International Journal on Advanced Science Engineering Information Technology

Vol.14 (2024) No. 5
ISSN: 2088-5334

Fine-tuning GAN Models with Unpaired Aerial Images for RGB to NDVI Translation in Vegetation Index Estimation

Hendri Darmawan [a], Mike Yuliana [a,*], Moch. Zen Samsono Hadi [a], Rahardhita Widyatra Sudibyo [a]

[a] *Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya (PENS), Surabaya, Indonesia*
Corresponding author: [*]mieke@pens.ac.id

***Abstract*—Calculating the Normalized Difference Vegetation Index (NDVI) requires expensive multispectral cameras, posing challenges such as high costs and the need for technical expertise. This research aims to develop a method to transform RGB images obtained from Unmanned Aerial Vehicle (UAV) into NDVI images. Our proposed method leverages the CycleGAN model, an unsupervised image-to-image translation framework, to learn the intricate mapping between RGB values and NDVI. The model was trained on unpaired datasets of RGB and NDVI images, sourced from paddy fields located in Gresik and Yogyakarta, Indonesia. This process successfully encapsulated the complex correlation between these two modalities. Various training strategies were systematically investigated, including weight initialization schemes, fine-tuning procedures, and learning rate policies, to optimize the model's performance. The fine-tuned CycleGAN demonstrated superior performance in creating synthetic NDVI images from unpaired dataset, surpassing other methods in terms of fidelity, quality, and structural coherence. The results were impressive, with a Normalized Root Mean Square Error (NRMSE) of 0.327, a Peak Signal-to-Noise Ratio (PSNR) of 16.330, an Oriented FAST and Rotated BRIEF (ORB) score of 0.859, and a Structural Similarity Index (SSIM) of 0.757. The best performing CycleGAN model was then deployed on a low-spec microcomputer device, specifically the Raspberry Pi 4B with an average computation time of 21.0077 seconds. Raspberry Pi 4B was chosen for its lightweight, compact dimensions, and compatibility with efficient battery power connections, making it suitable for drone deployment.**

***Keywords*—Fine-tuning; CycleGAN; aerial imagery; NDVI; vegetation index; unmanned aerial vehicle.**

Manuscript received 29 Apr. 2024; revised 1 Jun. 2024; accepted 5 Jul. 2024. Date of publication 31 Oct. 2024.

I. INTRODUCTION

The Normalized Difference Vegetation Index (NDVI) is an essential remote sensing tool for monitoring plant growth and chlorophyll content [1]. Traditionally, NDVI calculation requires expensive multispectral or hyperspectral cameras, posing challenges of high costs and technical expertise demands. Using RGB vegetation indices from UAV images offers a cost-effective and efficient way to monitor rice crops, serving as a viable alternative when multispectral images are not available [2]. Current methods for plant health monitoring using UAV and RGB images employ synthetic NDVI techniques, such as ExG, ExR, ExGr, NGRDI, IKAW, MGRVI, and GLI [3]. These methods compute various combinations of RGB channels to estimate vegetation indices. Upendar et al. [4] proposed identifying green vegetation using visible spectral color indices, such as the excess green index (ExG), excess red index (ExR), and excess green minus excess red index (ExGR). The ExGR index demonstrated the highest correct classification rate (93.03%) for distinguishing plant material from non-plant material at low illumination intensity. However, these indices can be affected by shadow texture when extracting vegetation, making it challenging to achieve high accuracy. Ribeiro et al. [5] and Lim et al. [6] demonstrated the effectiveness of the GLI vegetation index, derived from UAV imagery, in precision agriculture and vegetation detection. At the same time, Ribeiro et al. [5] found the NGRDI and GLI indices helpful in monitoring the growth rate and determining the harvest point of different green lettuce. Conversely, Lim et al. [6] confirmed GLI's superiority in distinguishing between vegetation and non-vegetation areas using RGB aerial data.

Jiang et al. [7] conducted study using digital cameras attached to UAV for monitoring crop growth in open-air fields. The focus was on the underexplored area of using these cameras to estimate Leaf Nitrogen Concentration (LNC) and monitor the status of crops. The study also evaluated the performance of vegetation indices derived from these cameras.

The results showed that IKAW was the most effective during the booting and heading stages of crop growth. Estrada et al. [8] implemented and evaluated ten vegetation indices to identify coffee leaf rust using RGB images. The Modified Green-Red Vegetation Index (MGRVI) showed the highest performance with 81% effectiveness. However, this study used a smartphone camera in Ecuadorian coffee fields, and the researchers acknowledged the need for further testing using UAV-acquired RGB and multispectral images, as well as the potential benefits of machine learning techniques.

In this research, we propose a novel approach to acquire wide-area imagery of rice fields using cost-effective UAV and transform the captured RGB images into NDVI representations without relying on specialized Near-Infrared (NIR) sensors. The proposed method leverages the CycleGAN model, an unsupervised image-to-image translation framework, to learn the complex mapping between RGB and NDVI domains. CycleGAN eliminates the need for paired training data, making it suitable for scenarios where paired data acquisition is challenging. The model captures the intricate relationship between these modalities by training on unpaired datasets of RGB and NDVI images from paddy fields. CycleGAN uses two generators and two discriminators. Each generator translates images between domains, while each discriminator distinguishes between synthesized and real images. However, the training of these models is inherently challenging due to the high-dimensional parameter space and the adversarial nature of the optimization process. Consequently, the choice of weight initialization scheme and learning rate policy can significantly impact the training dynamics, convergence behavior, and ultimately, the model's performance [9].

This research methodically examines the impact of two strategies for weight initialization: one using a transfer learning approach and the other using standard weight initialization. Transfer learning proves to be highly advantageous during the training of neural networks. It allows for the application of knowledge from a previous task to a new task, resulting in reduced training time, improved performance, and a lesser need for extensive training data. Additionally, we explored two policies for scheduling learning rates and five varying initial learning rates. The learning rate is considered one of the most crucial hyperparameters in training Convolutional Neural Networks (CNNs) [10]. Adjusting hyperparameters in machine learning algorithms is essential because it can impact the model's performance and tuning them can enhance the model's accuracy [11].

Given the lack of paired datasets, it's unfeasible to employ a supervised generative model like pix2pix for the conversion of RGB to NDVI [12]. However, our innovative proposed research involves the use of unsupervised methodologies, particularly the CycleGAN model, to convert RGB images into NDVI images, eliminating the need for paired datasets. This strategy has the potential to surpass the constraints of current synthetic NDVI techniques by harnessing the capabilities of the generative model to comprehend intricate correlations between RGB values and NDVI. Furthermore, once the model is developed, it can be utilized to produce NDVI representations from RGB images, eliminating the need for expensive multispectral cameras. Table 1 illustrates a comparison between the proposed research and previous studies.

