JUSIKOM PRIMA (Journal of Information Systems and Computer Science Prima)
Vol. 6 No. 2, February 2023 E-ISSN: 2580-2879

THE USE OF DATA AUGMENTATION AND EXPLORATORY DATA ANALYSIS IN ENHANCING IMAGE FEATURES ON APPLE LEAF DISEASE DATASET

Rifaldo[1], Daniel Ryan Hamonangan Sitompul[2], Stiven Hamonangan Sinurat[3],
Andreas Situmorang, Julfikar Rahmad[4], Dennis Jusuf Ziegel[5], Evta Indra[*6]
[1,2,3,4,5,6]Program Studi Sistem Informasi, Fakultas Teknologi dan Ilmu Komputer, Universitas Prima Indonesia
Jl. Sampul No.3, Sei Putih Barat, Medan Petisah
E-mail : *evtaindra@unprimdn.ac.id

ABSTRAK- Apples are an essential commodity produced in Batu City, Malang. In 2017, Batu City, Malang, produced 19.1 tons of apples, while in 2018, Batu City, Malang, produced 15.9 tons of apples. It can be concluded that the decline in the number of apple harvests in Batu City, Malang. With the influence of technology in agriculture, the influence of technology can be used to detect diseases on leaves to overcome the decrease in the number of harvests. With the Image Augmentation method used in this study, the existing dataset can have 6x more features. So that the healthy category, which previously had 516 image features, now has 3096 image features, the scab category, which previously had 592 image features, now has 3552 image features and the rust category, which previously had 622 image features, now has 3732 image features. With a dataset with 3000 image features, the model to be made can have a higher accuracy value. The model can be said to be sturdy/sturdy/good, or the model to be made can carry out the classification process with a good level of accuracy.

Kata kunci : Image Augmentation, Exploratory Data Analysis, Leaf Disease, Apple

1. INTRODUCTION

Apples are an essential commodity produced in Batu City, Malang. In 2017, Batu City, Malang produced 19.1 tons of apples, while in 2018, Batu City, Malang produced 15.9 tons of apples, so it can be concluded that the decline in the number of apple harvests in Batu City, Malang [1]. This decrease in production levels can be caused by crop failure. One of the causes of crop failure is leaf disease; if it is not treated quickly, it will damage the plants and cause the number of harvests to decrease. With the development of technology in agriculture, technological developments can be used to detect or treat diseases on leaves to overcome the decrease in the number of harvests. However, the availability of datasets on agricultural land is still relatively low, so we need a way to increase the variety/features in the dataset to increase the accuracy of the leaf disease detection model.

Previous studies that have discussed the classification of diseases on leaves have been carried out; an example is research [2]–[6]. Research [2] presents the process of adding features to leaf images using image saturation configurations. The output/results displayed in this study are leaf images that have a higher or lower contrast level. Research [3] presents the process of adding features to leaf images using rotation, saturation, and zoom configurations. The output/results displayed in this study are leaf images configured according to the provisions.

Based on the problems above, the authors suggest using image augmentation to add variations/features to the dataset. In this study, an image processing process will be made in which there are image augmentation stages to add features to the image of apple leaf disease. Image augmentation uses a python library, OpenCV, Scikit-Image, and the Exploratory Data Analysis stage with Plotly and Matplotlib. With data augmentation, the leaf disease detection model can produce a higher model accuracy and a high precision value.

2. METHODOLOGY

This research was conducted at the Data Analyst Laboratory at Universitas Prima Indonesia. The Graphic Processing Unit (GPU) from Google Colaboratory was used to conduct this research. The workflow of this research is shown in Figure 1. The research began by retrieving datasets from the Kaggle repository; the pre-processing stage was carried out after retrieving the datasets. In the pre-processing stage, the dataset will be configured with data augmentation. Data augmentation is done to get more features from an image. After pre-processing the data, the final stage that will be carried out is the exploratory data analysis stage.

