] are required to get enormous data, which is resource-intensive and time-consuming. As a result, it can be hard to gather enough data to train the deep learning models, which significantly limits the accuracy of chili disease recognition.

With small collection of datasets, several research [7-10] in the chili agricultural field has used the data augmentation method to increase the volume of datasets. The method generates data artificially via adding augmented images to the existing dataset through either oversampling or warping [11]. By using oversampling augmentation such as generative adversarial networks (GANs), augmented images with a low likelihood of occurring label (abnormal) are added to the original datasets, preventing a deep learning model from being biassed toward the majority label of images during the recognition process. Even though GANs have intriguing promise, they need a substantial number of initial original images in order to train and create an augmented image [12]. As a result, depending on the initial size of the original dataset, GANs may not be a viable option.

In contrast, warping augmentations such as geometric transformation alter original images in such a way that their labels are maintained [13] , and this is accomplished without requiring a minimum amount of the original image to be present. Most of research [10, 13-15] that employed geometric transformation to augment original images concentrated on single transformation operation such as rotation, flipping, and scaling. To the best of the author knowledge, there has been very limited research on numerous fusions of geometric transformation operations in order to produce augmented images throughout the years. Hence, this research examines the data augmentation method known as a geometric transformation and its several transformation fusions on a small chili dataset. The augmented and original images are then fed into two deep learning models, Convolutional Neural Network (CNN) and Residual Network (ResNet-18), developed from scratch for chili disease recognition.

The contributions of this research findings can be summarized as follows. According to the findings of this research, the optimal level of accuracy for recognising chilli diseases depends on the category of datasets used and the size of the deep learning model. The finding implies that the best optimal recognition accuracy came from small datasets with both original and augmented images that were fed to a larger-sized model. This is in contrast to datasets with only original data (original datasets) and datasets with only augmented data (augmented datasets). All of the research experiments reveal that the deep learning models created from scratch are accurate to a maximum reported accuracy of 99.7%.

The remainder of this paper is organised as follows. The original dataset of healthy and diseased chili leaf produced for this research is explained in Section II. Meanwhile, Section III delves into the data augmentation method's transformation process, focusing on geometric transformation and its several transformation fusions. The architectures of deep learning models for feature extraction and recognition purposes are then discussed in Section IV. In Section V, the experimental procedures and testings used in this research to acquire the accuracy performance findings are described and the conclusion of this research is presented in Section VI.

II. Chili Leaf Image Dataset

In this research, the camera of an Oppo Reno 2 smartphone was used to capture images of chili leaf in the Batu Pahat state of Johor, Malaysia. Both types of leaf showed healthy and indications of bacterial spot disease. Only 1200 original chili leaf images were able to be acquired due to the low quantity of chilli crops in the research site. Of those 1200 images, 600 showed healthy chili leaf, while the remaining 600 showed diseased chili leaf. The images are captured in auto-focus mode at a resolution of 3000 x 4000 pixels before being resized down to 224 x 224 pixels.

III. Geometric Transformation

Data augmentation can be described as the mapping of any method that artificially increases the original dataset using the preservation label of transformations [13]:

$$\varphi = Y \to Z \tag{1}$$

where Y is the original dataset and Z is the augmented dataset of Y. The original dataset that has been artificially increased is therefore expressed as:

$$Y' = Y \cup Z \tag{2}$$

where Y' stores the original dataset as well as the transformations described by φ. It is worth noting that the preservation label of transformations reflects that if an image d is an element of class f, then $\varphi(\mathrm{d})$ is likewise an element of class f. Given that there is an infinite number of mappings $\varphi(\mathrm{d})$ that fulfil the criterion of preservation label of transformations, this research assesses an augmentation method, namely the geometric transformation.

Geometric transformation is a data augmentation method that alters the image's geometry by relocating the locations of each pixel's value [16]. The image's fundamental pattern of a class is preserved, but it has been shifted to a new place and alignment. This research explores the types of geometric transformation such as reflection, translation, rotation, shearing, scaling, and several fusions between them.