TABLE I
COMPARATIVE STUDY ACROSS EXISTING APPROACHES

Method	Spectral bands used			Embedded deployment
	Red	Green	Blue	
ExG	✓	✓	✓	–
ExR	✓	✓	–	–
ExGr	✓	✓	✓	–
NGRDI	✓	✓	–	–
GLI	✓	✓	✓	–
IKAW	✓	–	✓	–
MGRVI	✓	✓	–	–
Ours	✓	✓	✓	✓

The following parts of this paper are arranged in the following manner. Section 2 presents a detailed overview of the materials and the proposed methodology. Section 3 outlines the experimental setup, results, and the interpretation. Finally, Section 4 concludes with a summary of the key findings from the study.

II. MATERIALS AND METHODS

This section provides a general description of the research framework, which comprises several key components: (1) data acquisition; (2) image preprocessing; (3) modeling with CycleGAN; (4) implementation of a synthetic NDVI approach that is not based on the generative model; (5) testing of image similarity metrics.

A. Data Acquisition

In this study, we collected image data using two types of UAVs: DJI Phantom 4 and DJI Mavic Air 2S, both equipped with RGB cameras. Additionally, we used a Mapir Survey 3 camera with RGN channels to capture NDVI images, which served as the target data. The Mapir Survey 3 camera captures NIR 850nm, red 660nm, and green 550nm light. We employed two types of RGB UAV to increase the variation of image characteristics, as they have different RGB camera specifications. We collected the data from paddy fields located in Gresik and Yogyakarta, Indonesia. During data acquisition, as illustrated in Fig. 1, we mounted the Mapir Survey 3 camera on the UAV and simultaneously captured RGB images using the UAV's built-in camera.

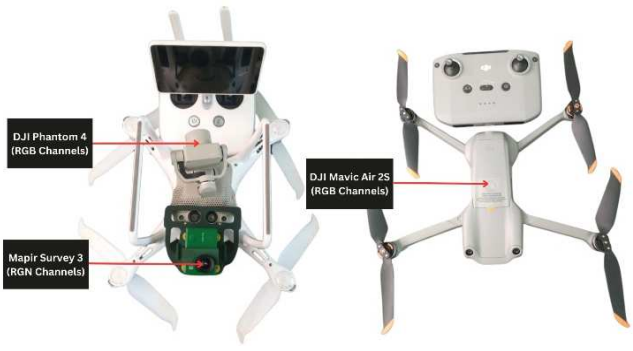


Fig. 1 Data acquisition setup. A DJI Phantom 2 flies over the field with a Mapir Survey camera to collect paired RGB-RGN image. The DJI Mavic Air 2S flies too to collect unpaired RGB image

The UAV was flown at an altitude of 25 to 30 meters. The collected data was in video format, necessitating frame extraction. Table 2 presents the distribution of the selected data. The test data consisted of paired data used to evaluate the NDVI generated by CycleGAN. Using our data test, this research also compared synthetic NDVI from several approaches to real NDVI

TABLE II
DATA DISTRIBUTION

Train		Test	
RGB	RGN	RGB	RGN
288	288	73	73

B. Image Preprocessing

After the frame extraction is performed, the next step is to convert the RGB image into the NDVI using Eq. (1) [3].

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{1}$$

Fig. 2 shows sample image of the Red and NIR channel and transformed NDVI. The subsequent image preprocessing stage involves uniforming the resolution between RGB and RGN images through square cropping. The resolution is unified to 416x416 pixels to decrease computational time during the training of images with CycleGAN. Additionally, contrast stretching techniques are employed to enhance the contrast quality of images prior to modeling [13].

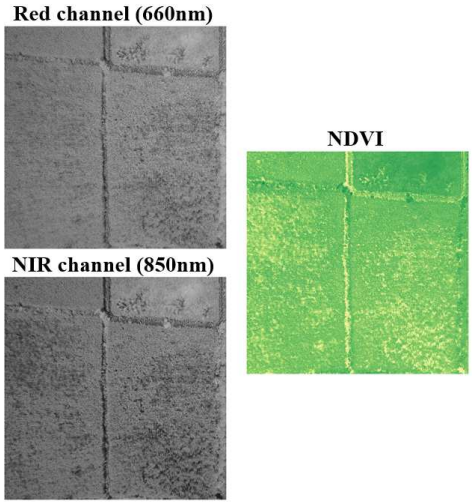


Fig. 2 Red channel (top-left) and NIR channel (bottom-left) with NDVI on the right

C. CycleGAN

CycleGAN is an approach for image-to-image translation tasks that learns a mapping between two domains (e.g., X: RGB and Y: NDVI) in the absence of paired training data [14]. The CycleGAN's architecture in Fig. 3 consists of two generator networks (G and F) and two discriminator networks (D_Y and D_X) along with feature extractors (E_Y and E_X).

For the mapping function G: X → Y and its discriminator DY, we define the objective expressed by Eq. (2) [15]. The identical adversarial loss is applied to the mapping function F : Y → X and its discriminator D_X i.e. $\mathcal{L}_{\text{GAN}}(F, D_X)$. The concept of adversarial training suggests that mappings G and F can be learned to produce outputs distributed identically to the target domains Y and X.

$$\mathcal{L}_{adv}(G, D_Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x)))] \tag{2}$$

For the mapping function G: X → Y and its discriminator DY, we define the objective expressed by Eq. (2) [15]. The identical adversarial loss is applied to the mapping function F : Y → X and its discriminator D_X i.e. $\mathcal{L}_{\text{GAN}}(F, D_X)$. The concept of adversarial training suggests that mappings G and F can be learned to produce outputs distributed identically to the target domains Y and X.

The image translation cycle ensures that each image x from domain X can be reconstructed back to its original form: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$. Similarly, for images y from domain Y, both G and F should satisfy backward cycle consistency: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$. The cycle consistency loss ensures bijective mapping by minimizing the reconstruction error after a cycle which is denoted by Eq. (3) [14].

