STAug: Copy-Paste Based Image Augmentation Technique Using Salient Target

JI-SOO KANG1 AND KYUNGYONG CHUNG2

1Department of Computer Science, Kyonggi University, Suwon-si 16227, South Korea
2Division of AI Computer Science and Engineering, Kyonggi University, Suwon-si 16227, South Korea

Corresponding author: Kyungyong Chung (dragonhci@gmail.com)

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2020R1A6A1A03040583).

ABSTRACT High-quality, large-capacity data are essential for training a deep learning vision model. However, to construct crop image data, absolute growth time is required for crop growth. In addition, it is characterized by unbalanced data, with fewer abnormal data than normal data. Therefore, building high-quality, large-scale datasets is challenging. Many studies have used data augmentation of plant images to solve this problem. However, plants require data augmentation that does not compromise their color, texture, or shape. This study proposes the use of salient target augmentation (STAug) as a data augmentation technique to protect the colors and shapes of plant images. The proposed method pastes one image’s salient target into a different image to mix the two images. It uses a salient object detection model to generate a salient object mask of the plant. Using the generated mask, a salient target was identified and cropped in a plant image, and the cropped image data were pasted to different background data for augmentation. Concat mask, a combination of each image’s salient object mask, was designed to create the label of the generated image. It is possible to create a rigid classification model by augmenting the data without damaging the plant features. To verify the performance of the proposed STAug, we compared its performance with that of other data-augmentation policies. When STAug and other augmentation techniques were applied in combination, an accuracy of 0.9733 was achieved. We demonstrated a better classification performance than when it was not applied.

INDEX TERMS Copy-paste, image augmentation, plant disease classifier, salient object detection.

I. INTRODUCTION

Data on plant diseases, plant colors, shapes, and textures are very significant. This is because diseases and harmful insects change the color, shape, and texture of plants. Unique colors and shapes appear, depending on plant diseases, harmful insects, or severity. Color, shape, and texture are important features for classifying diseases and insects [1]. It is necessary to use an image augmentation methodology that makes it possible to augment the data properly without any contamination of the main features of the plant. In a previous study, images were generated by applying basic manipulation or a deep learning model to plant data augmentation [2]. Basic image manipulation techniques include flipping, cropping, rotation, random erasing, geometric transformations, kernel filters, and feature space augmentation [3]. These data augmentation policies do not significantly transform the features of an original image, therefore they are limited in securing diversity and rigidity for learning. There has also been a study on data augmentation to generate new images using deep learning models, such as GAN or neural style transfer [4]. However, these deep-learning-based data generation methods contain difficult model learning and require significant effort. Paradoxically, extensive data are required to create a model. Figure 1 illustrates the changes in the texture, shape, and color of the tomato leaves. The images on the left show the leaves damaged by cabbage caterpillars, polyphagotarsone-mus latus, and canker.
II. RELATED WORKS

A. IMAGE AUGMENTATION TECHNIQUES FOR CLASSIFICATION MODELS

Large amounts of data are essential for deep learning-based image classification model learning. If sufficient data for learning are not obtained, overfitting directly affects the performance. However, it is difficult to build a large amount of quality data in specific domains such as healthcare and plants. Therefore, several previous studies have attempted to apply an image data augmentation technique to improve the size and quality of the training dataset [7], [8]. Image data augmentation techniques include modifying an original image for data augmentation [9], AutoAugment methods as basic modification techniques for determining an optimal combination policy [10], [11], [12], [13], and techniques for generating a new image using a deep learning model, such as GAN or neural style transfer [14], [15], [16]. Among them, the augmentation technique of modifying an original image can be easily used and expanded for application to deep vision learning. To generate a new image, the original image can be modified by transforming a single image or mixing at least two images. Figure 3 shows the classification of the image data augmentation techniques.