Reflection [17] mirrored an image around the horizontal (x-axis) or vertical (y-axis). It assists users in increasing the amount of images of an original dataset by requiring the original image matrices' rows to be inverted. In a horizontal reflection, the left and right sides of the image are turned horizontally. As shown below, the $\mathrm{f_x}$ and $\mathrm{f_y}$ components indicate the pixel's present location after reflection across the x-axis, while the coordinates of the object's original position in the image are denoted by x and y:

$$A = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} -1\ 0 \\ 0\ 1 \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix} \tag{3}$$

where A is the process equation of reflection on $the\ x$-axis. In a vertical flipping, the image is turned upside down such that the y-axis is on top and the x-axis is on the bottom. The f_x and f_y components indicate the pixel's present location after reflection across the y-axis, while the coordinates of the object's original position in the image are denoted by x and y, where B is the process equation of reflection on the y-axis:

$$B = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} 1\ 0 \\ 0\ -1 \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix} \tag{4}$$

Then there is translation [17], which is the process of shifting an object in an image from one location to another. The translation can be performed in four directions: down, up, right and left, which helps prevent positional bias in a set of translated images. The $\mathrm{f_x}$ and $\mathrm{f_y}$ components indicate the pixel's present location after translation, while the coordinates of the object's original position in the image are denoted by x and y, where C is the process equation of translation:

$$C = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \tag{5}$$

Next, rotation [18] entails spinning the original image, either in the left or right direction, with angles ranging from 1^{o} to 359^{o}. The f_x and f_y components indicate the pixel's present location after rotation while the coordinates of the object's original position in the image are denoted by x and y, where D is the process equation of rotation:

$$D = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} cos\varphi\ -sin\varphi \\ sin\varphi\ cos\varphi \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix} \tag{6}$$

Additionally, shearing [17] is the process of altering the shape of the original image in a single direction. Shearing can be done in either the x-axis or the y-axis direction. The $\mathrm{f_x}$ and $\mathrm{f_y}$ components indicate the pixel's present location after shearing while the coordinates of the object's original position in the image are denoted by x and y.

Consequently, (7) shows the shearing in the x-axis direction, whereas (8) shows the shearing in the y-axis direction. The E and F are the process equations of images sheared on the x-axis and the y-axis directions, respectively.

$$E = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} 1\ shX \\ 0\ 1 \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix} \tag{7}$$

$$F = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} 1\ 0 \\ shY\ 1 \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix} \tag{8}$$

In contrast, scaling [18], often known as zooming or cropping, is the process of enlarging and shrinking the original image in order to view more information. The operation of the process is to enlarge or shrink the image from a starting X, Y position to a destination X, Y. The $\mathrm{f_x}$ and $\mathrm{f_y}$ components indicate the pixel's present location after scaling, while the coordinates of the object's original position in the image are denoted by x and y,where G is the process equation of scaling.

$$G = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} Xscale\ 0 \\ 0\ Yscale \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix} \tag{9}$$

An object in an original image that has been reflected, whether on the x-axis or the y-axis, can be translated by shifting the reflected image into a new location, resulting in a fusion of reflection and translation transformations. Given that H and I are the process equations of images reflected on the x-axis and the y-axis, respectively, and then translated, the equations are as follows:

$$H = A \cup C \tag{10}$$

$$\mathrm{I} = B \cup C \tag{11}$$

This research also includes the fusion of reflection and scaling transformations. Given that J and K are the process equations of images scaled, and then reflected on the x-axis and the y-axis, respectively, the equations are as follows:

$$J = G \cup A \tag{12}$$

$$K = G \cup B \tag{13}$$

Additionally, this research also has a fusion of scaling and shearing transformations. Given that L and M are the process equations of images scaled, and then sheared on the direction of the x-axis and the y-axis, respectively, the equations are as follow:

$$L = G \cup E \tag{14}$$

$$M = G \cup F \tag{15}$$

Finally, there can be more than two fusions of geometric transformations. Given that N and O are the process equations of images scaled, then reflected on the x-axis and the y-axis, respectively, and lastly followed by translation, the equations are as follows:

$$N = G \cup A \cup C \tag{16}$$

$$O = G \cup B \cup C \tag{17}$$

IV. Developments of Deep Learning Model

In the leaf disease recognition domain, researchers have employed enhanced deep learning network architecture through various models [9-10] and applied them to chili disease recognition. This research employs two types of deep learning models: CNN and ResNet-18, which are developed from scratch using the Deep Network Designer [19]. An accuracy measure in [20] is used as a metric to evaluate the performance of these models. The architecture of each model is described in further detail in the following section.