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}\left[\left\|F(G(x)) - \mathrm{x}\right\|_1\right] + \mathbb{E}_{y \sim p_{\text{data}}(y)}\left[\left\|G(F(y)) - y\right\|_1\right] \tag{3}$$

Additionally, the identity loss ensures that the generators maintain color composition when translating to the other domain, as expressed by Eq. (4) [14].

$$\mathcal{L}_{\text{idt}}(F, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(x) - \mathrm{x}\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(y) - y\|_1] \tag{4}$$

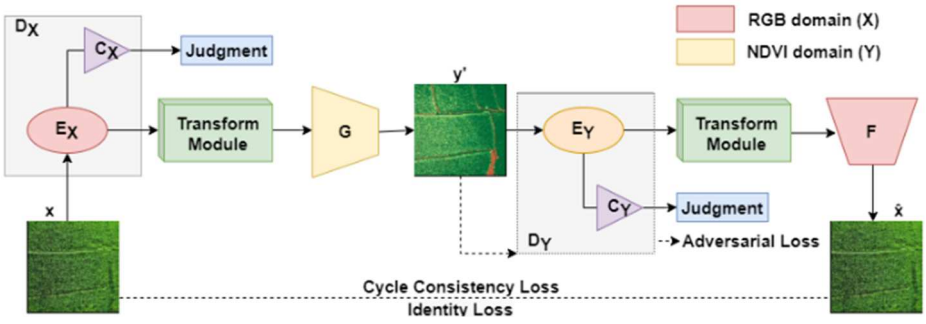


Fig. 3 CycleGAN: Forward translation

We used generators employing ResNet architecture with 9 blocks (~11.38M parameters), while the discriminators utilized a basic architecture (~2.765M parameters). The overall objective loss function is a weighted sum of adversarial, cycle consistency, and identity losses represented by Eq. (5). We utilize λ as a weighting factor to control the significance of these losses. Specifically, we set λ_1 to 10 and λ_2 to 0.5.

$$\mathcal{L}_{\text{GAN}}(G,F,D_Y,D_X) = \mathcal{L}_{adv}(G,D_Y) + \mathcal{L}_{adv}(F,D_X) + \lambda_1\mathcal{L}_{\text{cyc}}(G,F) + \lambda_1\lambda_2\mathcal{L}_{\text{idt}}(F,G) \quad (5)$$

D. Modeling Schemes

We fine-tuned the CycleGAN model for RGB to NDVI conversion using pretrained weights from sat2map and map2sat models [16]. This decision was based on their shared remote sensing origin, ability to process environmental features, and the need for abstract transformations. We also compared this with a model trained using normal weight initialization. Recognizing that performance depends on factors like fine-tuning and data characteristics, we conducted experiments involving hyperparameter and learning rate adjustments to understand the impact of fine-tuning on performance [17]. Algorithm 1, outlined in Table 3, provides an overview of the CycleGAN fine-tuning process and the subsequent testing phase.

TABLE III
ALGORITHM 1: CYCLEGAN FINE-TUNING PROCESS

Input:
- Training set RGB $\mathcal{D}_x = \{(x_i^t)\}_{i=1}^{n_x}$,
- Training set NDVI $\mathcal{D}_y = \{(y_i^t)\}_{i=1}^{n_y}$,
- Test set paired RGB and NDVI $\mathcal{D}_h = \{(x_i^h, y_i^h)\}_{i=1}^{n_h}$

Initialize:
- Let G and F be generators for domains RGB and NDVI respectively.
- Let D_X and D_Y be discriminators for domains RGB and NDVI respectively.
- Initialize G with sat2map pretrained weights.
- Initialize F with map2sat pretrained weights.
- Initialize D_X and D_Y with a normal distribution.

Training phase:
- For each epoch:
  - For each batch in $\mathcal{D}_x$ and $\mathcal{D}_y$:
    - Augment x_i and $\hat{y}_i$ using flip technique.
    - Generate $\hat{y}_i = G(x_i)$
    - Generate $\hat{a}_i = F(\hat{y}_i)$
    - Generate $\hat{x}_i = F(y_i)$
    - Generate $\hat{b}_i = G(\hat{x}_i)$

    Update G and F
    - Calculate adversial loss as per Eq. (2)
    - Calculate cycle loss as per Eq. (3)
    - Calculate identity loss as per Eq. (4)
    - Total the loss as per Eq. (5)
    - Update G, F and D_X, D_Y.

Output and testing phase:
- For each i in $\mathcal{D}_h$:
  - Generate $\widehat{y_i^h}$ using $G(x_i^h)$
  - Calculate NRMSE, PSNR, ORB, SSIM using $f(y_i^h, \widehat{y_i^h})$
- Average NRMSE, PSNR, ORB, SSIM.

The model was trained using the Adam optimizer to optimize CycleGAN loss function. We used image flipping as a data augmentation technique during training to diversify our dataset. We tested initial learning rates of 1e-5, 3e-4, 5e-4, 8e-4, and 15e-4 in experiments to determine the most effective rate for the model. Adam optimizer, as detailed in Eq. (6), refines both the generator and discriminator models iteratively with a unique learning rate for each weight [18].

$$\begin{aligned} m_{t+1} &= \beta_1 m_t + (1-\beta_1)\nabla_\theta \mathcal{L}_{GAN}(\theta_t) \\ v_{t+1} &= \beta_2 v_t + (1-\beta_2)\left(\nabla_\theta \mathcal{L}_{GAN}(\theta_t)\right)^2 \\ \hat{m}_{t+1} &= \frac{m_{t+1}}{1-\beta_1^{t+1}} \\ \hat{v}_{t+1} &= \frac{v_{t+1}}{1-\beta_2^{t+1}} \\ \theta_{t+1} &= \theta_t - \frac{\eta \hat{m}_{t+1}}{\sqrt{\hat{v}_{t+1}} + \epsilon} \end{aligned} \quad (6)$$

For efficient training convergence, we implemented a learning rate scheduler using cosine annealing and plateau decay over 200 epochs. The core component, cosine annealing, adjusts the learning rate over training iterations as outlined by Eq. (7) [19].

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + \cos\left(\frac{T_{\text{cur}}}{T_{max}}\pi\right)\right) \quad (7)$$

In the plateau learning rate policy, we start with an initial learning rate. This rate remains unchanged unless the loss doesn't decrease for a "patience" period of 5 epochs. If the loss doesn't decrease during this period, the learning rate is then reduced by a "decay factor" of 0.2 [20].

E. Non-neural Synthetic NDVI Approach

In our study, we have also reproduced an existing strategy that utilizes inter-channel image manipulations to create a synthetic NDVI depiction. We have integrated a variety of techniques from previous research, the formulas for which are outlined in Table 4. This method enables the extraction of synthetic NDVI from RGB images, bypassing the requirement for neural networks and offering alternative procedures for environmental and agricultural examination. We have re-executed these methodologies using our dataset for comparison with our proposed approach.

