

# Image-based Plant Disease Diagnosis with Unsupervised Anomaly Detection Based on Reconstructability of Colors

Ryoya Katafuchi<sup>1✉</sup>, \*Terumasa Tokunaga<sup>1✉</sup>,

**1** Department of Systems Design and Informatics, Kyushu Institute of Technology, Iizuka, Fukuoka, JAPAN



\* tokunaga@ces.kyutech.ac.jp

## Abstract

This paper proposes an unsupervised anomaly detection technique for image-based plant disease diagnosis. The construction of large and publicly available datasets containing labeled images of *healthy* and *diseased* crop plants led to growing interest in computer vision techniques for automatic plant disease diagnosis. Although supervised image classifiers based on deep learning can be a powerful tool for plant disease diagnosis, they require a huge amount of labeled data. The data mining technique of anomaly detection includes unsupervised approaches that do not require rare samples for training classifiers. We propose an unsupervised anomaly detection technique for image-based plant disease diagnosis that is based on the reconstructability of colors; a deep encoder-decoder network trained to reconstruct the colors of *healthy* plant images should fail to reconstruct colors of symptomatic regions. Our proposed method includes a new image-based framework for plant disease detection that utilizes a conditional adversarial network called pix2pix and a new anomaly score based on CIEDE2000 color difference. Experiments with PlantVillage dataset demonstrated the superiority of our proposed method compared to an existing anomaly detector at identifying diseased crop images in terms of accuracy, interpretability and computational efficiency.

## INTRODUCTION

Plant disease diagnosis is an important task for food safety and security. The PlantVillage project [1] was started to develop accurate image classifiers for plant disease diagnosis. This is a publicly available image dataset for developing automatic diagnostic techniques to identify plant diseases. It provides thousands of labeled images of healthy and diseased crop plants collected under controlled conditions. Such a large dataset has been used to establish deep learning challenges for developing an accurate image classifier for plant disease diagnosis.

In a comprehensive experiment using color crop images, AlexNet and GoogleNet achieved average accuracies of over 90% at identifying 26 diseases in 14 crop species [2]. Similarly, LeNet accurately classified diseased banana leaves under severe conditions [3]. Ferentinos (2018) [4] compared the performances of five convolutional neural network (CNN) models using leaf images obtained under both laboratory and real conditions. The VGG model achieved the best performance with a success rate of 99.53%. Another study using the PlantVillage dataset showed that deep learning provided outstanding performance compared with conventional machine learning techniques [5].

The results of the previous studies warrant further developments of image-based plant disease diagnosis techniques for more practical applications. Supervised image classifiers based on deep learning require a

huge amount of labeled data for training. Correcting samples with rare diseases often imposes a severe burden on human annotators, which can be a severe bottleneck for practical application. To address this issue, further studies are needed to develop an image-based diagnosis technique that is free of annotation costs for rare samples.

Anomaly detection is a data mining technique for identifying irregular or unusual patterns in datasets. This technique exhibits a wide range of applications, such as fraud detection for financial services, intrusion detection for networks, identification of disease markers for medical diagnosis, and failure detection for engineering systems. Typical approaches to anomaly detection are based on conventional machine learning. Simple clustering approaches are often used for unlabeled data [6] [7]. In cases where normal and anomalous labels are available, simple classification approaches such as support vector machines are used [8].

Recently, many anomaly detection techniques based on deep neural networks have been proposed in the fields of machine learning and computer vision. Deep anomaly detection can be categorized into three groups based on the type of machine learning: supervised approaches [9], unsupervised approaches [10], [11], [12], [13], [14], [15], and semi-supervised approaches [16], [17], [18]. [19] provides a comprehensive review of these approaches.

Generative adversarial networks (GANs) [20] demonstrated a great deal of success with image generation tasks [21] [22] [23] [24] [25] [26]. The excellent expressive power of GANs led to growing interest in utilizing them for real-world data analysis, including anomaly detection. GANs are typically applied to adversarial feature learning [27] of normal data and measuring anomaly scores for a given query image. AnoGAN [28] provides an unsupervised approach to detecting real-world anomalies, including the discovery of novel anomalies in medical imaging data. Recently, some extensions of AnoGAN have been proposed to overcome performance issues [29] [30] or improve the computational efficiency [31].

AnoGAN computes the anomaly score based on the reconstructability of normal samples. Because AnoGAN does not require anomalous data for training neural networks, it is applicable to diverse problems, including those within the natural sciences. However, this approach does not explicitly focus on colors in imaging data. In many real-world problems, color information is essential to discovering anomalies in datasets. For example, discoloration of leaves can be crucial information for detecting symptoms [32].

To the best of our knowledge, there have been no extensions of AnoGAN focused on detecting color anomalies. Moreover, AnoGAN exhibits two drawbacks for real-time applications: it requires a huge amount of normal data to learn a manifold of normal variability, and it requires an iterative procedure for calculating anomaly scores, which reduces computational efficiency.