2.1 Data Acquisition

Data Acquisition is a stage in Data Science that aims to retrieve data for modeling purposes [7], [8]. The dataset used to carry out Data Augmentation, and Exploratory Data Analysis (EDA) comes from the Kaggle Competition repository "Plant Pathology 2020 - FGVC7" [9] and partly comes from taking pictures of apple leaves belonging to Batu City agricultural land. This dataset contains images of apple leaves consisting of three categories: healthy

apple leaves, apple leaves with rust disease, and apple leaves with scab disease. Dataset details can be seen in Table 1.

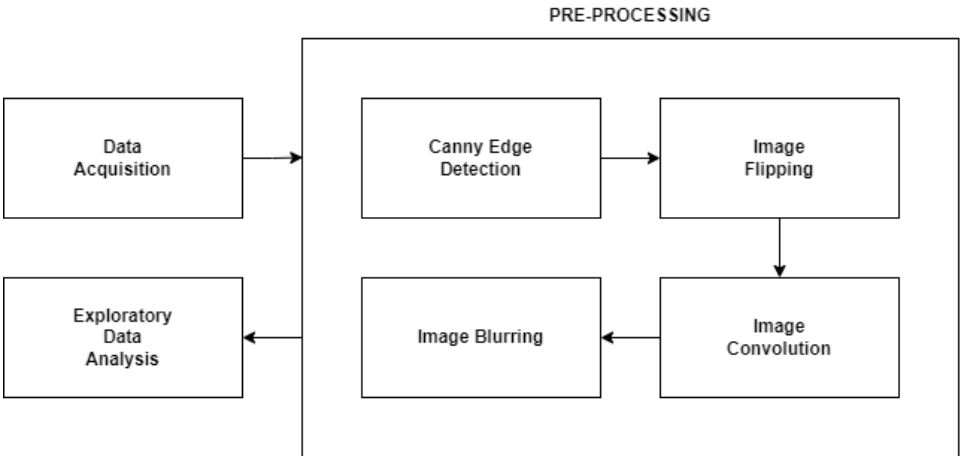

Figure 1. Research Methodology

Table 1. Dataset Details

Gambar	Kategori	Jumlah Gambar
	Healthy	516
	Scab	592
	Rust	622

2.2 Pre-Processing

2.2.1 Canny Edge Detection

Canny Edge Detection is an operator to perform edge detection of an image. Canny Edge Detection consists of 5 main parts: Noise Reduction, Gradient Calculation, Non-Maximum Suppression, Double Threshold, and Edge Tracking by Hysteresis [10], [11]. In this study, the Canny Edge operator has a function named edge_and_cut, which detects leaf edges and creates squares on leaves.

```
def edge_and_cut(img):
    emb_img = img.copy()
    edges = cv2.Canny(img, 100, 200)
    edge_coors = []
    for i in range(edges.shape[0]):
        for j in range(edges.shape[1]):
            if edges[i][j] != 0:
                edge_coors.append((i, j))

    row_min = edge_coors[np.argsort([coor[0] for coor in edge_coors])[0]][0]
    row_max = edge_coors[np.argsort([coor[0] for coor in edge_coors])[-1]][0]
    col_min = edge_coors[np.argsort([coor[1] for coor in edge_coors])[0]][1]
    col_max = edge_coors[np.argsort([coor[1] for coor in edge_coors])[-1]][1]
    new_img = img[row_min:row_max, col_min:col_max]

    emb_img[row_min-10:row_min+10, col_min:col_max] = [255, 0, 0]
    emb_img[row_max-10:row_max+10, col_min:col_max] = [255, 0, 0]
    emb_img[row_min:row_max, col_min-10:col_min+10] = [255, 0, 0]
    emb_img[row_min:row_max, col_max-10:col_max+10] = [255, 0, 0]
```