TABLE IV
FORMULA SYNTHETIC NDVI USING RGB IMAGE NON NEURALNETS BASED

Method	Formula
ExG [3]	$ExG = 2 * Green - Red - Blue$
ExR [3]	$ExR = 1.4 * Red - Green$
ExGr [3]	$ExGr = ExGr - ExR$
NGRDI [4]	$NGRDI = \frac{Green - Red}{Green + Red}$
GLI [5]	$GLI = \frac{2 * Green - Red - Blue}{2 * Green + Red + Blue}$
IKAW [6]	$IKAW = \frac{Red - Blue}{Red + Blue}$
MGRVI [7]	$MGRVI = \frac{Green^2 - Red^2}{Green^2 + Red^2}$

F. Image Similarity Assessment

Assessing the similarity between generated and actual NDVI images is vital for performance evaluation. We used multiple metrics to thoroughly examine the quality of the generated images. The Normalized Root Mean Square Error (NRMSE) calculated using Eq. (8) metric measures the average magnitude of pixel-wise differences between two images, normalized by the range of pixel intensities, with lower values indicating a closer match [21].

$$NRMSE = \frac{\sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y})^2}{n}}}{y_{max} - y_{min}} \tag{8}$$

Furthermore, we used the Peak Signal-to-Noise Ratio (PSNR) to assess the quality of the image and the accuracy of its reconstruction. PSNR, a commonly used measure in image compression, computes the proportion between the highest potential strength of a signal and the intensity of interfering noise calculated by Eq. (10) [22]. To measure PSNR we have to calculate Mean Squared Error (MSE) using Eq. (9) [23]. MSE measures the average squared differences between the estimated and actual pixels [22].

$$MSE = \frac{1}{HW} \sum_{p=0}^{H} \sum_{q=1}^{W} [\hat{y}(p, q) - y(p, q)]^2 \tag{9}$$

$$PSNR = 10\log_{10} (maxVal^2)/MSE \tag{10}$$

In addition to pixel-wise metrics, we utilized the Oriented FAST and Rotated BRIEF (ORB) feature-based metric [24]. This method identifies and matches keypoints between the generated and reference images using the ORB algorithm. The similarity score is computed as the ratio of matched regions to the total matches. To calculate the similarity between these features, we employed a Brute-Force matcher that uses Hamming distance. Higher scores indicate a stronger similarity in feature representations, encompassing structural and perceptual elements that go beyond simple pixel-level differences [25]. Furthermore, we incorporated the Structural Similarity Index (SSIM), a metric designed to align with human visual perception. Unlike pixel-wise metrics, SSIM considers changes in structural information, perceived brightness, and contrast, measuring the structural similarity between two images by considering luminance, contrast, and structure calculated by Eq. (11) [26]. A value of 1 indicates perfect structural similarity, reflecting the perceptual quality of the generated images [27]. In this context, μ is the local means, σ stands for the standard deviation, and σ_{xy} is the cross-covariance for images x and y.

$$SSIM\ (x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_x\sigma_y + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{11}$$

G. Deployment on an Embedded Platform

In this study, we deployed the best-trained model on a low-spec microcomputer device, specifically the Raspberry Pi 4B as seen in Fig. 4 [28]. The Raspberry Pi 4B is equipped with a 64-bit System on a Chip (SoC) that runs at 1.5GHz, powered by a quad-core Cortex-A72 (ARM v8) and the Broadcom BCM2711 chipset, with 4GB RAM. The rationale behind selecting the Raspberry Pi was its cost-effectiveness, aligning with the research goal of reducing the expensive costs associated with producing NDVI representations. Furthermore, the Raspberry Pi 4B's lightweight nature, weighing only 46 grams, and compact dimensions of 88*58*19.5mm. Additionally, its power consumption is relatively low, with an idle state drawing 540 mA (2.7 W), 1010 mA (5.1 W) under an uncached load of 100 requests with a concurrency of 10, and 1280 mA (6.4 W) under a 400% CPU load (stress --cpu 4), making it compatible with efficient battery power connections.

Fig. 4 An embedded microcomputer Raspberry Pi 4B where we implement our trained model

III. Results and Discussion

A. Evaluation on Training and Testing Phase

The learning curves in Fig. 5 and Fig. 6 show how weight initialization methods and initial learning rates affect the training performance of various configuration learning rate schedulers in a CycleGAN model. We show the loss function plotted over training iterations using LOESS smoothing technique to make the trends easier to interpret [29]. The "sat2map" initialization consistently outperforms the random "normal" initialization across all learning rate policies and values. This superior performance can be attributed to the pre-trained weights from a related task provide a more favorable initial state, allowing the optimization process to converge faster and potentially reach a lower final loss value [30].

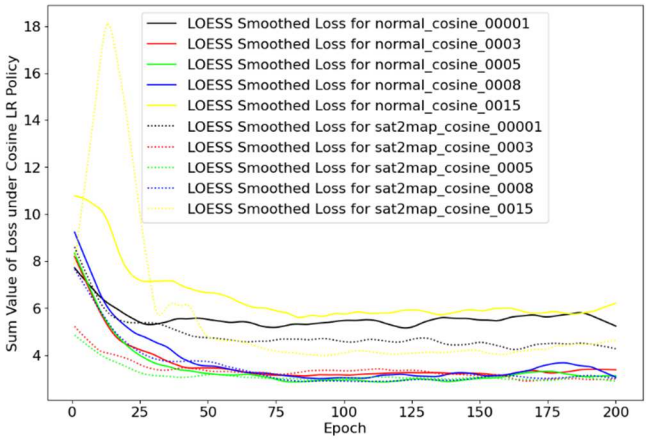


Fig. 5 CycleGAN Total Loss Components under Cosine LR Policy with LOESS Smoothing

Furthermore, the choice of learning rate scheduler has a significant impact on the convergence behavior and stability of the training process. The cosine learning rate policy exhibits smoother convergence with fewer oscillations compared to the plateau policy. This can be explained by the gradual annealing of the learning rate in the cosine policy, which allows for faster convergence in the initial stages while enabling finer adjustments towards the end, leading to a more stable optimization process. In contrast, the plateau learning rate policy involves abrupt changes in the learning rate, which can cause oscillations and plateaus in the loss curves. These abrupt

changes can disrupt the optimization process, potentially causing it to oscillate or get stuck in local minima, resulting in the observed instability and plateaus.