Image augmentation techniques for creating a new image by synthesizing at least two image samples have recently become popular owing to their simplicity and effectiveness. These methods generate new image data by overlaying at least two images or cutting and pasting them. Mixed images are randomly selected such that different labels are mixed in
one image. Therefore, the methods correct labels depending on the ratio or transparency before using them for learning. As a data augmentation technique, random image cropping and patching (RICAP) randomly selects four images, cuts them randomly, and generates one image. RICAP aids convolutional neural network learn associations between image objects and backgrounds. If a cut image has a background without objects, the network should identify features in the background rather than objects to learn. Overfitting can be prevented in this process [17]. A mixed image label is generated based on the ratio of the cut image. We assume that this technique can be applied to image augmentation. In this case, it is possible to utilize minor information generally ignored, such as small or partial features and background, at the time of model learning.

Mixup generates a new image by linearly interpolating two images [5]. Equation (1) defines the mixup. In Equation (1), \( \lambda \) represents the ratio of two images after the extraction from the beta distribution, \((\tilde{x}, \tilde{y})\) represents a label pair with the image in a data set, \((x, y)\) represent the virtual vector pair created by Mixup.

\[
\tilde{x} = \lambda x_i + (1 - \lambda) x_j, \quad \tilde{y} = \lambda y_i + (1 - \lambda) y_j \quad (1)
\]

Mixup is a learning-based vicinal risk minimization type that smoothly generates a decision boundary between the two classes. Overfitting occurs less frequently because it utilizes the vicinal distribution of the dataset and training dataset learning. This method aids to generalize a neural network model and is often used because of its simple implementation. In addition, because it linearly combines two different data, it has been applied not only to image data classification but also to sentence classification [18], audio classification [19], and GAN-based image generation [20].

Local dropout removes the pixels of a particular image region, including information, and inserts black or random noise patches for image-data training [21]. It is often used to enhance the performance of the classification models. However, it contains information on lost dropped regions. This is because some parts are not used in the image data [22]. Accordingly, CutMix was proposed to insert a different image in the patch region, which was replaced by noise pixels, and generated new image data [6]. In other words, an image with a different image that replaced a noise patch was inserted into one image. The ratio of the patch to be cut in the original image to the background was determined randomly and reflected in a label. CutMix-based image augmentation is effective for object detection and classification. In addition, it is possible to secure image diversity and increase generalization performance because the other two images are randomly selected and mixed based on the proportions of the patches [6]. Figure 4 shows mixed-sample-based image augmentation, which is the image data generated by RICAP [17], Mixup [5], and CutMix [6].

### B. CUT-PASTE BASED IMAGE ENHANCEMENT TECHNIQUE

Some studies have attempted to apply data augmentation techniques to improve object detection and segmentation accuracy. The annotation information of the segment mask and bounding box used in the two domains is saved in pixel units. Therefore, it is necessary to change the annotation to apply data augmentation [23]. Applying conventional data augmentation is complicated, such as single-image flipping and rotation or CutMix of cutting and mixing at least two images. Although there has been a study on changing an object’s position in a single image [24], it is difficult to secure diversity because data are augmented through a bit of transformation in a single image. Therefore, it is necessary to develop an augmentation technique to significantly augment the data with minimal overhead.

Inserting a segmented object into a different image was proposed as a data augmentation technique for object detection and instance segmentation. The first proposed method is image rendering of general 3D-modeled objects [25], [26]. However, [25], [26] generated a virtual segment object through 3D modeling such that the object was different from the data in the real world. Therefore, a cut paste using a real-instance object was designed [27]. As in its naming, the method cuts or copies a particular object and then pastes it into a different image to generate new image data.