In this paper, We propose a new anomaly detection method for detecting plant diseases at the image level and visualizing symptomatic regions at the pixel level. The proposed method uses a conditional adversarial network called pix2pix [26] for learning inverse mapping from converted grayscale images to original color images. The simplicity of this strategy means that the proposed method should work well even in cases where a large amount of normal data is unavailable, unlike AnoGAN. We applied the proposed method to the PlantVillage dataset to explore its utility. Also, we propose a simple, and easy-to-interpret anomaly score that is based on the CIEDE2000 color difference. Because the proposed method does not require any iterative procedure for calculating the anomaly score, the computational efficiency is expected to be sufficient for real-time disease detection.

## RELATED WORK

The present work was motivated by AnoGAN [28] and its extensions [29] [30] [31]. AnoGAN relies on the concept of reconstructing normal data from latent variables. This framework is applicable to diverse problems, but its effectiveness at detecting color anomalies has not been demonstrated in previous studies. Our proposed method relies on the reconstruct colors. Specifically, we hypothesized that a rich generative

model trained to reconstruct colors of normal data will fail to color anomalous regions in images. Unlike AnoGAN and its extensions, our focus was on detecting color anomalies, such as discolored parts on plants. Thus, the proposed method can be viewed as an extension of AnoGAN but in a different direction from previous studies.

## METHOD

### Outline of the Proposed Method

Fig. 1 shows a schematic of the proposed method. It consists of five steps for image-level detection and pixel-level visualization of plant diseases:

1. **Preparation:** Anomalies in color images are detected in terms of the reconstructability of colors at the pixel level with a conditional adversarial network. Consider a set of  $M$  pairs of color images and grayscale images. Let  $\mathbf{I}_c(x_i) \in \mathbb{R}_+^3$  ( $i = 1, 2, \dots, 256 \times 256$ ) be a pixel value representing a color at the pixel position  $x_i$ . Similarly, let  $I_g(x_i) \in \mathbb{R}_+$  ( $i = 1, 2, \dots, 256 \times 256$ ) be a pixel value representing the intensity at the pixel position  $x_i$ . For notational simplicity, color and grayscale images are expressed as  $\mathbf{I}_c$  and  $\mathbf{I}_g$ , respectively. Pairs of images,  $\mathbf{I}_c$  and  $\mathbf{I}_g$  are used to train the GAN so that the DCED network learns the inverse mapping  $G : \mathbf{I}_g \mapsto \mathbf{I}_c$ . These training images are selected only from normal data, while the test data include both normal and anomalous data. For evaluation purposes, we use a set of  $N$  color images with an array of binary image-wise ground-truth labels  $l_n \in \{1, -1\}$  ( $n = 1, 2, \dots, N$ ).
2. **Training:** The left half of Fig. 1 illustrates concurrent training on generator and discriminator networks. Pairs of images  $\mathbf{I}_g$  and  $\mathbf{I}_c$  are used as the input and output respectively of  $G$ . Pairs of  $\mathbf{I}_c$  and  $G(\mathbf{I}_g)$  are used as *fake* pairs to train the discriminator network  $D$ .
3. **Color reconstruction:** For the test data, the reconstructed color image  $\hat{\mathbf{I}}'_c$  is obtained from a query color image  $\mathbf{I}'_c$  by using the trained generator network.
4. **Calculation of color anomaly score:** The anomaly score is calculated for a given query color image  $\tilde{\mathbf{I}}_c$  based on color differences between the reconstructed color image  $G(\mathbf{I}'_c)$  and the original color image  $\mathbf{I}'_c$ . The color difference  $d_i$  is calculated from  $\mathbf{I}'_c(x_i)$  and  $\hat{\mathbf{I}}'_c(x_i)$  for all  $i$ . The anomaly score based on color difference is obtained simply by summing  $d_i$  for all  $i$ .
5. **Anomaly detection:** Finally, a query image is classified as normal or anomalous by a simple thresholding of the anomaly score.

### Color Reconstruction by Pix2pix

We used pix2pix [26] for color reconstruction. Pix2pix is a general framework for image-to-image translation based on a deep convolutional GANs (DCGANs). The generator network is U-Net, which is a DCED network with skip structures. Skip structures enable a DCED network to learn both global and local features efficiently. The discriminator is the convolutional PatchGAN, which only penalizes the structure at the patch scale.