A. *CNN Architecture*

Output, input, and hidden layers are the three primary layers of a CNN model [21]. It is common for the hidden layers to have convolutional with rectified linear unit (ReLU) function, pooling, and fully connected layers. The convolutional layer comprises a collection of filters that are used to identify features of varying sizes. Each filter convolves throughout an input image by moving horizontally for a certain amount of time, then moves vertically for another amount of time until the whole image has been convolved. A nonlinear activation function, which is the ReLU, is then applied to the convolution process' outputs. The layer which pools the neuron cluster outputs from the convolution layer into a single neuron is called the pooling layer. The pooled output is then given to a fully connected layer, which adds a bias vector and multiplies it by a weight matrix before feeding it to a softmax layer, which executes the classification operation (output). The architecture of a CNN model is shown in Fig. 1.

B. *ResNet-18 Architecture*

ResNet is suggested in [22] as a solution to the issues of performance deterioration and gradient vanishing caused by the depth expansion of an CNN model. Convolution layers, pooling layers, fully linked layers, softmax layers, and shortcut connections make up the architecture of a ResNet-18 model shown in Fig. 2. The shortcut connections represent the connections that travel between two layers. There are two main kinds of pooling layers in the ResNet-18 model architecture in this research. The first of these layers is the max-pooling layer, which chooses the maximum element from the area of the feature map covered by the convolution filter. For the second layer which is the average pooling layer, instead of picking the maximum element, it works by calculating the average value of the element from the region of the feature map.

The building layer of ResNet-18 is seen in Fig. 3 with an input x parameter and the desired output $H(x)$. The block makes use of a shortcut connection that enables it to immediately learn the residual $F(x) = H(x) - x$ in order to generate the desired output $[F(x) + x]$, hence avoiding performance deterioration and gradient vanishing due to an excessive number of convolutional layers.

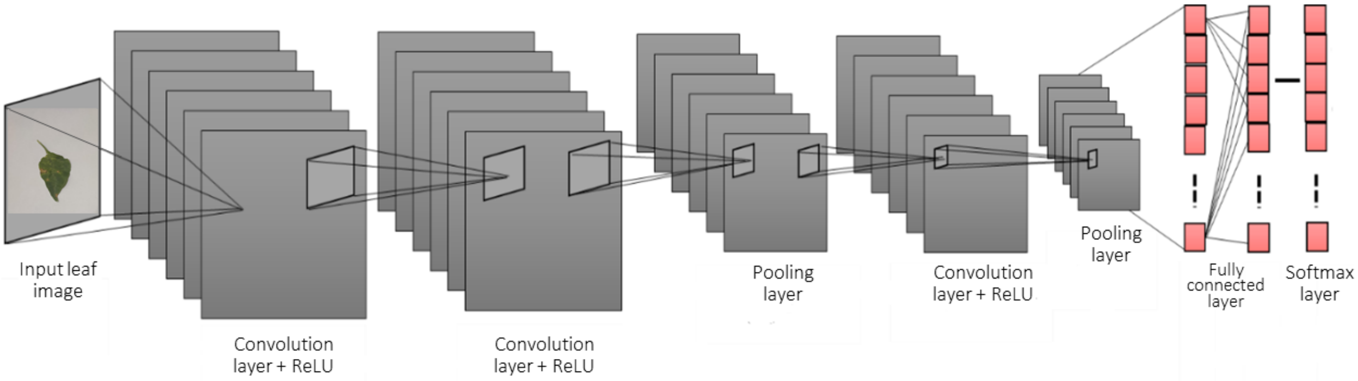

Fig. 1. CNN Model Architecture.