For cut-paste-based data augmentation, three points must be considered. The first is to select source images for the objects to be cut and the target image that will become a background. The second is to choose the number of segment objects cut in the selected source images and select the final segment object group to copy. The last is to select the position of the selected target image in which the segment object is pasted. Many studies have been conducted on natural images considering the virtual context of images when a segment...
object is pasted in the third point [27], [28]. In these studies, the visual context was modeled to generate realistic images, and thus, the pasted segment objects were made as natural as possible. For example, a flight object was positioned on a sky background, and the lamb objects are on a farm. Ghiassi et al. [29] proposed a simple copy-paste that made it possible to select a segment object randomly and paste it in a random position in the background data without considering the visual context and proved that the proposed method had high performance and efficiency through testing [29]. Using a random copy-paste object, selecting a background position for the paste, and selecting the number of segment objects is conducted entirely randomly. Therefore, it supports easy implementation and excellent portability, and requires no additional cost or inference time for learning. As the method was tested with the COCO [30] and LVIS [31] datasets, state-of-the-art was achieved at that time.

III. STAug: COPY-PASTE BASED IMAGE AUGMENTATION TECHNIQUE USING SALIENT TARGET

A. MASK GENERATION AND SALIENT TARGET EXTRACTION WITH SALIENT OBJECT DETECTION

Plant disease data generally consists of at least one type of plant image and annotation, and the annotation mainly contains the class for each disease [32], [33]. Some detailed annotated data include metadata such as date or area, and others often include plant and disease codes, severity of disease, and coordinates of the bounding box of a salient object [34]. In other words, most plant blight data have no segmentation mask information for the plant instance with blight in the pixel unit. The STAug proposed in this study is based on the copy-paste augmentation used in segmentation; therefore, it requires mask annotation. To apply the proposed STAug, a plant mask must be extracted from the image. We used a deep learning model to create the mask information of a segmented object with blight.

The main methods used by the deep learning model to create a mask in an image are instance segmentation and semantic segmentation. Mask R-CNN and DeepLab are typical deep learning models that perform high instance segmentation. A deep-learning model based on instance segmentation requires considerable resources and time. Generating a mask is a preliminary work to apply STAug, which was proposed as a data augmentation method. Therefore, it is necessary to use a technique to generate a mask in a short time and produce good performance. A conventional instance segmentation deep learning model is not an appropriate strategy for creating a mask. In the case of plant blight data, one object, rather than multiple objects, exists at the center of an image [34]. The mask generated in a deep learning model should be a binary mask for a related object. A conventional instance segmentation model segments an image based on an object such that it is inappropriate. Therefore, this study applies salient object detection (SOD) to generate a mask. SOD is a technique that detects the most salient object in an image and generates a binary mask [35]. It calculates the probability that a salient object belongs to a specific pixel in an image to determine the most salient object; therefore, it is a relatively lightweight model [36]. Because SOD does not require object classification or the detection of multiple objects, it generates a salient object’s mask quickly and produces high accuracy. Therefore, SOD is suitable for plant blight data that feature the location of one blight plant at the center of an image and the fast and accurate detection of a binary mask for the plant. Owing to its fast mask generation speed, preliminary work is appropriate for applying STAug.

BASNet [36] was used to extract a plant’s mask from among the salient object detection models. BASNet uses a pre-trained model with DUSTS-TR [35] data. BASNet has an U-Net structure consisting of a prediction module (En-De) and a residual refinement module. The residual refinement module generated a sharp mask boundary. In a single GPU, it took approximately 0.015 s (70 fps) to infer a 320 × 320 (height × width) image. As such, its speed is relatively high. Therefore, BASNet is an appropriate model for mask generation, which is a preliminary work for applying STAug.

