

Oil Palm USB (Unstripped Bunch) Detector Trained on Synthetic Images Generated by PGGAN

Wahyu Sapto Aji ^{1*}, Kamarul Hawari bin Ghazali ², Son Ali Akbar ³

^{1,2,3} Faculty of Electrical and Electronics Engineering (FTKEE), Universiti Malaysia Pahang Al-Sultan Abdullah, Pahang, Malaysia

Email: ¹ wahyu.aji@te.uad.ac.id, ² kamarul@ump.edu.my, ³ son.akbar@te.uad.ac.id

*Corresponding Author

Abstract—Identifying Unstripped Bunches (USB) is a pivotal challenge in palm oil production, contributing to reduced mill efficiency. Existing manual detection methods are proven time-consuming and prone to inaccuracies. Therefore, we propose an innovative solution harnessing computer vision technology. Specifically, we leverage the Faster R-CNN (Region-based Convolution Neural Network), a robust object detection algorithm, and complement it with Progressive Growing Generative Adversarial Networks (PGGAN) for synthetic image generation. Nevertheless, a scarcity of authentic USB images may hinder the application of Faster R-CNN. Herein, PGGAN is assumed to be pivotal in generating synthetic images of Empty Fruit Bunches (EFB) and USB. Our approach pairs synthetic images with authentic ones to train the Faster R-CNN. The VGG16 feature generator serves as the architectural backbone, fostering enhanced learning. According to our experimental results, USB detectors that were trained solely with authentic images resulted in an accuracy of 77.1%, which highlights the potential of this methodology. However, employing solely synthetic images leads to a slightly reduced accuracy of 75.3%. Strikingly, the fusion of authentic and synthetic images in a balanced ratio of 1:1 fuels a remarkable accuracy surge to 87.9%, signifying a 10.1% improvement. This innovative amalgamation underscores the potential of synthetic data augmentation in refining detection systems. By amalgamating authentic and synthetic data, we unlock a novel dimension of accuracy in USB detection, which was previously unattainable. This contribution holds significant implications for the industry, ensuring further exploration into advanced data synthesis techniques and refining detection models.

Keywords—PGGAN; USB; Detector; Faster R-CNN; Synthetic Image.

I. INTRODUCTION

USB (Unstripped Bunch) is a cause of losses in palm oil production [1]. The average loss through USB in Malaysia is indicated to be 0.05% [2]. In Indonesia, another significant palm oil producer, the loss due to USB is 2.2% [3]. If the USB is not appropriately managed, losses from the USB may reach up to 40% in certain circumstances [4]. Sadly, manual USB monitoring is still practiced today, which has to change [2]. Faster R-CNN, a reasonably accurate object detector, may be used to address this issue. The availability of USB images limits the use of Faster R-CNN in addressing this issue.

II. USB (UNSTRIPPED BUNCH) AND EFB (EMPTY FRUIT BUNCH)

Unstripped Bunch (USB) and Empty Fruit Bunch (EFB) are two terms used to describe the processed oil palm fruit

bunches that exit a thresher in a palm oil mill while still containing oil palm fruitlet. Fig. 1 shows a USB image, while Fig. 2 depicts an EFB.

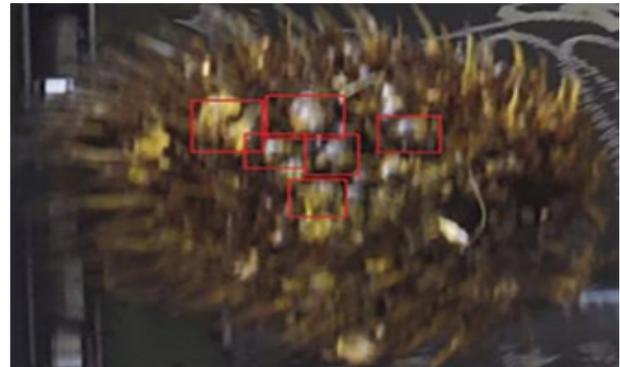


Fig. 1. USB with fruitlet (marked with red squares) attached



Fig. 2. EFB (Empty Fruit Bunch) sample

The threshing process produces USB and EFB in a palm oil mill. Adzmi et al. (2012) classified USBs as empty bunches with more than 20 fruitlets attached [5]. According to different research, USB is a bunch of oil palm fruit that contains at least 30% fruitlet [6].