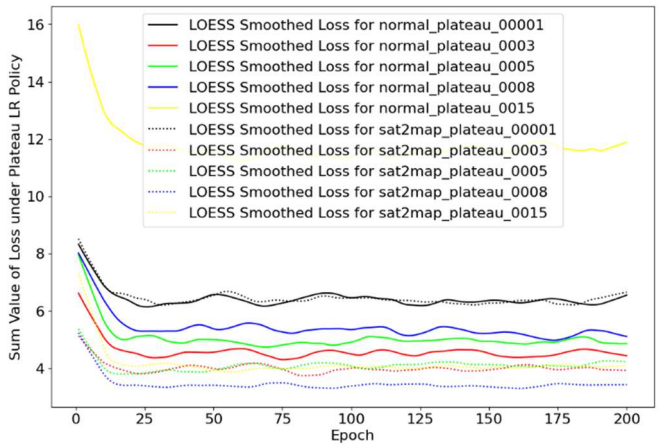


Fig. 6 CycleGAN Total Loss Components under Plateau LR Policy with LOESS Smoothing

The numerical evidence in Table 5 corroborates the superiority of the "sat2map" initialization combined with the cosine learning rate policy, achieving the lowest final loss values across all learning rates.

TABLE V
SUMMATION OF CYCLEGAN LOSS COMPONENTS IN THE LAST ITERATION

Initiallr	Normal		Fine-tuned	
	plateau	cosine	plateau	cosine
1e-5	6.711	5.047	6.554	4.57
3e-4	4.827	3.449	4.123	2.841
5e-4	5.877	3.642	3.762	3.052
8e-4	4.237	**3.051**	3.136	**2.75**
15e-4	9.885	6.807	4.301	3.181

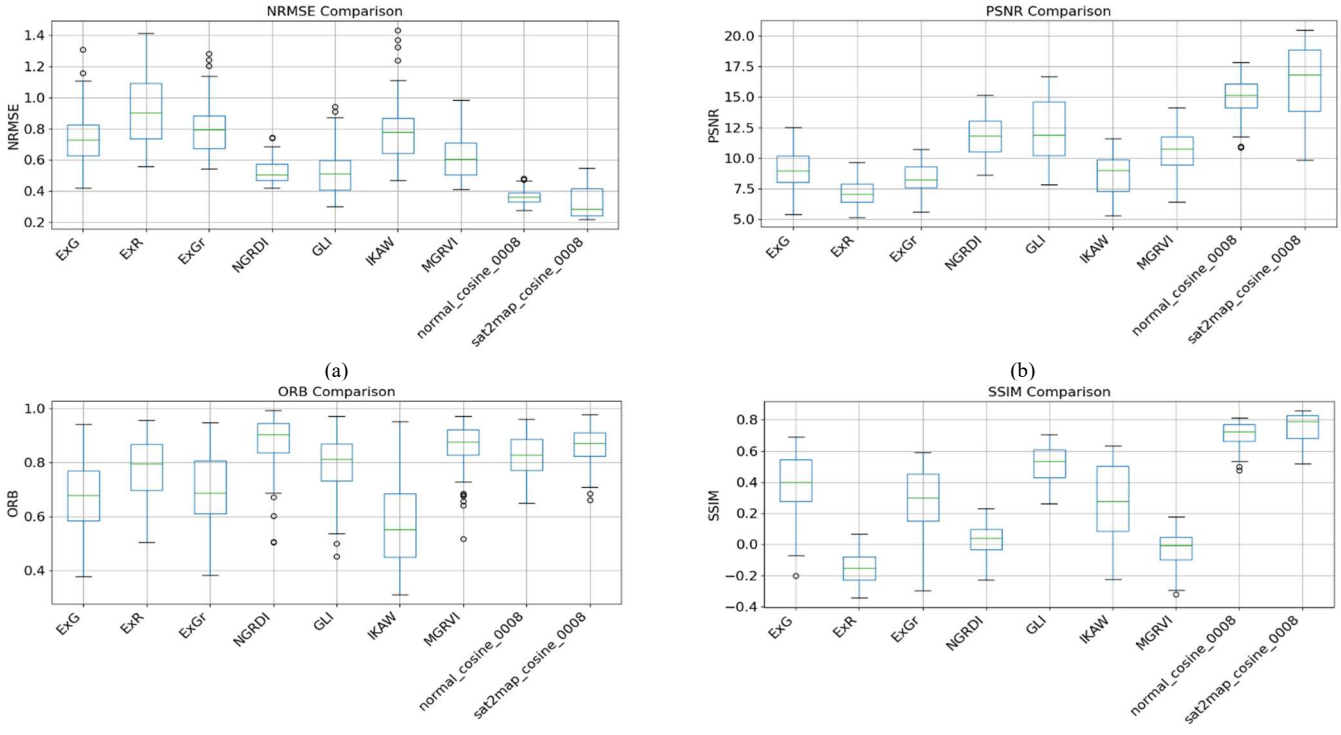
Additionally, an initial learning rate of 8e-4 is identified as the optimal value, yielding the lowest loss in all scenarios. Notably, as the initial learning rate increases from 1e-5 to 8e-4, the loss values generally decrease for both normal and fine-tuned setups. However, beyond 8e-4, the losses start to increase again. This could be due to the initial learning rate being too large. Although the learning rate decays over time from learning rate adjustment policies, the initial large steps can cause the optimizer to overshoot and oscillate around the minimum, preventing it from converging smoothly or getting stuck in suboptimal regions of the loss landscape. Therefore, it is crucial to find a good initial learning rate that allows the optimizer to converge effectively without overshooting or getting stuck. Results showed that both too-large and too-small learning rates can hinder the model's ability to learn effectively [31].

B. Comparation with the Synthetic NDVI Non-neural Approach

We evaluated both the existing and our proposed approach using our dataset, and subsequently contrasted them with the actual NDVI values. The CycleGAN-centric technique demonstrated enhanced efficacy in comparison to the current synthetic NDVI methods, as evidenced by multiple evaluation metrics illustrated in Table 6 and the boxplot in Fig. 7.