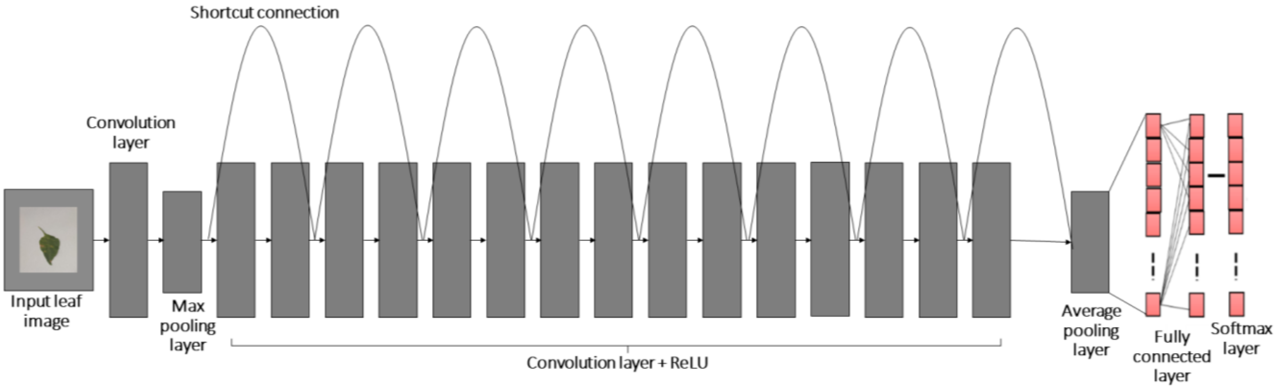

Fig. 2. ResNet-18 Model Architecture.

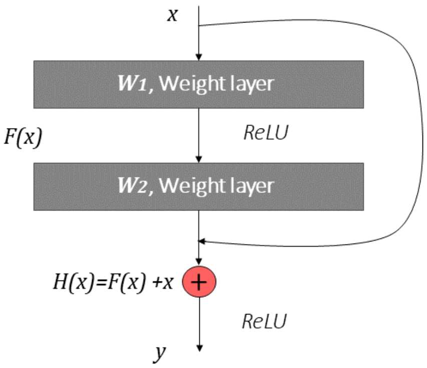

Fig. 3. The Building Layer of ResNet-18.

The ResNet building layer in Fig. 3 employs the residual mapping function [23] described below , where σ:

$$P = W_2\sigma(W_1 x) \tag{18}$$

denotes the activation function of ReLU. Through a second activation function of ReLU, the output y can be obtained:

$$y = F(x, \{W_i\}) + x \tag{19}$$

A linear transformation of output y can also be obtained by multiplying W_s to x in (19) as shown below.

$$y = F(x, \{W_i\}) + W_s x \tag{20}$$

V. Results and Discussion

The accuracy performance results for both the CNN and ResNet-18 models on original and augmented datasets of healthy and diseased chili leaf are acquired via the use of experimental setup and testing, which are detailed in the following section.

A. Experimental Setup

All of the models in the experiments run on MATLAB® with an Intel® CoreTM i3 processor operating at 3.4 GHz. The models' networks are fed data from three different categories of datasets: datasets with only original data (original datasets), datasets with only augmented data (augmented datasets), and datasets with both original and augmented data (original + augmented datasets). Only 1200 images from each category datasets are fed into the models in order to preserve the data balance. Therefore, 40 images from the original datasets are chosen to be augmented, consisting of 20 random images of healthy chili leaf and 20 random images of diseased chili leaf. During the augmentation process, which comprises of 15 geometric transformations, 600 augmented images are produced and preserved in the augmented datasets, while the original images are discarded. For original + augmented datasets, 300 images of original and augmented healthy chili leaf, as well as 300 images of original and augmented diseased chili leaf, are used. Table I shows the specific information for all the datasets in this research.

TABLE I. DATASETS INFORMATION

Type of datasets	*Ori_HC	*Ori_DC	*Aug_HC	*Aug_DC	Total images
Original	600	600	0	0	1200
Augmented	0	0	600	600	1200
Original + Augmented	300	300	300	300	1200

*Ori_HC = Original images of healthy chili leaf
*Ori_DC = Original images of diseased chili leaf
*Aug_HC = Augmented images of healthy chili leaf
*Aug_DC = Augmented images of diseased chili leaf

When the CNN and ResNet-18 models are fed with a dataset, 70% of the data in the dataset is utilised to train the models, while the remaining 30% is used to test the models. During training, the hyperparameter settings [24] of both models, such as batch size, learning rate, maximum epoch, testing frequency and optimizer, are fixed such that the optimum performance of both models is equal. Table II summarises the fixed hyperparameter settings for both models.