Figure 5 illustrates the results of the BASNet-based salient object detection. As shown in Fig. 5, a plant is detected as a salient object in an image, and a mask is generated. SOD generates a plant mask and extracts a ST for image augmentation using the generated mask. BASNet, as a mask generation network, is pretrained with data other than plant image data. Therefore, it is difficult to ensure that the mask boundary is sharply cut in a plant image. To smoothen the generated mask boundary and cut a ST naturally, a Gaussian blur is applied to the mask boundary. Thus, it is possible
to secure the margin of safety of the mask and an actual plant, and extract a ST without significant loss of the plant. Equation (2) describes the operation for extracting a ST from a plant image using a mask.

\[
\text{Salient Target } (I) = x_\alpha (I) \land x_\beta (I)
\]  

(2)

Using Equation (2), it is possible to extract the overlaid parts of the two images. The extracted image is a ST. Figure 6 illustrates the process of extracting a ST using a mask.

![Figure 6. The results of extracting a salient target with the mask.](image)

**B. DATA AUGMENTATION USING STAUG**

The proposed STAug utilizes a mask-based ST extracted from plant image data using salient object detection. STAug, proposed as an image augmentation method, is used to make a solid model during training and utilizes the copy-paste image augmentation methodology for image instance segmentation. Figure 7 illustrates the STAug application process.

In Equation (3), \( \lambda \) indicates the ratio of \( x_\alpha \) to \( x_\beta \) in the ST for the new image \( \tilde{x} \). For the ratio \( \lambda \), the masks used for the salient object extraction of \( x_\alpha \) and \( x_\beta \) are merged, and a concat mask is generated. The generated concat mask is the final binary mask of the new image data, \( \tilde{x} \). The mask size of the salient object \( x_\beta \) positioned in front is set to \( \lambda \), and the mask size of the non-salient object \( x_\alpha \) is set to \( 1-\lambda \). Because \( \lambda \) follows a beta distribution, the relative ratio \( \lambda \) should be calculated when the total size of the generated concat mask is estimated to be one. Figure 8 shows the process of generating the concat mask of \( \tilde{x} \) and calculating the combination ratio \( \lambda \).

![Figure 7. Process of applying STAug.](image)

![Figure 8. Process of generating the concat mask of \( \tilde{x} \) and calculating the combination ratio \( \lambda \).](image)
background image \( \alpha \) with image \( \beta \), including the ST to paste, and generating a new label \( \tilde{y} \) for training is possible.

\[
\text{loss}(\tilde{y}) = (1 - \lambda) \times \text{loss}(y_\alpha, f(\tilde{x})) + \lambda \times \text{loss}(y_\beta, f(\tilde{x}))
\]

\[
\text{score} = (1 - \lambda) \times \text{score}(y_\alpha, f(\tilde{x})) + \lambda \times \text{score}(y_\beta, f(\tilde{x}))
\]

C. PLANT DISEASE CLASSIFICATION MODEL USING STAug

STAug was performed in each training iteration, and the application rate was determined using the application probability \( p \). The proposed STAug is independently performed from a single image modification augmentation, so that it can be applied in duplicate. In other words, it is possible to apply flipping, cropping, or rotation to images \( \alpha \) and \( \beta \) using STAug. Applying STAug only or in duplicate with other augmentation policies makes flexible extensibility possible. Because the image generated by the proposed method varies significantly, it is possible to overcome the limitation of a single image manipulation augmentation, which has difficulty securing various data distributions. In addition, by entirely copying and pasting a ST as an infected crop object, it is possible to effectively augment the data without contaminating the main features. Because the mask information on an image with no mask annotation is generated independently by salient object detection, it is possible to apply the proposed method without additional data labeling. The images generated by STAug include the following:

- mixed image \( \tilde{x} \)
- label \( \tilde{y} \) adjusted with \( \lambda \)
- bounding box of the ST
- mask annotation of the ST

Although the proposed method was applied to classification in this study, it can be applied to object detection and instance segmentation for image augmentation. Figure 9 illustrates the plant disease classification process using STAug.

EfficientNet was used as a base model in which the proposed STAug was applied to plant disease classification. EfficientNet supports compound scaling to adjust the model depth and width as well as the size of the input image, thereby determining the optimal performance [37]. Because the base model features easy and fast learning and supports a variety of scaling, it is suitable for checking whether the proposed data augmentation method significantly affects classification performance. A plant disease classification model is designed to transfer the learning of EfficientNet-b0 and EfficientNet-b3 models pre-trained with ImageNet [38] data.