Figure 2. Making of Canny Edge Function

2.2.2 Image Flipping

Image Flipping is an operator for flipping (flipping horizontally and vertically) an image [12], [13]. In this study, Image Flipping has a function called invert(), which performs horizontal and vertical rotations of apple leaf images.

```
def invert(img):
    fig, ax = plt.subplots(nrows=1, ncols=3, figsize=(30, 20))
    ax[0].imshow(img)
    ax[0].set_title('Original Image', fontsize=24)
    ax[1].imshow(cv2.flip(img, 0))
    ax[1].set_title('Vertical Flip', fontsize=24)
    ax[2].imshow(cv2.flip(img, 1))
    ax[2].set_title('Horizontal Flip', fontsize=24)
    plt.show()
```

Figure 3. Making of Image Flipping Function

2.2.3 Image Convolution

Image Convolution is an operator to configure saturation (increase or decrease brightness) in an

image [14], [15]. In this study, the Image Convolution operator has a conv() function, configuring saturation on apple leaf images.

```
def conv(img):
    fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(20, 20))
    kernel = np.ones((7, 7), np.float32)/25
    conv = cv2.filter2D(img, -1, kernel)
    ax[0].imshow(img)
    ax[0].set_title('Original Image', fontsize=24)
    ax[1].imshow(conv)
    ax[1].set_title('Convolved Image', fontsize=24)
    plt.show()
```

Figure 4. Making of Image Convolution Function

2.2.4 Image Blurring

Image Blurring is an operator for configuring blur (blurring images) or adding noise to images [16], [17]. In this study, the Image Blurring operator has a blur() function, configuring blur on apple leaf images.

```
def blur(img):
    fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(20, 20))
    ax[0].imshow(img)
    ax[0].set_title('Original Image', fontsize=24)
    ax[1].imshow(cv2.blur(img, (100, 100)))
    ax[1].set_title('Blurred Image', fontsize=24)
    plt.show()
```

Figure 5. Making of Image Blurring Function

2.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a data analysis process to understand the contents of a dataset and to look for patterns contained in the dataset [18]–[20]. In this study, EDA was carried out to determine the mean of the color distribution in apple leaf images, the parallel categories of apple leaf images, and the dataset distribution for each disease category.

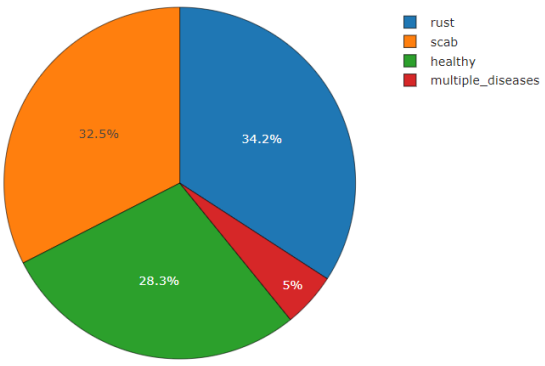

Figure 6. Exploratory Data Analysis

3. RESULT AND DISCUSSION

3.1 Data Acquisition

In this study, data acquisition (Data Acquisition) was carried out by importing data from the Kaggle repository to Google Colaboratory. Data import is done with the help of the Application Programming Interface (API). After the data is obtained, then what is done is to extract the data. Details of this process can be seen in Figure 7.

```
!cp /content/drive/MyDrive/kaggle_API_credential/kaggle.json ~/.kaggle/kaggle.json
```

```
! chmod 600 ~/.kaggle/kaggle.json
```

```
! kaggle competitions download plant-pathology-2020-fgvc7
```

```
Downloading plant-pathology-2020-fgvc7.zip to /content
 99% 775M/779M [00:07<00:00, 125MB/s]
100% 779M/779M [00:07<00:00, 115MB/s]
```

```
! unzip plant-pathology-2020-fgvc7.zip
```

```
Archive:  plant-pathology-2020-fgvc7.zip
  inflating: images/Test_0.jpg
  inflating: images/Test_1.jpg
```