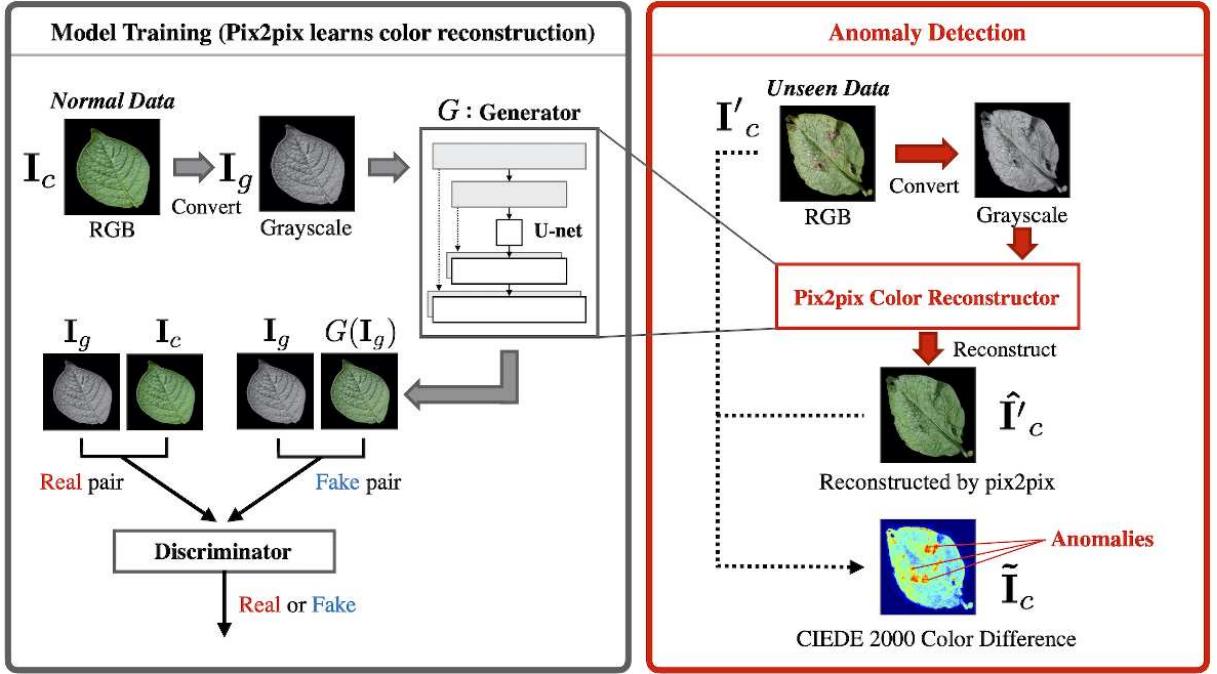


Fig 1. Outline of the proposed method.

Let the input and output variables for pix2pix be  $x$  and  $y$ , respectively. Now, the loss function for training pix2pix can be expressed as follows:

$$L_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log (1 - D(x, G(x, z)))] \quad (1)$$

where  $\mathbb{E}[\cdot]$  indicates the expected value and  $z$  is a random noise vector. During training, pix2pix seeks to minimise  $L_{cGAN}$  with respect to  $G$ , and at the same time maximise it with respect to  $D$  under  $L_1$ -regularization:

$$\begin{aligned} G^* &= \arg \min_G \max_D \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log (1 - D(x, G(x, z)))] \\ &+ \lambda \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1] \end{aligned} \quad (2)$$

### CIEDE2000 Color Anomaly Score

For a given query color image, the anomaly score is calculated for each pixel. We propose a new anomaly score based on CIEDE2000, which reflects differences as perceived by humans. This should align the provides anomaly score with visual inspection by human. The CIEDE2000 color difference  $d_i$  is calculated from  $I'_c(x_i)$  and  $\hat{I}'_c(x_i)$ . Then, the anomaly score is obtained simply by summing  $d_i$  for all  $i$ . We briefly describe the concept of CIEDE2000 in the Appendix. For more details, see [33].

## EXPERIMENT

### Dataset

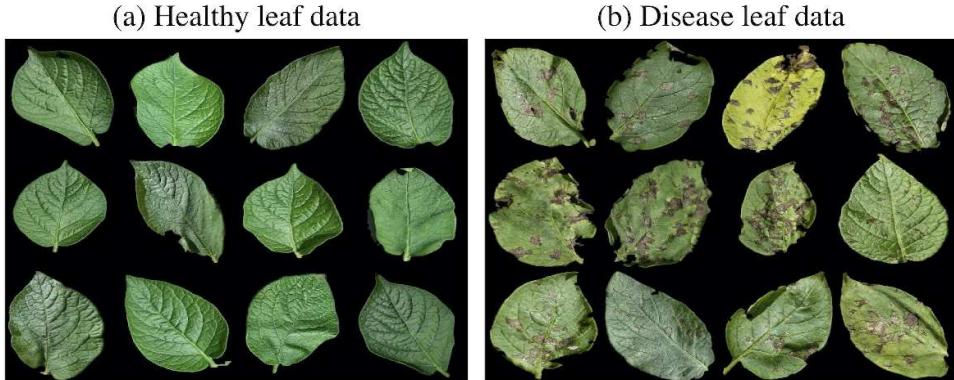
To evaluate the performance of the proposed method, we used a dataset that is publicly available through the PlantVillage project [1]. The dataset contains 54,306 images of healthy and diseased plants covering

**Table 1.** Number of images used in the experiment.

Training data	Healthy leaf images	76
Test data	Healthy leaf images	76
	Diseased leaf images	100

14 crops: *apples*, *blueberries*, *cherries*, *corns*, *grapes*, *oranges*, *peaches*, *bell peppers*, *potatoes*, *raspberries*, *soybeans*, *squash*, *strawberreis*, and *tomatoes*. Each image exhibits three different versions: *RGB color*, *grayscale*, and *segmented*.