The hyperparameters for learning the blight classification model are as follows. In terms of a model’s input image size, EfficientNet-b0 has an input image of \( 224 \times 224 \) (height \( \times \) width), and EfficientNet-b3 has an input image of \( 300 \times 300 \) pixels. Adam was used as an optimizer and OneCycleLR as a scheduler. The initial learning rate was set to 0.0005 and a weight decay technique was applied. Cross-entropy loss was applied as the loss function for learning. In terms of the batch size, EfficientNet-b0 used 128, and EfficientNet-b3 used 64. Learning was performed for 50 epochs.

IV. EXPERIMENTS AND RESULTS

A. EXPERIMENTAL ENVIRONMENT AND PLANT DISEASE DATA

Table 1 presents the experimental environment for the proposed STAug-based image augmentation of the plant disease classification model.

| Experiment environment | Detail |
|------------------------|--------|
| CPU                    | AMD Ryzen 9 3950X 16-Core Processor |
| GPU                    | GeForce RTX 3090 |
| RAM                    | 64GB |
| OS                     | Ubuntu 18.04.6 LTS |
| Language               | python 3.7.11 |
| Deep Learning Framework| pytorch 1.9.0 |

The plant disease diagnosis data offered by the AlHub were used [34]. These data are the disease image of ten types of horticultural crops of facilities. Among them, four
types of crops that were most cultivated and highly damaged were selected. These were eggplant, pepper, strawberry, and pumpkin. Normal images of each type of crop, seven types of diseases of the crops, and severity of disease (early, middle, and late stages) were labeled and then categorized into 25 classes. The disease data were less than normal, thus they were imbalanced. The most labeled ‘eggplant_normal’ class had 34,234 data, whereas the ‘branch_leaf-fungus_late’ class had 22 data. The difference between the class with the minor data and major data was approximately 1,500 times; therefore, a severe data imbalance occurred. The experimental dataset was split into training, validation, and test datasets in a 6:2:2 ratio. The training dataset numbered 48,171, validation dataset 16,063, and test dataset 16,059.

B. PERFORMANCE EVALUATION

Two experiments were conducted to analyze the performance of the STAug. More specifically, hyperparameter tuning for the optimal application of STAug and a performance comparison according to the application of STAug were analyzed. Some hyperparameters must be set manually when using STAug. The classification performance was based on the set values of the hyperparameters. Therefore, this study aimed to determine the optimal hyperparameters.

The hyperparameters are the STAug application probability ‘p’ during training and population from which to select the data \((x_β, y_β)\) to be pasted. This study determines the optimal application probability ‘p’ from 0 to 1. Table 2 lists the performance comparison results when STAug was applied to EfficientNet-b0 at its application rate p. Performance was lowest when STAug was never applied. When the application rate was 0.5, the accuracy was highest. In addition, the performance was low when STAug was applied to all the learning data after p was set to 1.0.

### Table 2. Performance comparison result based on STAug application rate p.

| STAug application rate \(p\) | Accuracy |
|-----------------------------|----------|
| 0                           | 0.9261   |
| 0.2                         | 0.9530   |
| 0.5                         | 0.9608   |
| 0.7                         | 0.9489   |
| 1.0                         | 0.9317   |

This indicates that the model overfits the STAug-applied data and does not accurately classify the images. Therefore, determining an optimal application rate is more important than applying STAug unconditionally. In this study, the best performance was observed when p was set as 0.5. However, this was based on the model, data, and training conditions.