Figure 7. Steps of Data Acquisition

After the data is extracted, the next step is to preview the dataset that has been taken. In this study, each image in the dataset has been categorized in CSV format to reduce file size. Details of this process can be seen in Figure 8.

```
train_data.head()
```

	image_id	healthy	multiple_diseases	rust	scab
0	Train_0	0	0	0	1
1	Train_1	0	1	0	0
2	Train_2	1	0	0	0
3	Train_3	0	0	1	0
4	Train_4	1	0	0	0

Figure 8. Dataset Preview

3.2 Canny Edge Detection

In this study, Canny Edge Detection has stages: Noise Reduction, Image Gradient Intensity Search, Rounding, Non-Maximum Suppression, and Hysteresis Thresholding. Noise Reduction is made using a 5x5 Gaussian Filter, Gradient Image Intensity Search is done with a Sobel Kernel filter in each direction (horizontal or vertical), and Rounding is done by rounding the four corners, which represent the vertical, horizontal, and both sides of the diagonal, Non-Maximum Suppression is done by how to scan the image to remove pixels that are not needed and Hysteresis Thresholding is done to detect which part is a corner or side. The result of this step is a two-dimensional binary map (0 or 255) that displays the angles in the image. A preview of the results of this operator can be seen in Figure 9.

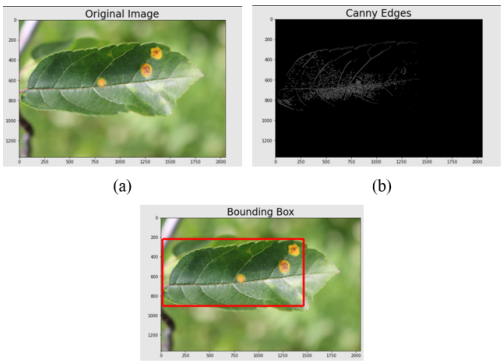

Figure 9. a) Original Image, (b) Canny Edge Process, (c) Bounding Box

3.3 Image Flipping

In this study, Flipping is done using a simple transformation involving index-switching on the image channel. In vertical Flipping, the row order is changed, while in horizontal Flipping, the column order is changed. The results of this operator can be seen in Figure 10.

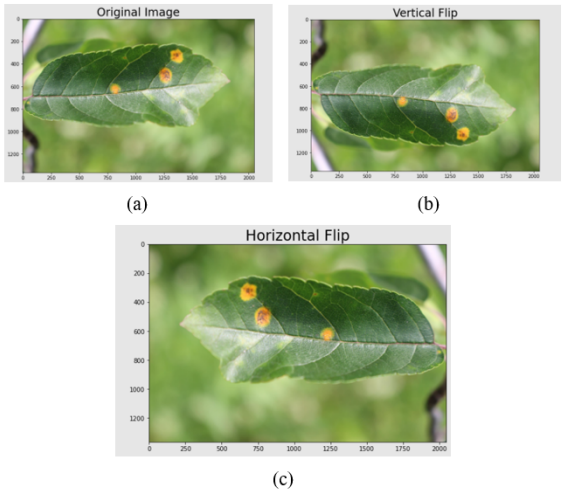

Figure 10. a) Original Image, b) Vertical Flip, c) Horizontal Flip

It can be seen that the image of the apple leaf is only inverted vertically and horizontally. The general features of the images still look the same, but for computer algorithms, the inverted images have different features. This transformation can make deep learning models that will be used better/more robust.

3.4 Image Convolution

In this study, Image Convolution was performed simply, involving a moving kernel (2D matrix) to scan the entire image and calculate each point in the pixel. The results of this operator can be seen in Figure 11.

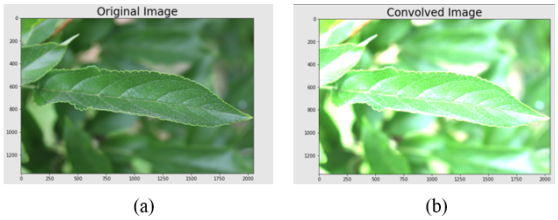

Figure 11. a) Original Image, b) Convolved Image

It can be seen that this operator has a contrast enhancement effect (Sunshine Effect) on apple leaf images. The resulting output can be used as an additional feature to add a more accurate/robust model.