III. SYNTHETIC DATASET AND PGGAN

A. Dataset for Specific Applications

Deep learning (DL) or advanced machine learning models have been demonstrated to require supplies of large-scale datasets [7]. As a general rule of thumb, thousands of photos per category are required to train DL models in order to attain human-level performance [8]. Given the benefits of data size, the computer vision community has developed several extensive image datasets with millions of labeled images, including ImageNet [9], Microsoft Common Objects in

Context [10], Large Vocabulary Instance Segmentation [11], and Open Images [12], to name a few.

There are currently relatively few publicly accessible image datasets in specialized applications, such as the agricultural product processing industry, such as palm oil mills, that contain hundreds or even thousands of photos per category, comparable to the computer vision datasets discussed above. Leveraging modern AI capabilities, particularly DL techniques, in agriculture is severely inhibited by the lack of large-scale image datasets related to agricultural or processing activities. Data/image augmentation, which increases the size and variety of data sets, is a popular strategy to overcome the limitations of physically acquired data [13].

B. GAN (Generative Adversarial Network) to Produce Synthetic Datasets

Goodfellow et al. (2014) proposed GAN as a new generative modeling framework [14] to synthesize new data with the same characteristics from training examples, visually approximating the training data set. Various GAN-based methods have been proposed for image synthesis in recent years [15], [16], [17], [18], [19], [20], [21], [22], [23], and [24] with applications spreading rapidly from computer vision and machine learning communities to domain-specific areas such as medical [25] [26], [27], [28], [29], and remote sensing [30], [31], [32] [33], [34], [35], [36], [37], [38], [39], [40], and [41]; industrial process [42], [43], [44], [45], [46], [47], and [48]; and agriculture [49], [50], [51], [52].

Occasionally, models may be unsuitable for executing a particular job due to the characteristics they have been trained. Hence, the process of training a new model is a challenging endeavor. An adequate dataset is necessary for this task. If the required dataset is not accessible, it becomes imperative to generate a dataset that aligns with the particular issue or domain of interest. The ability of GAN to produce synthetic data is an opportunity to overcome this problem, so many researchers use GAN to build synthetic datasets that are used to train object detectors [53], [54] [55], [56], [57], [58], [59], [60], and [61].

The implementation of GAN to build synthetic datasets for agricultural product detection has been carried out by several researchers. It has been used to generate realistic images to train deep learning models, improving fruit recognition performance [62], [63], [64], [65], [66], and [67]. The GAN architecture used includes the CGAN (cycleGAN) [68], [69], and [70]; Boundary Equilibrium GAN [71] and Deep Convolutional GAN [72]. The GAN model used is not limited to the models that have been mentioned but continues to develop; this shows the potential of GAN in improving object detection performance.

Several researchers have also reported improvements in the ability of object detectors trained with synthetic datasets. Fei et al. (2021) reported that a fruit detection model developed with YOLOv3 and enhanced with GAN-synthesized images for the day domain yielded 37.2 mAP (mean Average Precision) instead of 37.0 mAP [64]. The GAN-generated synthetic data implementation for Faster R-CNN is reported to provide performance improvements; with

GAN-augmented data, VGG16-based Faster R-CNN trained with augmented data yields the best average accuracy of 90%, which is 28% higher than the accuracy of VGG16 without data augmentation [73].

In their study, Yuwana et al. (2020) investigated the use of GANs using multilayer perceptron architecture as both the model generator and discriminator. Their research aimed to facilitate illness detection by synthesizing tea leaf images belonging to four distinct classes: healthy and three different types of sick leaves. The classification accuracy of GAN and DCGAN (Deep Convolutional GAN) was 88.84% and 88.86%, respectively, while using 1000 synthetic pictures per class. This indicates an enhancement of around 2.5% compared to the basic model that did not include image augmentation [38]. Goma and El-Latif (2021) achieved a recognition accuracy of 97.9% in their study by using DCGAN to generate synthetic images in facilitating the first identification of tomato plants infected with the tomato mosaic virus. This accuracy was acquired using augmented data, resulting in an improvement of around 1% compared to the accuracy achieved without augmentation [39].