TABLE II. FIXED HYPERPARAMETER SETTINGS

Hyperparameter setting	Value
Batch size	32
Learning rate	0.0001
Maximum epoch	30
Testing frequency	50
Optimizer	Stochastic gradient descent with momentum (sgdm)

In each experiment, the accuracy performance of a developed model given an input dataset is determined using the following formula:

$$Accuracy_{Od} = \frac{Testing\ accuracy_{Ori\_HC} + Testing\ accuracy_{Ori\_DC}}{2} \quad (21)$$

$$Accuracy_{Ad} = \frac{Testing\ accuracy_{Aug\_HC} + Testing\ accuracy_{Aug\_DC}}{2} \quad (22)$$

$$Accuracy_{OAd} = \frac{Testing\ accuracy_{Oad\_HC} + Testing\ accuracy_{Oad\_DC}}{2} \quad (23)$$

*Od = Original datasets

*Ad = Augmented datasets

*OAd = Original + augmented datasets

*Oad_HC= Ori_HC + Aug_HC

*Oad_DC = Ori_DC + Aug_DC

B. *Experimental Testing*

Fig. 4 displays the geometric transformations that were performed on images of healthy and diseased chili leaf in order to create augmented images. The performed geometric transformations are based on the process equations in section III.

The accuracy results obtained by developed models when applied to the three categories of datasets, which are referred to as original datasets, augmented datasets and original + augmented datasets are shown in Table III.

Table III shows that after applying both models to the three categories of datasets, the accuracy performance achieved for each dataset varies. The CNN model produced accuracies of 92.3%, 82.5%, and 94.2% for the original datasets, augmented datasets, and original + augmented datasets, respectively. Conversely, the ResNet-18 model achieved accuracies of 99.2%, 91.8 %, and 99.7% for the original datasets, augmented datasets, and original + augmented datasets, respectively. In every experiment undertaken, it can be concluded that the ResNet-18 model outperforms the CNN model in terms of accuracy.

Despite the fact that both models were trained on 600 images per class, the average accuracy attained from the original datasets was only 95.8%. The geometric transformation method improved performance and yielded the best accuracy result of the two models, with an average recognition accuracy of 97% from both models that can be seen from original + augmented datasets. These findings indicate that the geometric transformation method improves the abilities of the models to generalise [25] by modifying the orientation of original image while retaining its original information.

On the other hand, the accuracy results from the augmented datasets showed that if the models were only trained with augmented images, the accuracy dropped by 8.6% and 9.8% on average compared to the accuracy of the original datasets and the original + augmented datasets. This is due to the black pixel areas (background areas) in the augmented images created by geometric transformation and the absence of the attention mechanism [26] that is found in the original images. The deep learning models use more background areas of the augmented images as distinct regions in the training process, leading to lower accuracy performance.