For STAug, selecting the data \((x_β, y_β)\) to be pasted for each training iteration was required. For selecting data \((x_β, y_β)\), it was possible to choose one out of all data regardless of plants, disease, and severity, or select data in a particular class, if necessary. The data used in this study showed a pattern of imbalance: the concentration of data in normal classes. To maximize the data augmentation effect for the reduction in data imbalance, this study considered a method of putting weight on the minor class when data \((x_β, y_β)\) were selected. The population for selecting data \((x_β, y_β)\) was 21 out of 25 classes (excluding the top four). EfficientNet-b0, with an STAug application rate of 0.5, was used for the experiment.

Table 3 lists the performance comparison of different \((x_β, y_β)\) selection methods. Accuracy improved slightly, however the learning efficiency increased significantly. For random selection, the model achieved the highest performance at 25 epochs. In the case of minor class weighing, it achieved the highest performance at 13 epochs, saving approximately 12 h in training. In the random selection method, data \((x_β, y_β)\) were highly likely to be selected for the major class. In this case, STAug would likely augment major classes more, leading to a lower learning efficiency in the imbalanced dataset. If the minor class gained weight, STAug selected data to paste only in the minor class. Therefore, it was possible to learn more minor class data and thus support stable and fast learning in an imbalanced dataset. It is desirable to adjust and apply a selection method for data \((x_β, y_β)\) and to select a population in line with necessities and purposes.

### Table 3. Performance comparison result based on \((x_β, y_β)\) selection methods.

| \((x_β, y_β)\) selection methods | Accuracy | Convergence epoch | Training Time |
|----------------------------------|----------|-------------------|---------------|
| Random selection                 | 0.9608   | 25                | 24h 35m       |
| Minor class weighing selection   | 0.9631   | 13                | 12h 47m       |

To check whether the proposed STAug significantly affected the performance of a plant disease classification model, we conducted an ablation study of STAug. EfficientNet-b0 and EfficientNet-b3 were used as backbone networks for the classification model. Five data-augmentation policies were compared. Table 4 lists a comparison of the five augmentation policies. Albumentation [39] was used as the data-augmentation library. Each augmentation policy’s application probability ‘p’ was set to 0.2.

In the first policy, only image size adjustment and normalization are applied for classification model training. The techniques used generally for data augmentation are applied in the second policy. In the third policy, the techniques to augment images by changing image colors are applied. That analyzes how the augmentation techniques based on color changes affect plant disease data. The proposed STAug is applied to data based on the optimized hyperparameter value in the fourth policy. That is aimed at verifying how much the proposed image data augmentation technique is effective. The proposed STAug is applied together with the basic augmentation techniques often used in the fifth policy. Since STAug works independently from other augmentation techniques, it is possible to apply them duplicated. Other augmentation
techniques can be applied to the images $x_\alpha$ and $x_\beta$ all or each so that it is possible to generate a variety of data distribution. Table 5 presents a performance comparison when different augmentation policies are applied to the classification models’ ‘EfficientNet-b0’ and ‘EfficientNet-b3’.

**TABLE 4.** Augmentation policies in comparison.

| Policy               | Augmentation                              |
|----------------------|------------------------------------------|
| Original Data        | -                                        |
| Basic Augmentation   | CLAHE, RandomBrightnessContrast, ColorJitter, RGBShift, RandomSnow, RandomResizedCrop, ShiftScaleRotate, HorizontalFlip, VerticalFlip, Rotate, RandomRotate90 |
| Color Augmentation   | CLAHE, RandomBrightnessContrast, ColorJitter, RGBShift, RandomSnow |
| STAug                | CLAHE, RandomBrightnessContrast, ColorJitter, RGBShift, RandomSnow |
| STAug+               | CLAHE, RandomBrightnessContrast, ColorJitter, RGBShift, RandomSnow |