Fig. 2 shows examples of segmented images in the PlantVillage dataset for (a) healthy and (b) diseased leaves. In the experiments, we used segmented potato plant images comprising 152 healthy leaves and 1,000 leaves with early blight, which is caused by the fungus *Alternaria solani*. Symptoms appear on older leaves as small brown spots. As the disease progresses, it spreads throughout the leaf surface and eventually makes it turn yellow and then wither.



**Fig 2.** Examples of potato plant segmented images in the PlantVillage dataset. (a) Healthy leaf Data, (b) Diseased leaf Data.

We divided these images into training and test sets for pix2pix, as given in Table 1. Half of the healthy-leaf images were allocated to the training set, and half were allocated to the test set. We randomly chose 100 diseased-leaf images for the test set. All images demonstrated a resolution of  $256 \times 256$  pixels.

## Experimental Setup

Here, we briefly describe the experimental setup for training the GANs. The optimization problem described in Eq. 2 was solved with the Adam optimizer at a learning rate of 0.00015. The momentums were set to  $\beta_1 = 0.5$ , and  $\beta_2 = 0.999$ . The discriminator has a PatchGAN architecture with a patch size of  $64 \times 64$ . Training was terminated after 150 epochs. The hyperparameter for  $L_1$  regularization was set to  $\lambda = 10$ .

For comparison, we use AnoGAN and the conventional unsupervised anomaly detection methods: Convolutional Autoencoders (CAEs) with L2 and SSIM [36], One-Class SVM (OC-SVM) [35] as baseline methods for evaluating the performance of our proposed method.

AnoGAN was also trained with the Adam optimizer at a learning rate was 0.0001. Momentums were set to be the same as those for pix2pix. AnoGAN training was terminated after 2,000 epochs. The latent

variable dimension in AnoGAN was set to 30. To calculate the anomaly scores, the weight coefficients for the residual loss and discrimination score were set to 0.9 and 0.1, respectively. Also, we calculated simple color histogram similarity to measure the color differences aiming at highlighting a characteristic of CIEDE2000.

We used Pytorch (version 1.3.0) code for pix2pix which is available at <https://github.com/phillipi/pix2pix>. We modified the Keras (version 2.2.4) code for AnoGAN which is available at <https://github.com/tkwoo/anogan-keras>. All computations were performed on a GeForce RTX 2028 Ti GPU based on a system running Python 3.6.9 and CUDA 10.0.130.

## Color Differences at the Pixel Level

Fig. 3 shows example results for the pixel-level visualization of test data. The top row (Healthy) shows the results for healthy potato leaves. The middle and lower rows (Disease (1) and Disease (2), respectively)) show the results for two diseased leaf examples. Fig. 3(a) shows the original color images, and Fig. 3(b) shows the grayscale conversion. Fig. 3(c) shows the color images reconstructed by pix2pix. Fig. 3(d) shows heat maps visualizing the CIEDE2000 color difference between the original and reconstructed color images. Warm colors indicate a large color difference.

As shown in Figs. 3(c), and (d), the healthy-leaf images were successfully reconstructed. However, the symptomatic brown spots and yellow discoloration in the diseased-leaf images were not reconstructed. These results aligned with our expectations ;because we trained pix2pix only with healthy-leaf images, the generator network could not reconstruct colors in symptomatic regions. Consequently, symptomatic regions exhibited large CIEDE2000 color differences. In contrast, healthy leaves exhibited no significant color differences.

For comparison, Fig. 3(e) shows the images reconstructed by AnoGAN. The generator network was trained with healthy-leaf images just like pix2pix. Fig. 3(f) presents heat maps visualizing the residuals for each pixel between the original and reconstructed color images using the same format as in Fig. 3(d).

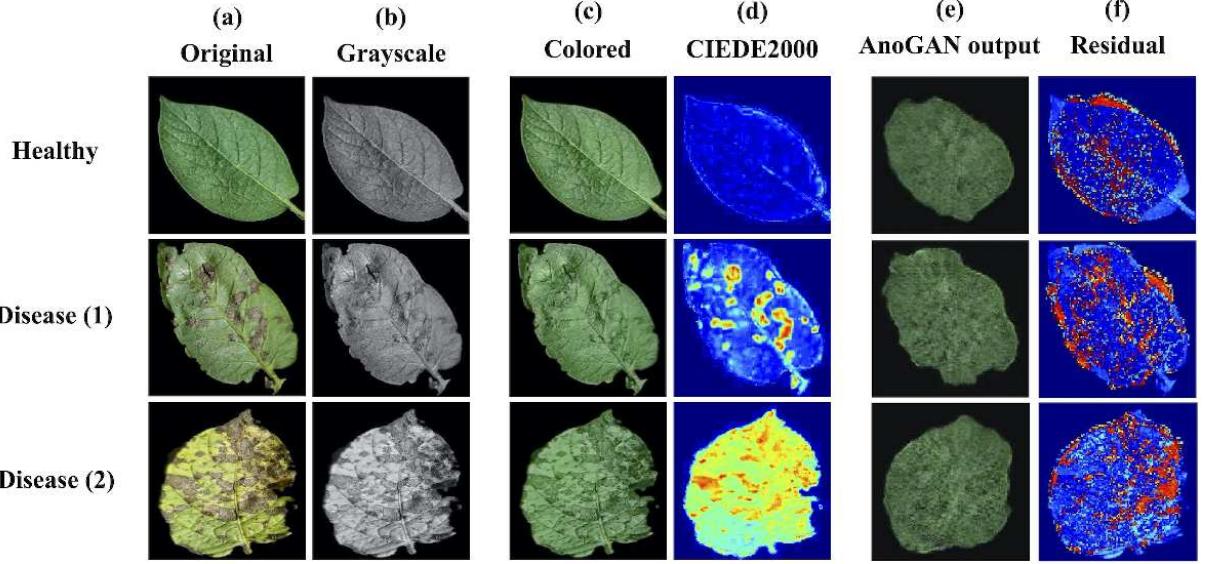
The leaves were clearly reconstructed, incompletely, which strongly affected the pixel-level residuals. In particular, artificially highlighted regions can be observed around the edges of leaves in both the healthy and diseased cases. In contrast, the Disease (2) images indicate that most symptomatic regions, with yellow discoloration were not highlighted in the heat map. These results suggest the limitations of AnoGAN for visualizing symptomatic regions of plant leaves at the pixel level.