GAN has a high probability of experiencing a mode collapse. Mode collapse is a phenomenon that occurs when the model generator is unable to effectively capture and represent the entire spectrum of potential outputs, resulting in a limited and distinct set of outputs that deviate from the patterns seen in the training data. The discriminator exhibits excessive proficiency, making it unable to discriminate synthetic data generated by the generator, impeding knowledge acquisition. As a result, generators face challenges in producing multiple outputs, leading to a tendency for generators to get stuck in repeating patterns, limiting the available variance in the resulting samples. The duplicate image problem can be solved by modifying the GAN training method.

C. PGGAN

As a modified GAN, Progressively Growing GAN (PGGAN) [74] can solve the problem of duplicate samples [75] using a multi-stage training method. PGGAN utilizes the concept of a progressive neural network that was first proposed by Rusu et al. in 2016 [76]. PGGAN requires synchronously developing generator and discriminator networks and gradual model training from low-resolution (4×4 pixels) to high-resolution images by adding network layers. This strategy substantially increases training speed (two to six times faster) and image stability for large image resolutions. As a result of the incremental expansion of the convolutional layers, the generator and discriminator model can effectively learn coarse-scale details initially and then fine-scale details through the training process.

The PGGAN is a GAN trained on multi-stage resolution; at each stage, the generator and discriminator are added with a new convolution layer with higher resolution. A fundamental GAN comprises two neural network models: a generative model G (Generator) that learns the distribution of unseen training data and a discriminative model D (Discriminator) that learns to classify whether samples are from the training data distribution. The generator takes a random noise vector z as input and synthetic data $G(z)$ as

output; a discriminator D takes x or $G(z)$ as input and a probability $D(x)$ or $D(G(z))$ as output to indicate whether it is synthetic or from the authentic data distribution, as shown in Fig. 3.

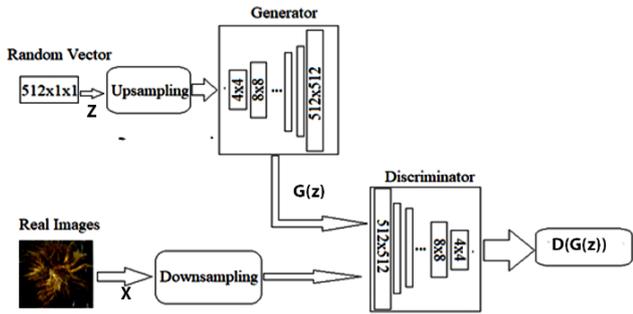


Fig. 3. PGGAN architecture

The generator and the discriminator are both simultaneously trained using the stochastic gradient descent (SGD) algorithm, and their training can be viewed as a two-player minimax game as an objective function equation (1).

$$\max_D \min_G V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_{data}(z)} \left[\log \left(1 - D(G(z)) \right) \right] \quad (1)$$

The discriminator attempts to maximize $V(D, G)$ (probability $D(x)$), whereas the generator attempts to minimize it. In other words, the discriminator distinguishes between the probabilities of images x in $p_{data}(x)$ (the distribution of authentic data) and the noise distribution $p_{data}(z)$. On the contrary, the generator generates samples to deceive the discriminator.

In the PGGAN's discriminator and PGGAN's generator, the network is predominantly composed of up sample and down sample blocks. The backbone structure of the up-sample block in the generator network is upsampling2d-convolution2d-LReLU activation-convolution2d-LReLU, as shown in Fig. 4.