TABLE VI
AVERAGE SCORE OF PERFORMANCE METRICS ON TEST DATA TOWARDS TRUE NDVI

Method	NRMSE	PSNR	ORB	SSIM
ExG [3]	0.742	9.104	0.687	0.381
ExR [3]	0.931	7.125	0.786	-0.142
ExGr [3]	0.803	8.364	0.702	0.283
NGRDI [4]	0.529	11.890	**0.878**	0.038
GLI [4], [5]	0.517	12.359	0.792	0.517
IKAW [6]	0.781	8.688	0.578	0.297
MGRVI [7]	0.618	10.657	0.857	-0.028
Ours (normal)	0.367	15.041	0.822	0.710
Ours (fine-tuned)	**0.327**	**16.330**	0.859	**0.757**



(a) (b) (c) (d)

Fig. 7 Boxplot: (a) NRMSE (b) PSNR (c) ORB (d) SSIM

Specifically, the fine-tuned model obtained the lowest NRMSE of 0.327, indicating the generated NDVI images closely approximated the true values. The fine-tuned model attained the highest PSNR of 16.330 dB, implying the produced NDVI images exhibited the highest quality and similarity to the true images among all methods. Fig. 8 shows how different vegetation indices representation from aerial imagery compare to each other.

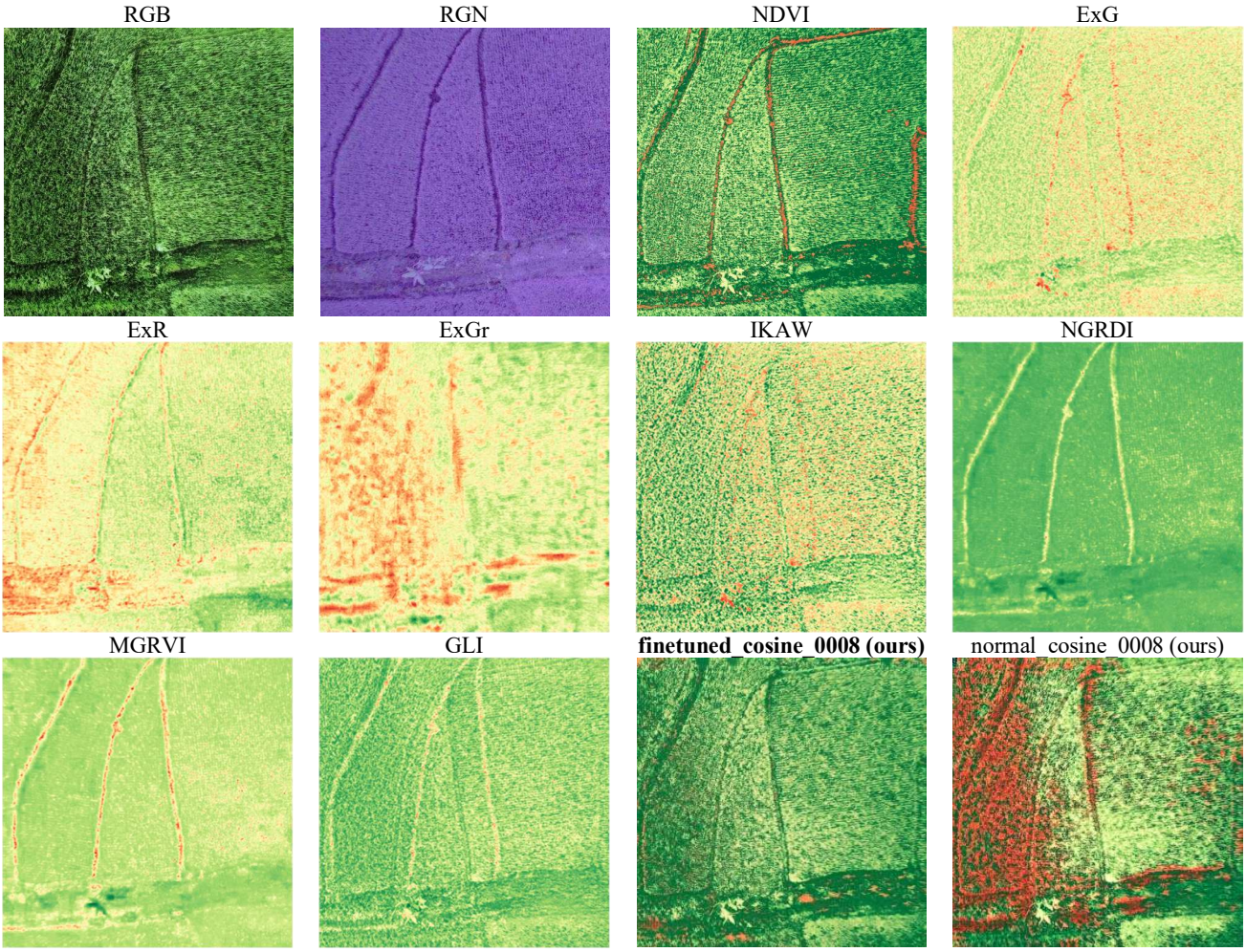


Fig. 8 Comparative visualization of vegetation indices from aerial imagery

Regarding feature similarity, the fine-tuned model's ORB score of 0.859 was comparable to top-performing synthetic methods like NGRDI's 0.878 and MGRVI's 0.857. Furthermore, the fine-tuned model achieved the highest SSIM of 0.757, demonstrating the highest structural similarity between the generated and true NDVI images. SSIM can take on values within the range of -1 to 1 [32]. From the results, negative SSIM values were observed for some methods. This situation can arise when the generated NDVI images have contrasting structural patterns, such as inverted intensities or distorted representation compared to the true NDVI images.

The discrepancy between SSIM and ORB values arises from their distinct operational focuses. SSIM assesses image quality based on perceived changes in structural information, considering luminance, contrast, and structure, which is more aligned with human visual perception [33]. On the other hand, ORB is designed for fast key point detection and matching, focusing on identifying and comparing local features between images [34]. Images can have structural similarity (similar key point locations) but differ in intensity levels, leading to a high ORB score but a low (or negative) SSIM score.

From Fig. 8, we can see that the RGB image depicts the natural appearance of foliage but fails to capture intricate details, serving as a precursor for deriving more refined indices. The RGN image, composed of red, green, and near-infrared channels, is instrumental in producing the NDVI, which highlights vegetative regions in green, with other colors identifying barren zones. Among the array of indices generated, the finetuned_cosine_0008 index stands out for its exceptional detail and contrast, offering a clear differentiation among various paddy plants and closely mirroring the ground truth or NDVI imagery. In contrast, the normal_cosine_0008 index, while providing clarity and distinction, exhibits red tinges that falsely suggest unhealthy vegetation, a discrepancy not observed in the ground truth representation.

C. Comparation of Processing Time on Embedded Platform

We conducted an evaluation of the processing duration for existing techniques utilizing our compact computing device, the Raspberry Pi 4B. The results in Table 7 reveal that the GLI method demonstrates the shortest processing time, completing tasks in only 0.221067 seconds, whereas CycleGAN exhibits the lengthiest processing duration,

requiring 21.0077 seconds. This is significantly slower than the other methods, with the processing time being almost 100 times slower than the next slowest method, MGRVI.