The incomplete reconstruction was most likely caused by a lack of training data. Accordingly, the reconstruction would be improved by adding healthy leaf images to the training data. However, this should cause the generator network, to fail to reconstruct diseased-leaf images, which may generate substantive artifacts. Thus, our proposed method provides a more efficient pixel-level visualization of anomalies in images in comparison with AnoGAN. In addition, our proposed method works well even though only 76 healthy-leaf images were used for training.

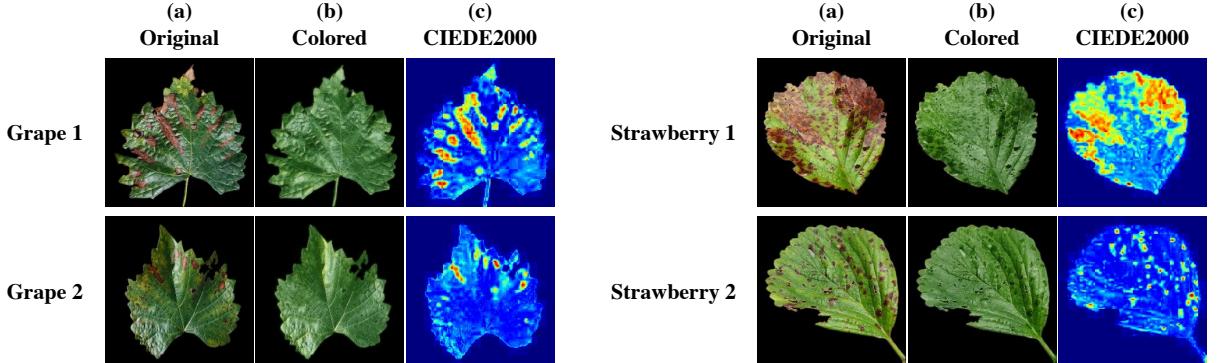
Fig. 4 shows additional examples of pixel-level visualization of grape and strawberry leaf images with the proposed method. Similar to the previous results with potatoes, the symptomatic regions on these leaves were successfully highlighted. Thus, the proposed method works well at detecting symptomatic regions of various plants.

## Performance Evaluation of Image-level Disease Detection

Fig. 5 presents histograms for three anomaly scores: (a) the CIEDE2000 anomaly score (i.e., proposed method), (b) color histogram similarity, and (c) AnoGAN anomaly score. The anomaly scores for healthy and disease-leaf images are indicated in red and blue, respectively. The results indicate that the CIEDE2000 anomaly score is more intuitive for distinguishing healthy and diseased samples at the image level.



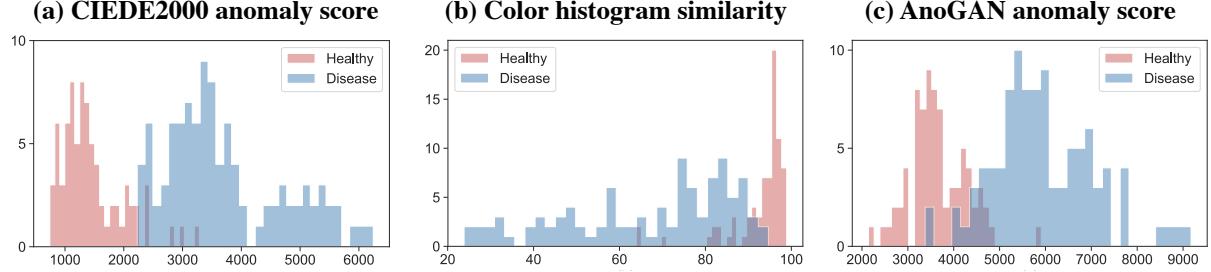
**Fig 3.** Examples of pixel-level disease visualization of plants: (a) original color images, (b) grayscale conversion, and (c) color images reconstructed by pix2pix. (d) CIEDE2000 color differences between the original color images and reconstructed color images. (e) Images reconstructed by AnoGAN. (f) Residuals between the original color images and images reconstructed by AnoGAN.