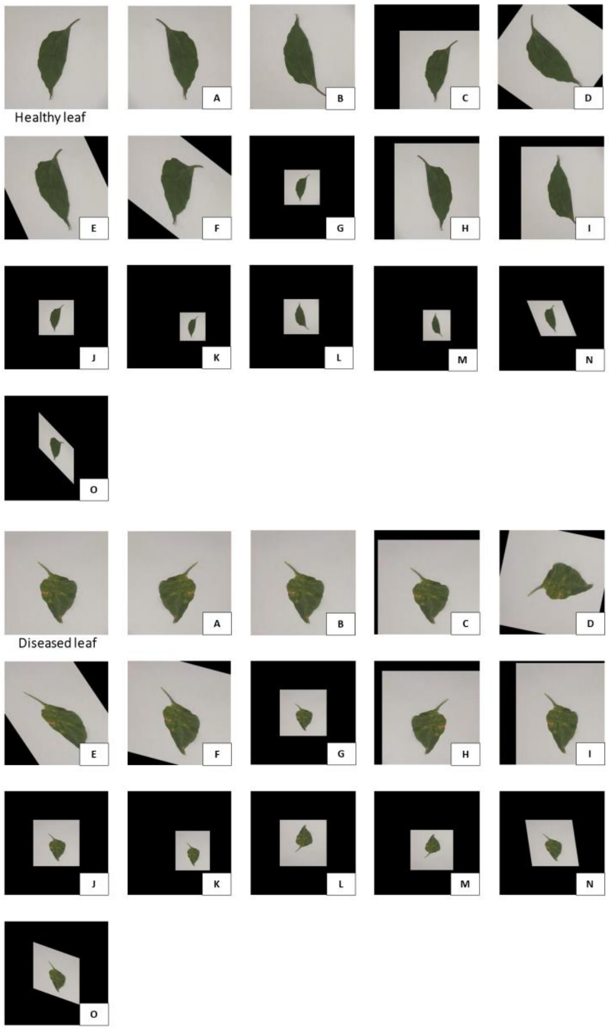

Fig. 4. Applied Geometric Transformation on Healthy and Diseased Chili Leaf Images. A) Reflection on tℎe x-axis. B) Reflection on tℎe y-axis. C) Translation. D) Rotation. E) Shearing on the x-axis. F) Shearing on the y-axis. G) Scaling. H) Reflection on tℎe x-axis and Translated. I) Reflection on tℎe y-axis and Translated. J) Scaling and Reflected on the x-axis. K) Scaling and Reflected on the y-axis. L) Scaling and Sheared on the x-axis. M) Scaling and Sheared on the y-axis. N) Scaling, Reflected on the x-axis and then Translated. O) Scaling, Reflected on the y-axis and then Translated.

TABLE III. ACCURACY RESULTS OF DATASETS

Type of datasets	Accuracy given by CNN	Accuracy given by ResNet-18
Original datasets	92.3%	99.2%
Augmented datasets	82.5%	91.8%
Original+augmented datasets	94.2%	99.7%

VI. Conclusion

This research proposed the data augmentation method known as a geometric transformation and its several transformation fusions on a small chili dataset and tested on two deep learning models, CNN and ResNet-18. A clear improvement in accuracy performance results were seen for both models after adding augmented images into the original datasets. The accuracy of both models went up by 94.2% for the CNN model and 99.7% for the ResNet-18 model.This suggests that a combination of the original and augmented images can improve the accuracy performance of the models substantially. Further research also revealed that ResNet-18 had the highest accuracy performance among both models when no data augmentation was used.

Acknowledgment

This research was supported by Universiti Tun Hussein Onn Malaysia (UTHM) through Tier 1 (vot H916). Special appreciation to the IoT Laboratory, UTHM, for providing related facilities.

References

[1] H. Thakur, S. K. Jindal, A. Sharma, and M. S. Dhaliwal, “A monogenic dominant resistance for leaf curl virus disease in chilli pepper (Capsicum annuum L.),” Crop Prot., vol. 116, pp. 115–120, 2019, doi:10.1016/j.cropro.2018.10.007.

[2] L. Colney, W. Tyagi, and M. Rai, "Morphological and molecular characterization of two distinct chilli cultivars from North-Eastern India with special reference to pungency related genes," Sci. Hortic. (Amsterdam), vol. 240, pp. 1–10, 2018, doi: 10.1016/j.scienta.2018.05.045.

[3] M. Lu, C. Chen, Y. Lan, J. Xiao, R. Li, J. Huang, Q. Huang, Y. Cao and C. T. Ho, “Capsaicin-the major bioactive ingredient of chili peppers: bio-efficacy and delivery systems,” Food Funct., vol. 11, no. 4, pp. 2848–2860, 2020, doi: 10.1039/D0FO00351D.

[4] G. Dhingra, V. Kumar, and H. D. Joshi, “Study of digital image processing techniques for leaf disease detection and classification,” Multimed. Tools Appl., vol. 77, no. 15, pp. 19951–20000, 2018, doi: 10.1007/s11042-017-5445-8.

[5] S. Mishra, A. Dash, and L. Jena, “Use of deep learning for disease detection and diagnosis,” Bio-inspir. Neurocomput. (Singapore), pp. 181–201, 2021, doi: 10.1007/978-981-15-5495-7_10.

[6] A. C. Ferreira, L. R. Silva, F. Renna, H. B. Brandl, J. P. Renault, D. R. Farine, R. Covas and C. Doutrelant, “Deep learning-based methods for individual recognition in small birds,” Methods Ecol. Evol., vol. 11, no. 9, pp. 1072–1085, 2020, doi: 10.1111/2041-210X.13436.

[7] G. Sahuri and Rosalina. “Implementation of deep learning methods in detecting disease on chili leaf,” Advanc. Stud. Comput., Sci. Eng., vol. 9, no. 6, pp. 10-15, 2020.