3.5 Image Blurring

In this study, Image Blurring adds noise to the apple leaf image, which can produce a less clear image. The results of this operator can be seen in Figure 12.

Figure 12. a) Original Image, b) Blurred Image

It can be seen that the transformation of Image Blurring can blur images by reducing low-level features and also maintaining significant and high-level features. The resulting output can be used as an additional feature to add a more accurate/robust model.

3.6 Exploratory Data Analysis

In this study, the Exploratory Data Analysis stage was carried out to find the mean (mean) of the color distribution of apple leaf images, find out the parallel categories of apple leaf images and divide the dataset for each.

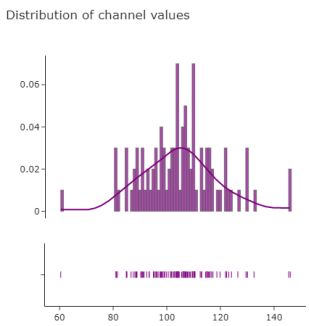

Figure 13. Distribution of Channel Values

The channel values contained in the plot above are typically distributed and centered on a value of 105.

The maximum value of the activated channel is 255. This means that the mean of the channel values is less than half of the maximum value. It can be concluded that each channel is activated at a minimum.

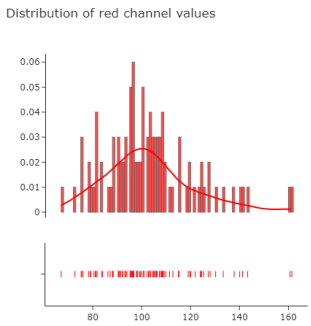

Figure 14. Distribution of Red Channel Values

The red channel has typical values but has a positive skew (decreasing to the right). It can be concluded that the red color channel is concentrated at low values, around 100. There are also many variations in the red color values in all the apple leaf images.

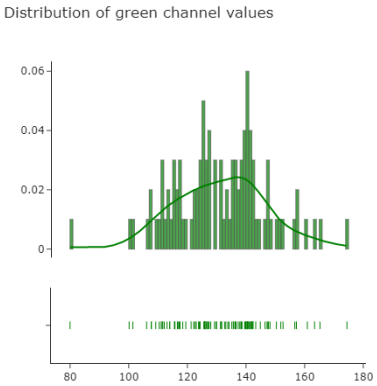

Figure 15. Distribution of Green Channel Values

The green channel has a more uniform distribution than the red channel and a smaller peak value. The distribution has a decreasing shape to the right and has a mode of 140. It can be concluded that more green colors are found because the leaves are green.

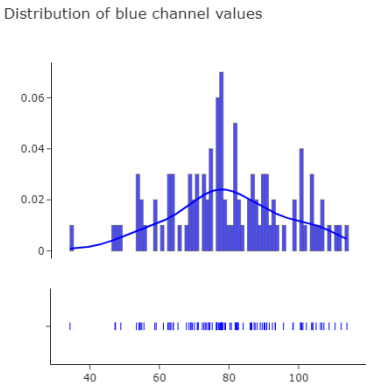

Figure 16. Distribution of Blue Channel Values

The blue channel has the most uniform distribution values compared to other colors with minimal skew. The blue color channel has many variations across all images.

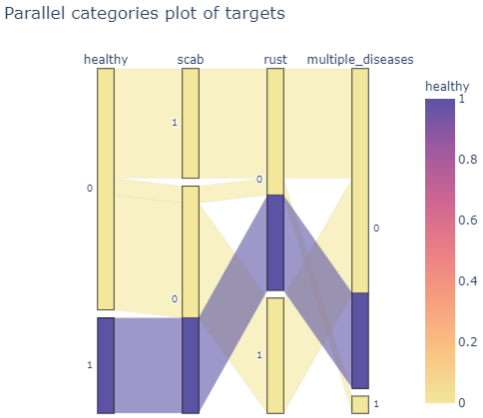

Figure 17. Paralel Category of Apple Leaf Disease

In the graph above, it can be seen that the relationship between the four categories is contained in the dataset. As predicted, a healthy leaf (healthy == 1) definitely has no disease. Also, all unhealthy leaves can have either scabs, rust, or multiple diseases.