**Fig 4.** Other examples of pixel-level disease visualization for grapes (first and second rows) and strawberries (third and fourth rows): (a) original color images, (b) reconstructed color images, and (c) CIEDE2000 color differences.

To demonstrate the utility of the CIEDE2000 anomaly score in a more objective manner, Fig. 6, shows receiver operating characteristic (ROC) curves for image-level disease detection with the three types of anomaly scores: CIEDE2000 (red), color histogram similarity (blue), and ANoGAN (green). The corresponding area under the ROC curves (AUC) is specified in parentheses in the figure legend. The shapes of the ROC curves indicate that the CIEDE2000 anomaly score demonstrates several useful properties: a high true positive rate and low false positive rate superior to those of the other two anomaly scores.

Table 2 presents the statistical performance of the proposed method at image-level disease detection in terms of the precision, recall, and  $F_1$ -score. These indices were calculated the top-100 test images sorted in decreasing order with respect to anomaly score. The best performance for each index is bolded. The



**Fig 5.** Histograms of three anomaly scores.

**Table 2.** Comparison of image-level disease detection performances with the PlantVillage dataset.

Method	Precision	Recall	$F_1$ -Score	AUC
CIEDE2000 Anomaly Score	<b>0.94</b>	<b>0.94</b>	<b>0.94</b>	<b>0.99</b>
color Histogram Similarity	0.86	0.86	0.86	0.94
AnoGAN Anomaly Score	0.92	0.92	0.92	0.97
AutoEncoder L2 Loss	0.67	0.68	0.69	0.67
AutoEncoder SSIM Loss	0.89	0.90	0.90	0.94
OneClass SVM	0.91	0.92	0.92	0.95

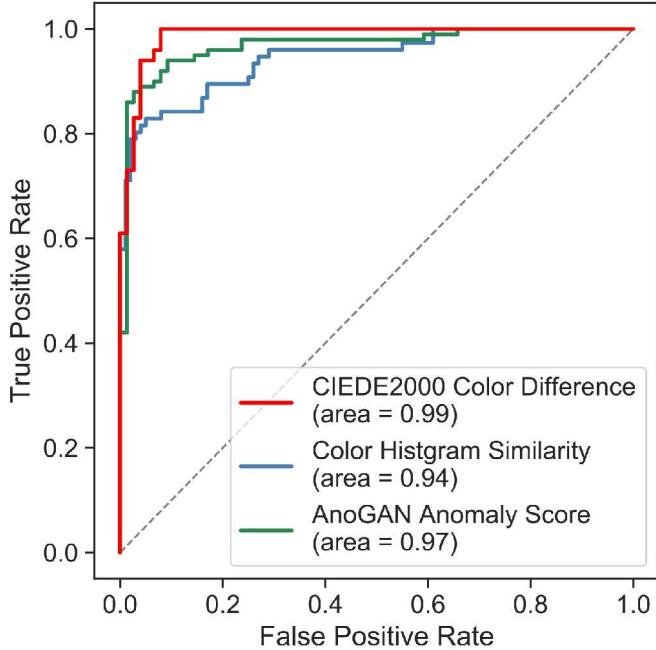
CIEDE2000 anomaly score demonstrated a superior performance compared to the two baseline anomaly scores for all indices.

### Computational Efficiency for the Anomaly Score

During detection, AnoGAN needs to determine the latent space location for a given query image based on iterative backpropagation, which leads to the anomaly score [28] [31]. This process reduces the computational efficiency of AnoGAN. In contrast, our proposed method does not require an iterative procedure during detection. Table 3 presents the mean computation times in milliseconds with standard deviations for calculating the CIEDE2000 anomaly score and AnoGAN anomaly score of a given query image. The results suggest that our approach offers superior computational efficiency compared to AnoGAN, which is clearly an important consideration for the practical application of real-time disease detection in plantations.

**Table 3.** Computation times for anomaly scores.

	<i>Healthy</i>	<i>Diseased</i>
Our method	$61.24 \pm 17.21$ ms	$62.09 \pm 18.33$ ms
AnoGAN	$3884 \pm 1138$ ms	$4574 \pm 1538$ ms



**Fig 6.** ROC curves based on three anomaly scores.

## CONCLUSIONS

We proposed a novel method for detecting plant diseases from images that relies on color reconstructability. Similar to AnoGAN, the proposed method detects anomalies in test data based on unsupervised training of a generator and discriminator by normal data. Unlike AnoGAN, however, the proposed method predominantly focuses on color anomalies in images.

We compared the performance of the proposed method with baseline methods including AnoGAN in terms of accuracy, interpretability, and computational efficiency. Experiments with the PlantVillage dataset showed that the proposed method performed better than AnoGAN at image-level anomaly detection. Because the CIEDE2000 anomaly score is simple and aligns with human visual inspection, it can be intuitively visualized as a heat map at the pixel level. In representative examples, symptomatic regions on leaves such as brown spots and yellow discoloration were efficiently highlighted. No serious artifacts were observed in either healthy- or diseased-leaf images, in contrast to the residual maps based on AnoGAN.