[8] T. L. Lin, H. Y. Chang, and K. H. Chen, “Pest and disease identification in the growth of sweet peppers using faster R-CNN,” IEEE Cons. Elect. (Taiwan), 2019, doi: 10.1109/ICCE-TW46550.2019.8991893.

[9] Y. H. Gu, H. Yin, D. Jin, J. H. Park, and S. J. Yoo, “Image-based hot pepper disease and pest diagnosis using transfer learning and fine-tuning,” Front. Plant Sci., vol. 12, 2021, doi: 10.3389/fpls.2021.724487.

[10] K. Deeba and B. Amutha, “ ResNet - deep neural network architecture for leaf disease classification,” Microprocess. Microsyst., 2020, doi: 10.1016/j.micpro.2020.103364.

[11] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” J. Big Data, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[12] S. Tim, G. Ian, Z. Wojciech, C. Vicki, R. Alec and C. Xi, “Improved techniques for training GANs,” arXiv preprint, 2016, doi:10.48550/arXiv.1606.03498.

[13] L. Taylor and G. Nitschke, “Improving deep learning with generic data augmentation,” IEEE Comput. Intel., 2018, doi: 10.1109/SSCI.2018.8628742.

[14] A. Mikołajczyk and M. Grochowski,“Data augmentation for improving deep learning in image classification problem,” Inter. PhD. Work.,pp. 117–122, 2018, doi: 10.1109/IIPHDW.2018.8388338.

[15] P. Pawara, E. Okafor, L. Schomaker, and M. Wiering, “Data augmentation for plant classification,” Inter. Conf. Advanc. Conc. Intel. Visi. Syst., pp. 615–626, 2017, doi: 10.1007/978-3-319-70353-4_52.

[16] X. Wang, K. Wang, and S. Lian, “A survey on face data augmentation for the training of deep neural networks,” Neural. Comput. Appl., vol. 32, no. 19, pp. 15503–15531, 2020, doi: 10.1007/s00521-020-04748-3.

[17] A. Vyas, S. Yu, and J. Paik, “Fundamentals of digital image processing,” Sign. Commun. Tech. (Singapore), pp. 3–11, 2018, doi: 10.1007/978-981-10-7272-7_1.

[18] L. Sifre and S. Mallat, “Rotation, scaling and deformation invariant scattering for texture discrimination,” IEEE Comput. Vision. Pattern. Recog., 2013, doi: 10.1109/CVPR.2013.163.

[19] Borda, M., Terebes, R., Malutan, R., Ilea, I., Cislariu, M., Miclea, A., and Barburiceanu, S., “Supervised deep learning classification algorithms. randomness and elements of decision theory applied to signals, pp. 205–215, 2021, doi.:10.1007/978-3-030-90314-5_15.

[20] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” Comput. Electron. Agric., vol. 147, pp. 70–90, 2018, doi:10.1016/j.compag.2018.02.016.

[21] R. B. Arif, M. A. B. Siddique, M. M. R. Khan, and M. R. Oishe, “Study and observation of the variations of accuracies for handwritten digits recognition with various hidden layers and epochs using convolutional neural network,” Elect. Eng. Inform. Comm. Tech., 2018, doi: 10.1109/CEEICT.2018.8628078.

[22] S. Kiranyaz, T. Ince, and M. Gabbouj, “Real-time patient-specific ECG classification by 1-D convolutional neural networks,” IEEE Trans. Biomed. Eng., vol. 63, no. 3, pp. 664–675, 2016, doi: 10.1109/TBME.2015.2468589.

[23] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Confer. Comput. Visi. Patter. Recog., 2016, doi: 10.1109/CVPR.2016.90.

[24] F. Hutter, L. Kotthoff, and J. Vanschoren, ”Automated machine learning: methods, systems, challenges,” Springer, 2019, doi: 10.1007/978-3-030-05318-5.

[25] G. Li, L. Liu, G. Huang, C. Zhu, and T. Zhao, “Understanding data augmentation in neural machine translation: two perspectives towards generalization,” Emp. Method. Nat. Lang. Process., 2019, doi: 10.1109/CVPR42600.2020.00815.

[26] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, “Attention branch network: learning of attention mechanism for visual explanation,” Comput. Vision. Pattern. Recog., 2019, doi: 10.1109/CVPR.2019.01096.