TABLE VII
PROCESSING TIME FOR DIFFERENT APPROACH USING RASPBERRY PI 4B

Method	Processing time (seconds)
ExG	0.247562
ExR	0.262393
ExGr	0.269933
NGRDI	0.243913
GLI	**0.221067**
IKAW	0.247733
MGRVI	0.272105
CycleGAN (ours)	21.0077

On average, excluding CycleGAN's outlier performance, the methods exhibit an approximate processing time of 0.2521 seconds. The underlying reason for the slow computation of CycleGAN is its AI-based architecture. During inference, it needs to perform computational operations that involve a large number of parameters that have been honed through extensive training. Consequently, it demands more time for inference relative to other methods that rely on straightforward channel manipulation based on predefined algorithms. Despite its slower speed, CycleGAN yields higher accuracy, indicating a trade-off between processing time and accuracy. Consequently, despite its slower processing, CycleGAN remains an advantageous choice for generating synthetic NDVI when prioritizing accuracy over speed.

IV. CONCLUSION

This research proposed fine-tuned CycleGAN models for transforming RGB images into vegetation indices, enhancing the potential for more efficient and cost-effective smart farming monitoring using UAV and RGB camera. CycleGAN excelled in creating synthetic NDVI images, surpassing other methods in fidelity, quality, and structural coherence. Its efficacy was highlighted by its superior performance across a range of metrics such as NRMSE, PSNR, ORB feature similarity, and SSIM, indicating its capability to generate high-quality NDVI images. The results were as follows: NRMSE of 0.327, PSNR of 16.330, ORB score of 0.859, and SSIM of 0.757. The experimental results demonstrated the importance of appropriate weight initialization, initial learning rate and schedulers for efficient CycleGAN training. Leveraging transfer learning through the "sat2map" weight initialization and employing a gradually annealing cosine learning rate policy significantly improved the training performance, leading to faster convergence, lower final loss values, and a more stable optimization process. Future work could focus on addressing the high computational resource requirements for training and inferring GAN models.

NOMENCLATURE

$\hat{a}_i$	the reconstruction of RGB image
$\hat{b}_i$	the reconstruction of NDVI image
β_1, β_2	exponential decay rates
C_1, C_2	constants to stabilize the division
D_X, D_Y	discriminators for domains RGB and NDVI respectively
F	generators for domains NDVI
G	generators for domains RGB
$\mathcal{L}_{\text{GAN}}$	overall objective loss function
$\mathcal{L}_{\text{adv}}$	adversarial loss
$\mathcal{L}_{\text{cyc}}$	cycle consistency loss
$\mathcal{L}_{\text{idt}}$	identity loss
λ_1, λ_2	weighting factors for GAN loss
m_{t+1}	first-moment of the gradients
n	the quantity of data points within a dataset
p, q	the pixel coordinates of the image
v_{t+1}	second moment of the gradient
x	image from domain RGB of size $HxWxC$
y	image from domain NDVI of size $HxWxC$
η	learning rate
θ_t	the weights and biases during time t
ϵ	small constant
μ_x, μ_y	the local means of images x and y respectively
σ_x, σ_y	the standard deviations of images x and y respectively
σ_{xy}	the cross-covariance between images x and y

ACKNOWLEDGMENT

Funding for this research was provided by *Penelitian Produk Vokasi* (P2V) Scheme 2024, administered by *Direktorat Jenderal Pendidikan Vokasi.*

REFERENCES

[1] J. J. Chen, S. Zhen, and Y. Sun, "Estimating Leaf Chlorophyll Content of Buffaloberry Using Normalized Difference Vegetation Index Sensors," *HortTechnology*, vol. 31, no. 3, pp. 297–303, Jun. 2021, doi:10.21273/horttech04808-21.

[2] M. Y. Abu Sari, Y. M. Mohmad Hassim, R. Hidayat, and A. Ahmad, "Monitoring Rice Crop and Paddy Field Condition Using UAV RGB Imagery," *JOIV : International Journal on Informatics Visualization*, vol. 5, no. 4, p. 469, Dec. 2021, doi: 10.30630/joiv.5.4.742.

[3] F. Kazemi and E. Ghanbari Parmehr, "Evaluation of RGB Vegetation Indices Derived from UAV Images for Rice Crop Growth Monitoring," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. X-4/W1-2022, pp. 385–390, Jan. 2023, doi:10.5194/isprs-annals-x-4-w1-2022-385-2023.

[4] K. Upendar, K. N. Agrawal, N. S. Chandel, and K. Singh, "Greenness identification using visible spectral colour indices for site specific weed management," *Plant Physiology Reports*, vol. 26, no. 1, pp. 179–187, Jan. 2021, doi: 10.1007/s40502-020-00562-0.

[5] A. L. A. Ribeiro et al., "Vegetation Indices for Predicting the Growth and Harvest Rate of Lettuce," *Agriculture*, vol. 13, no. 5, p. 1091, May 2023, doi: 10.3390/agriculture13051091.

[6] L. S. Eng, R. Ismail, W. Hashim, and A. Baharum, "The Use of VARI, GLI, and VIgreen Formulas in Detecting Vegetation In aerial Images," *International Journal of Technology*, vol. 10, no. 7, p. 1385, Nov. 2019, doi: 10.14716/ijtech.v10i7.3275.

[7] J. Jiang et al., "Using Digital Cameras on an Unmanned Aerial Vehicle to Derive Optimum Color Vegetation Indices for Leaf Nitrogen Concentration Monitoring in Winter Wheat," *Remote Sensing*, vol. 11, no. 22, p. 2667, Nov. 2019, doi: 10.3390/rs11222667.

[8] E. Estrada-Peraza, E. Alvarez-Huezo, G. Girón-Morales, and Y. Rodriguez-Gallo, "RGB Image-Based Coffee Rust Detection: Application of Vegetation Indices and Algorithm Development," 2023 *IEEE Central America and Panama Student Conference (CONESCAPAN)*, vol. 13, pp. 23–28, Sep. 2023, doi: 10.1109/conescapan60431.2023.10328429.

[9] H. Yang, J. Liu, H. Sun, and H. Zhang, "PACL: Piecewise Arc Cotangent Decay Learning Rate for Deep Neural Network Training," *IEEE Access*, vol. 8, pp. 112805–112813, 2020, doi: 10.1109/access.2020.3002884.