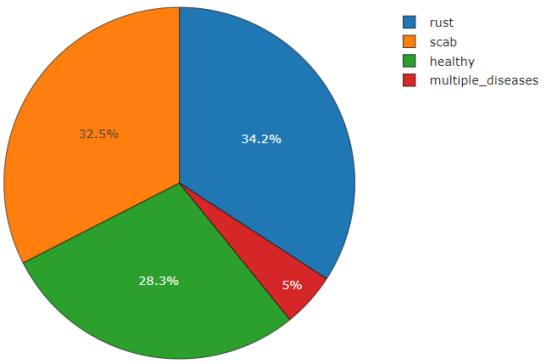

Figure 18. Distribution of Each Category

In the pie chart above, it can be seen that there are more unhealthy leaves in the dataset (rust, scab), namely 66.7% or as many as 1214 images, there are 28.3% healthy leaves or as many as 516 images, and only there are 5% of leaves that have complications (multiple diseases) or as many as 91 images.

4. CONCLUSION

With the influence of technology in agriculture, the influence of technology can be used to detect diseases on leaves to overcome the decrease in the number of harvests. With the Image Augmentation method used in this study, the existing dataset can have 6x more features. So the healthy category, which previously had 516 image features, now has 3096 feature images, the scab category, which previously had 592 feature images, now has 3552 feature images, and the rust category, which previously had 622 feature images, now has 3732 feature images. With a dataset with 3000 image features, the model to be made can have a higher accuracy value, and the model can be said to be robust/solid/sound, or the model to be made can carry out the classification process with a good level of accuracy.

BIBLIOGRAPHY

[1] Badan Pusat Statistika, "Produksi Buah-Buahan dan Sayuran Tahunan Menurut Jenis Tanaman di Kota Batu (Ton), 2017 dan 2018," 2018, Accessed: Oct. 11, 2022. [Online]. Available: https://batukota.bps.go.id/statictable/2019/12/03/549/produksi-buah-buahan-dan-sayuran-tahunan-menurut-jenis-tanaman-di-kota-batu-ton-2017-dan-2018.html

[2] J. Arun Pandian, G. Geetharamani, and B. Annette, "Data Augmentation on Plant Leaf Disease Image Dataset Using Image Manipulation and Deep Learning Techniques," in *2019 IEEE 9th International Conference on Advanced Computing (IACC)*, Dec. 2019, pp. 199–204. doi: 10.1109/IACC48062.2019.8971580.

[3] P. Enkvetchakul and O. Surinta, "Effective Data Augmentation and Training Techniques for Improving Deep Learning in Plant Leaf Disease Recognition," *Applied Science and Engineering Progress*, Jan. 2021, doi: 10.14416/j.asep.2021.01.003.

[4] U. Barman, D. Sahu, G. G. Barman, and J. Das, "Comparative Assessment of Deep Learning to Detect the Leaf Diseases of Potato based on Data Augmentation," in *2020 International Conference on Computational Performance Evaluation (ComPE)*, Jul. 2020, pp. 682–687. doi: 10.1109/ComPE49325.2020.9200015.

[5] J. Wongbongkotpaisan and S. Phumeechanya, "Plant Leaf Disease Classification using Local-Based Image Augmentation and Convolutional Neural Network," in *2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, May 2021, pp. 1023–1027. doi: 10.1109/ECTI-CON51831.2021.9454672.

[6] M. Li *et al.*, "FWDGAN-based data augmentation for tomato leaf disease identification," *Comput Electron Agric*, vol. 194, p. 106779, Mar. 2022, doi: 10.1016/j.compag.2022.106779.