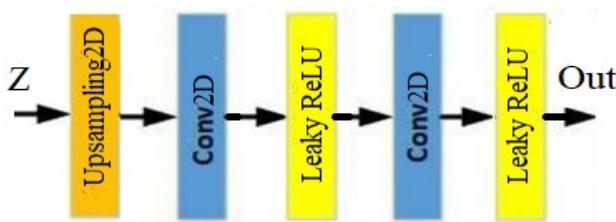


Fig. 4. Up sample block

The backbone structure of the down sample block in the discriminator is conv2d- LReLU-Conv2d-LReLU-averagepooling2d, as depicted in Fig. 5

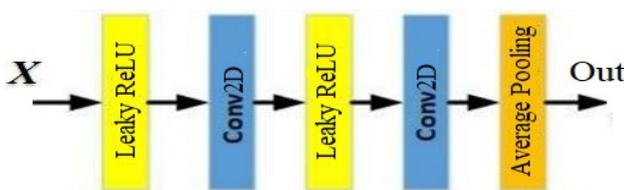


Fig. 5. Down sample block

D. Faster R-CNN

Faster R-CNN, introduced in 2015 by Ross Girshick et al., is one of the most well-known object recognition designs that employ convolutional neural networks alongside YOLO (You Look Only Once) [77] and SSD (Single Shot Detector) [78]. Apart from YOLO, the Faster R-CNN is one of the best-performing object detectors in terms of accuracy [79] [80]. Its good performance has led to the emerging application of the faster R-CNN for the detection of special objects, such as agricultural products [81], [82], [83], [84], and [85].

Faster R-CNN is a two-stage object detector. The first stage is an RPN (Region Proposed Network), and the second is the classifier. Fig. 6 depicts the architecture of Faster R-CNN in general. Various bounding boxes are assigned to be region proposals through the region proposal networks and the deep, fully-connected convolutional neural networks; then, the results are normalized through the ROI (Region of Interest) Pooling Layer. The Fully Connected layers extract image features for object-class classification and perform bounding box regression.

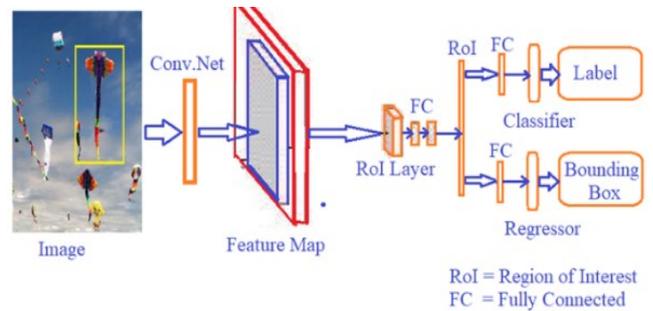


Fig. 6. Faster R-CNN architecture

IV. METHODS

A comprehensive experimental approach was designed to evaluate the potential of synthetic image generation using PGGAN, especially in improving the performance of Unstripped Bunch (USB) detection. The initial experiment was carried out by conducting PGGAN training using authentic images. Authentic images were taken from a surveillance camera installed on a USB conveyor inside the palm oil mill. The total number of authentic images for the PGGAN training dataset was 800 images, consisting of 400 USB and 400 EFB images, respectively. Images used for the training were selected randomly.

The trained PGGAN was then used to produce 1,000 synthetic images, each consisting of 500 EFB images and 500 USB images. These synthetic images were used to train USB detectors using VGG16 as a feature extractor. The VGG-16 was selected as the feature extractor due to its smaller convolution filter, which reduces the tendency of the network to overtrain. Moreover, VGG-16 is the least size-wise model that allows spatial characteristic understanding of an image [86]. The synthetic images produced by PGGAN were split into 800 images for training and 200 for validation.

The USB detector was also trained with the dataset of authentic images, with the same number of images and training parameters as the USB detector was trained with synthetic images. A performance comparison of USB/EFB detectors that were trained with synthetic image dataset and

those trained using authentic image dataset was carried out to assess whether the use of synthetic images generated by PGGAN could improve the performance of the USB detectors.

A. PGGAN Training

Progressive Growing GAN is an extension of the GAN training procedure that involves training the GAN to produce very small images, such as 4×4 pixels, and then progressively increasing the resolution scale of the resulting images to 8×8 , 16×16 , or larger, as desired. This allows progressive GAN to generate synthetic images with a 1024-by-1024-pixel resolution.

However, in this research, the resolution of the images used for USB detector training was limited to 512×512 pixels. The particular 512×512 resolution limit was determined since the largest side of input images in Faster R-CNN was set at 600 pixels.