[10] J. Raitoharju, "Convolutional neural networks," *Deep Learning for Robot Perception and Cognition*, pp. 35–69, 2022, doi: 10.1016/b978-0-32-385787-1.00008-7.

[11] H. Darmawan, M. Yuliana, and Moch. Z. Samsono Hadi, "GRU and XGBoost Performance with Hyperparameter Tuning Using GridSearchCV and Bayesian Optimization on an IoT-Based Weather

Prediction System," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 13, no. 3, pp. 851–862, Jun. 2023, doi: 10.18517/ijaseit.13.3.18377.

[12] C. Davidson, V. Jaganathan, A. N. Sivakumar, J. M. P. Czarnecki, and G. Chowdhary, "NDVI/NDRE prediction from standard RGB aerial imagery using deep learning," *Computers and Electronics in Agriculture*, vol. 203, p. 107396, Dec. 2022, doi: 10.1016/j.compag.2022.107396.

[13] H. Darmawan, M. Yuliana, and Moch. Z. S. Hadi, "Cloud-based Paddy Plant Pest and Disease Identification using Enhanced Deep Metric Learning and k-NN Classification with Augmented Latent Fusion," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 6, pp. 158–170, Dec. 2023, doi: 10.22266/ijies2023.1231.14.

[14] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251, Oct. 2017, doi: 10.1109/iccv.2017.244.

[15] I. Goodfellow et al., "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, Jul. 2017, doi: 10.1109/cvpr.2017.632.

[17] M. Wojciuk, Z. Swiderska-Chadaj, K. Siwek, and A. Gertych, "Improving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization," *Heliyon*, vol. 10, no. 5, p. e26586, Mar. 2024, doi: 10.1016/j.heliyon.2024.e26586.

[18] B. Yan, Z. Yang, H. Sun, and C. Wang, "ADE-CycleGAN: A Detail Enhanced Image Dehazing CycleGAN Network," *Sensors*, vol. 23, no. 6, p. 3294, Mar. 2023, doi: 10.3390/s23063294.

[19] X. Xu et al., "Artificial intelligence driven digital whole slide image for intelligent recognition of different development stages of tongue tumor via a new deep learning framework," *Engineering Reports*, vol. 6, no. 1, Jun. 2023, doi: 10.1002/eng2.12706.

[20] H. Darmawan, M. Yuliana, and Moch. Z. Samsono Hadi, "Realtime Weather Prediction System Using GRU with Daily Surface Observation Data from IoT Sensors," *2022 International Electronics Symposium (IES)*, vol. vii, pp. 221–226, Aug. 2022, doi: 10.1109/ies55876.2022.9888468.

[21] T. Banet, A. G. Smith, R. McGrail, D. H. McNear, and H. Poffenbarger, "Toward improved image-based root phenotyping: Handling temporal and cross-site domain shifts in crop root segmentation models," *The Plant Phenome Journal*, vol. 7, no. 1, Jan. 2024, doi: 10.1002/ppj2.20094.

[22] Y. Chen, A. Janowczyk, and A. Madabhushi, "Quantitative Assessment of the Effects of Compression on Deep Learning in Digital Pathology Image Analysis," *JCO Clinical Cancer Informatics*, no. 4, pp. 221–233, Nov. 2020, doi: 10.1200/cci.19.00068.

[23] S. Rani, A. Kurniawardhani, and Y. A. W. Rendani, "Steganography on Digital Color Image Using Modulo Function and Pseudo-Random Number Generator," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 11, no. 6, p. 2470, Dec. 2021, doi: 10.18517/ijaseit.11.6.12687.

[24] F. D. Adhinata, A. Harjoko, and - Wahyono, "Object Searching on Real-Time Video Using Oriented FAST and Rotated BRIEF Algorithm*," International Journal on Advanced Science, Engineering and Information Technology*, vol. 11, no. 6, p. 2518, Dec. 2021, doi: 10.18517/ijaseit.11.6.12043.

[25] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image Quality Assessment: Unifying Structure and Texture Similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020, doi: 10.1109/tpami.2020.3045810.

[26] I. H. Suvon et al., "Business Category Classification via Indistinctive Satellite Image Analysis Using Deep Learning," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 13, no. 6, pp. 2219–2230, Dec. 2023, doi: 10.18517/ijaseit.13.6.19059.

[27] J. Peng et al., "Implementation of the structural SIMilarity (SSIM) index as a quantitative evaluation tool for dose distribution error detection," *Medical Physics*, vol. 47, no. 4, pp. 1907–1919, Jan. 2020, doi: 10.1002/mp.14010.

[28] H. Tarek, H. Aly, S. Eisa, and M. Abul-Soud, "Optimized Deep Learning Algorithms for Tomato Leaf Disease Detection with Hardware Deployment," *Electronics*, vol. 11, no. 1, p. 140, Jan. 2022, doi: 10.3390/electronics11010140.

[29] A. Von Eye, W. Wiedermann, and S. Von Weber, "Configural analysis of oscillating progression," *Journal for Person-Oriented Research*, vol. 7, no. 1, pp. 14–21, Aug. 2021, doi: 10.17505/jpor.2021.23448.

[30] E. Hattula, L. Zhu, J. Raninen, J. Oksanen, and J. Hyyppä, "Advantages of Using Transfer Learning Technology with a Quantative Measurement," *Remote Sensing*, vol. 15, no. 17, p. 4278, Aug. 2023, doi: 10.3390/rs15174278.

[31] Md. J. Uddin, Y. Li, Md. A. Sattar, Z. Most. Nasrin, and C. Lu, "Effects of Learning Rates and Optimization Algorithms on Forecasting Accuracy of Hourly Typhoon Rainfall: Experiments With Convolutional Neural Network," *Earth and Space Science*, vol. 9, no. 3, Mar. 2022, doi: 10.1029/2021ea002168.

[32] W. Nawaz, M. H. Siddiqi, and A. Almadhor, "Adaptively Directed Image Restoration Using Resilient Backpropagation Neural Network," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, May 2023, doi: 10.1007/s44196-023-00259-w.

[33] V. Mudeng, M. Kim, and S. Choe, "Prospects of Structural Similarity Index for Medical Image Analysis," *Applied Sciences*, vol. 12, no. 8, p. 3754, Apr. 2022, doi: 10.3390/app12083754.

[34] J. Zhang, "Research on the algorithm of image feature detection and matching," *Applied and Computational Engineering*, vol. 5, no. 1, pp. 527–535, May 2023, doi: 10.54254/2755-2721/5/20230636.