Because the proposed method does not require any iterative computation for calculating anomaly scores, the mean computation time is significantly less than that of AnoGAN. The computational efficiency means that the proposed method could be practical for application to real-time image-based plant disease detection. Future studies are warranted to explore the practicality of automatic diagnosis systems for detecting plant diseases on a global scale based on the idea of color reconstructability.

## ACKNOWLEDGMENTS

This work was supported by JST, PRESTO Grant Number JPMJPR1875, Japan. The authors would like to thank Enago ([www.enago.jp](http://www.enago.jp)) for the English language review.

## References

1. Hughes DP, Salathe M. An open access repository of images on plant health to enable the development of mobile disease diagnostics; 2016. Available from: [arXiv:1511.08060](https://arxiv.org/abs/1511.08060).
2. Mohanty SP, Hughes DP, Salathe M. Using deep learning for Image-Based plant disease detection. *Frontiers in Plant Science*. 2016;22.
3. Amara J, Bouaziz B, Algergawy A. A Deep Learning-based Approach for Banana Leaf Diseases Classification. In: *Lecture Notes in Informatics* 2017; 2017.
4. Ferentinos KP. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*. 2018;145:311–318.
5. D. Radovanović, S. Đukanović Image-Based Plant Disease Detection: A Comparison of Deep Learning and Classical Machine Learning Algorithms. In: 24th International Conference on Information Technology (IT); 2020.
6. Xiong L, Poczos B, Schneider JG. Group anomaly detection using flexible genre models. In: *NIPS2011*; 2011. p. 1071–1079.
7. Zimek A, Schubert E, Kriegel H. A survey on unsupervised outlier detection in high dimensional numerical data. *Statistical Analysis and Data Mining*. 2012;5(5).
8. Chen Y, Zhou X, Huang TS. One-Class SVM for leaning in image retrieval. In: *Proceedings IEEE International Conference on Image Processing* 2001; 2001.
9. Chalapathy R, Menon AK. Robust, deep and inductive anomaly detection. In: *Proceedings of ECML PKDD* 2017; 2017.
10. Patterson J, Gibson A. *Deep Learning: A Practitioner’s Approach*. O’Reilly Media, Inc.; 2017.
11. Tuor A, Kaplan S, Hutchinson B, Nichols N, Robinson S. Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. In: *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*; 2017.
12. Sutskever I, Hinton G, Taylor G. The recurrent temporal restricted Boltzmann machine. In: *Advances in Neural Information Processing Systems 21 (NIPS 2008)*; 2008. p. 1601–1608.
13. Vincent P, Larochelle H, Bengio Y, Manzagol PA. Extracting and composing robust features with denoising autoencoders. In: *Proceedings of the 25th international conference on Machine learning*; 2008. p. 1096–1103.
14. Rodriguez P, Wiles J, Elman JL. A recurrent neural network that learns to count. *Connection Science*. 1999;11(1):5–40.
15. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C. Neural architectures for named entity recognition. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics; 2016. p. 260–270. Available from: <https://www.aclweb.org/anthology/N16-1030>.
16. Edmunds R, Feinstein E. Deep semi-supervised embeddings for dynamic targeted anomaly detection; 2017.

17. Racah E, Beckham C, Maharaj T, Kahou SE, Prabhat, Pal C. ExtremeWeather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. In: Advances in Neural Information Processing Systems; 2017. p. 3402–3413.
18. Perera P, Patel VM. Learning Deep Features for One-Class Classification. IEEE Transactions on Image Processing. 2019;28(11):5450–5463.
19. Chalapathy R. Generative Adversarial Networks; 2019. Available from: [arXiv:1901.03407](https://arxiv.org/abs/1901.03407).
20. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al.. Deep learning for anomaly detection: A Survey; 2014. Available from: [arXiv:1406.2661](https://arxiv.org/abs/1406.2661).
21. Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. In: ICLR2016; 2016.
22. Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In: NIPS2016; 2016.
23. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X. Improved techniques for training GANs; 2016. Available from: [arXiv:1606.03498](https://arxiv.org/abs/1606.03498).
24. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A. Improved training of Wasserstein GANs. In: NIPS2017; 2017.
25. Mao X, Li Q, Xie H, Lau RYK, Wang Z, Smolley SP. Least squares Generative Adversarial Networks. In: 2017 IEEE International Conference on Computer Vision (ICCV); 2017.
26. Isola P, Zhu JY, Zhou T, Efros AA. Image-to-Image translation with conditional adversarial networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017.
27. Donahue J, Krakenbuhl P, Darrell T. Adversarial feature learning. In: The International Conference on Learning Representations (ICLR); 2017.
28. Schlegl T, Seeböck P, Waldstein SM, Schmidt-Erfurth U, Langs G. Unsupervised anomaly detection with Generative Adversarial Networks to guide marker discovery. In: International Conference on Information Processing in Medical Imaging; 2017. p. 146–157.
29. Zenati H, Foo CS, Lecouat B, Manek G, Chandrasekhar VR. Efficient GAN-Based anomaly detection. In: 6th International Conference on Learning Representations (ICLR2018); 2018.
30. Akcay S, Atapour-Abarghouei A, Breckon TP. GANomaly: semi-supervised anomaly detection via adversarial training; 2018. Available from: [arXiv:1805.06725](https://arxiv.org/abs/1805.06725).
31. Schlegl T, Seeböck P, Waldstein SM, Langs G, Schmidt-Erfurth U. f-AnoGAN: Fast unsupervised anomaly detection with generativeadversarial networks. Medical Image Analysis. 2019;54:30–44.
32. Riley MB, Williamson MR, Maloy O. Plant Disease Diagnosis; 2002. Available from: <https://www.apsnet.org/edcenter/disimpactmngmnt/casestudies/Pages/PlantDiseaseDiagnosis.aspx>.
33. Sharma G, Wu W, Dalal EN. The CIEDE2000 color-difference formula: implementation notes,supplementary test data, and mathematical observations. Color Research and Application. 2005;30(1).