**TABLE 5.** Comparison of classification performance according to the augmentation policy.

| Policy       | EfficientNet-b0 | EfficientNet-b3 |
|--------------|-----------------|-----------------|
|              | Accuracy | F1-score | Accuracy | F1-score |
| Original Data| 0.9261   | 0.9192   | 0.9448   | 0.9393   |
| Basic Augmentation | 0.9593   | 0.9574   | 0.9659   | 0.9641   |
| Color Augmentation | 0.9509   | 0.9484   | 0.9603   | 0.9591   |
| STAug        | 0.9631   | 0.9613   | 0.9703   | 0.9688   |
| STAug+       | 0.9686   | 0.9678   | 0.9733   | 0.9729   |

According to the comparative analysis, the case where the proposed STAug was applied showed the best performance among the two classification models. In particular, STAug+, which used an additional augmentation technique based on STAug, showed the best performance. This indicates that when the proposed technique and a different augmentation technique are duplicated, the performance is improved, and such techniques can be applied variously depending on the data and purposes. This study used plant disease data whose colors, shapes, and textures were significant, and color change-based augmentation insignificantly influenced the plant disease data. However, in the case of data in which colors are not significant, it is possible to apply STAug and color change-based augmentation policies in a duplicated manner depending on the purpose.

V. CONCLUSION

This study proposes STAug, a copy-paste-based image-augmentation technique. The proposed STAug represents salient target augmentation and augments images without damaging the color, shape, and texture of images. To apply the copy-paste augmentation technique mainly used for object detection and segmentation of the classification model, BASNet, as a salient object detection model, is applied to the image data without the annotated mask and to generate a mask. Based on the generated mask, a salient object was cut into an image. A cut salient object is referred to as a ST. When a ST is cut, a Gaussian blur is applied to the mask boundary, making it smooth and natural. The cut ST was fully copied and pasted to a different background image. In this manner, a new image was generated for image augmentation, and two different images were mixed. Therefore, the labels must be changed. A cat mask was generated by combining the ST region of the two images, and then the mask ratio for the cat mask in the two images was calculated. The calculated ratio is the combined ratio, which follows the beta distribution. The ratio of a salient object was set to $\lambda$, and that of a non-salient object was set to $1-\lambda$. In this manner, the label was changed and used for learning. A plant disease classification model was designed to apply the proposed STAug. The proposed image data augmentation technique significantly improved the performance of the classification model.

To achieve the best classification model performance during training with STAug, an optimal hyperparameter must be determined. To determine whether the hyperparameter affected the performance of the classification model, we conducted an ablation study using STAug. Consequently, when STAug was applied to EfficientNet-b0 after its application rate was set to 0.5, the data $\beta$ to copy was selected in a minor class, and the classification accuracy was 0.9631, which indicates the highest performance. The proposed method supported stable and fast learning in an imbalanced dataset when data ($\lambda, \beta$) were selected to assign weight to minor classes rather than randomly. In addition, a performance comparison was conducted for the five augmentation policies. The best performance was achieved when the proposed STAug was applied. Therefore, it was proved that the proposed method achieved a significant performance improvement. Mainly, when STAug was applied together with a basic image augmentation technique in a duplicated manner, the accuracy of the EfficientNet-b3 model reached 0.9733, which indicates the best performance. It was observed that when the proposed STAug and a different technique were applied together, the performance significantly improved, and the model was successfully generalized. This indicates that the proposed augmentation technique can be extended in several ways.

In future study we will select augmentation techniques that will lead to a positive performance improvement when they are used together with the proposed STAug in a duplicated manner. In addition, we will investigate whether overfitting prevention and generalization will be made possible in datasets other than plant blight data.

REFERENCES

[1] A. J. Hati and R. R. Singh, “Artificial intelligence in smart farms: Plant phenotyping for species recognition and health condition identification using deep learning,” *AI*, vol. 2, no. 2, pp. 274–289, Jun. 2021.

[2] J. Liu and X. Wang, “Plant diseases and pests detection based on deep learning: A review,” *Plant Methods*, vol. 17, no. 1, pp. 1–18, Feb. 2021.