[7] D. T. Husni *et al.*, "ANALISIS BIG DATA PENJUALAN VIDEO GAMES MENGUNAKAN EDA," *Jurnal Teknik Informasi dan Komputer (Tekinkom)*, vol. 5, no. 1, p. 43, Jun. 2022, doi: 10.37600/tekinkom.v5i1.517.

[8] Y. A. Zebua *et al.*, "PREDIKSI PENETAPAN TARIF PENERBANGAN MENGGUNAKAN AUTO-ML DENGAN ALGORITMA RANDOM FOREST," *Jurnal Teknik Informasi dan Komputer (Tekinkom)*, vol. 5, no. 1, p. 115, Jun. 2022, doi: 10.37600/tekinkom.v5i1.508.

[9] Fine-Grained Visual Categorization 7, "Plant Pathology 2020 - FGVC7," 2020.

https://www.kaggle.com/competitions/plant-pathology-2020-fgvc7 (accessed Oct. 11, 2022).

[10] M. Kalbasi and H. Nikmehr, "Noise-Robust, Reconfigurable Canny Edge Detection and its Hardware Realization," *IEEE Access*, vol. 8, pp. 39934–39945, 2020, doi: 10.1109/ACCESS.2020.2976860.

[11] K. Gaurav and U. Ghanekar, "Image steganography based on Canny edge detection, dilation operator and hybrid coding," *Journal of Information Security and Applications*, vol. 41, pp. 41–51, Aug. 2018, doi: 10.1016/j.jisa.2018.05.001.

[12] J. Nalepa *et al.*, "Data Augmentation via Image Registration," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 4250–4254. doi: 10.1109/ICIP.2019.8803423.

[13] I. Kostrikov, D. Yarats, and R. Fergus, "Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels," Apr. 2020.

[14] Z. Li *et al.*, "Maize leaf disease identification based on WG-MARNet," *PLoS One*, vol. 17, no. 4, p. e0267650, Apr. 2022, doi: 10.1371/journal.pone.0267650.

[15] S. Mathulaprangsan and K. Lanthong, "Cassava Leaf Disease Recognition Using Convolutional Neural Networks," in *2021 9th International Conference on Orange Technology (ICOT)*, Dec. 2021, pp. 1–5. doi: 10.1109/ICOT54518.2021.9680655.

[16] F. Wang, H. Liu, and J. Cheng, "Visualizing deep neural network by alternately image blurring and deblurring," *Neural Networks*, vol. 97, pp. 162–172, Jan. 2018, doi: 10.1016/j.neunet.2017.09.007.

[17] P. Bansal, R. Kumar, and S. Kumar, "Disease Detection in Apple Leaves Using Deep Convolutional Neural Network," *Agriculture*, vol. 11, no. 7, p. 617, Jun. 2021, doi: 10.3390/agriculture11070617.

[18] M. Radhi, A. Amalia, D. R. H. Sitompul, S. H. Sinurat, and E. Indra, "ANALISIS BIG DATA DENGAN METODE EXPLORATORY DATA ANALYSIS (EDA) DAN METODE VISUALISASI MENGGUNAKAN JUPYTER NOTEBOOK," *Jurnal Sistem Informasi dan Ilmu Komputer Prima(JUSIKOM PRIMA)*, vol. 4, no. 2, pp. 23–27, Feb. 2022, doi: 10.34012/jurnalsisteminformasidanilmukomputer.v4i2.2475.

[19] J. Tummers, C. Catal, H. Tobi, B. Tekinerdogan, and G. Leusink, "Coronaviruses and people with intellectual disability: an exploratory data analysis," *Journal of Intellectual Disability Research*, vol. 64, no. 7, pp. 475–481, Jul. 2020, doi: 10.1111/jir.12730.

[20] M. Cavallo, M. Dolakia, M. Havlena, K. Ocheltree, and M. Podlaseck, "Immersive Insights: A Hybrid Analytics System forCollaborative Exploratory Data Analysis," in *25th ACM Symposium on Virtual Reality Software and Technology*, Nov. 2019, pp. 1–12. doi: 10.1145/3359996.3364242.