34. Paul Bergmann, Michael Fauser, David Sattlegger, Carsten Steger MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
35. Scholkopf Bernhard, Smola Alexander J, Bach Francis and others Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press. 2002
36. Bergmann Paul, Lowe Sindy, Fauser Michael, Sattlegger David, Steger Carsten Improving unsupervised defect segmentation by applying structural similarity to autoencoders. arXiv preprint arXiv:1807.02011, 2018

## APPENDIX

### CIEDE2000 Color Difference

The CIEDE2000 color difference is calculated by using the color space  $L^*a^*b^*$ , which is suitable for expressing colors based on human perception. The color difference is based on three parameters: the lightness difference ( $\Delta L'$ ), chroma difference ( $\Delta C'$ ) and hue difference ( $\Delta H'$ ). These are weighted by the functions ( $S_L, S_C, S_H$ ), parametric weighting factors ( $k_L, k_C, k_H$ ), and rotation term ( $R_T$ ). All parametric weighting factors were set to  $k_L = k_C = k_H = 1$ .

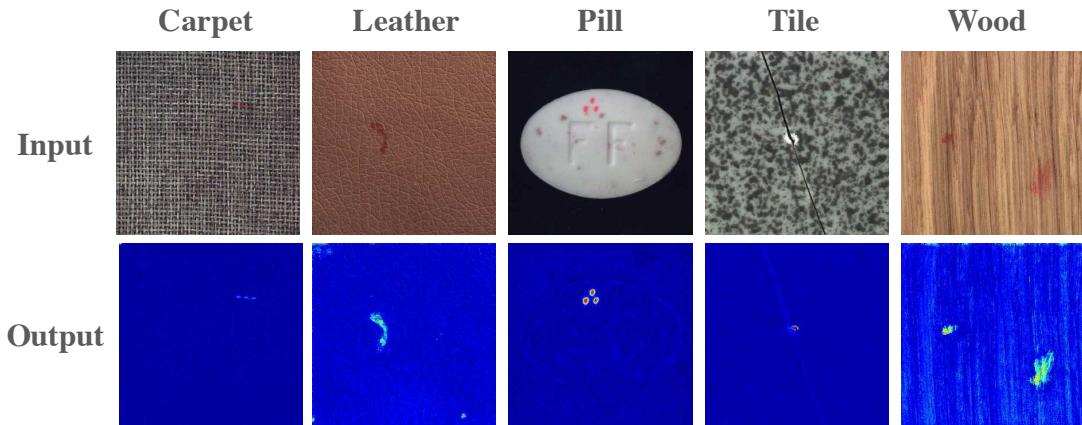
The CIEDE2000 color difference between two points in the  $L^*a^*b^*$  color space  $(L_1^*, a_1^*, b_1^*)$  and  $(L_2^*, a_2^*, b_2^*)$  is calculated as follows:

$$\Delta E_{00}(L_1^*, a_1^*, b_1^*, L_2^*, a_2^*, b_2^*) = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'}{k_C S_C}\right)^2 + \left(\frac{\Delta H'}{k_H S_H}\right)^2 + \left(R_T\left(\frac{\Delta C'}{k_C S_C}\right)\left(\frac{\Delta H'}{k_H S_H}\right)\right)} \quad (3)$$

The Python code used for implementing this formula is available at <https://github.com/scikit-image>. Further details and a derivation of the CIEDE2000 color difference equation are provided by [33].

### Application for MVTec Anomaly Detection Dataset

We validated the effectiveness of our anomaly detection method for industrial problems by using the MVTec Anomaly Detection Dataset (MVTec AD) [34]. This is a comprehensive dataset for benchmarking anomaly detection methods with a focus on industrial applications. It contains 5,354 high resolution color images of different object and texture categories with annotated pixel-level ground-truth regions for all anomalies. Fig. 7 shows examples of pixel-level anomaly visualizations with the proposed method. The top row shows the input images. We selected five categories from MVTec AD whose anomalies affect the color. The proposed method effectively highlighted anomalies on industrial products.



**Fig 7.** Examples of pixel-level anomaly visualization with MVTec AD.