As seen in Fig. 7, PGGAN was trained in stages. In the first stage, it was trained with a 4×4 image, and the weights generated at this size were used to train a larger resolution in the subsequent stage. Since our study limited the resolution size to 512×512 , the process would be stopped when 512×512 image resolution was achieved.

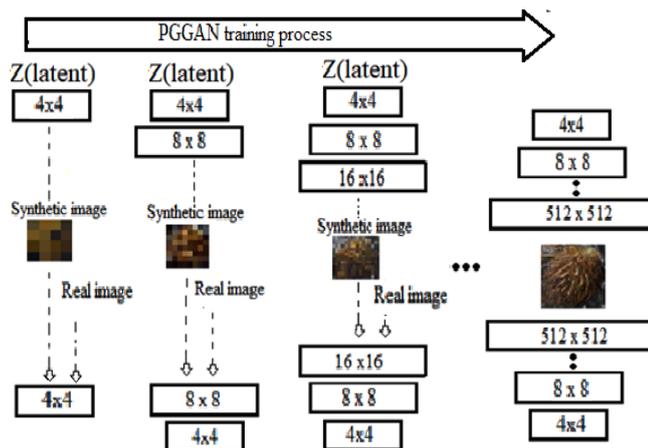


Fig. 7. PGGAN training process

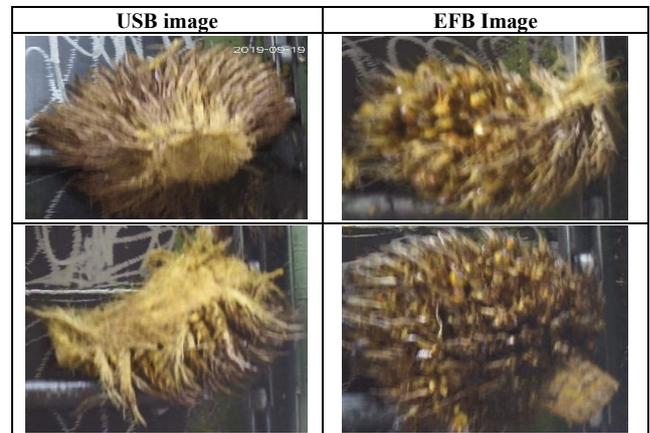
The training parameters are shown in Table I. The Adam optimizer was chosen for optimization because it is relatively insensitive to the training hyperparameters, such as learning rate and momentum; thus, the resulting PGGAN model became more stable. The He normal initializer was selected as the kernel initializer to prevent the gradient from vanishing or exploding and improve the network stability and convergence during training [87]. A linear activation function was used for the output layer to produce a continuous output. A learning rate of 0.001 was chosen to speed up the training process.

TABLE I. PGGAN FINE TUNING TRAINING PARAMETERS

Training parameter	Value
Kernel initializer	He Normal
Optimizer	Adam
Output activation function	Linear
Learning rate	0.001

The total number of images used to train PGGAN was 1000, comprising two datasets: the USB and EFB datasets. Each data set contained 500 related authentic images. Table II shows samples of EFB and USB images that were used to train PGGAN.

TABLE II. USB AND EFB IMAGES SAMPLES FOR PGGAN TRAINING



In this research, PGGAN network training was performed using the NVIDIA GeForce RTX 3080 GPU and took two weeks. PGGAN minibatch sizes varied depending on the available memory budget ($4 \times 4:32$, $8 \times 8:32$, 16×16 , 64×64 , 128×128 , $256 \times 256:8$, and $512 \times 512:2$).

B. USB Detector Training

The USB detector was trained using images generated by PGGAN (synthetic images), authentic images, and a merged dataset of both images. The training for each dataset was carried out separately. A performance comparison of USB detectors that were trained using different datasets (synthetic images, authentic images, and images of both types) was conducted. The USB detector training parameters were set using the same values. Table III lists the specification of the training parameters.

TABLE III. FINE TUNING TRAINING PARAMETERS OF THE USB DETECTOR

Training parameter	Value
Anchor ratios	[1:1], [1.56:1.2], [1.2:1.56]
ROI threshold	0.8
Backbone	VGG16
Learning rate	0.00001
Classifier regression	8.0, 8.0, 4.0, 4.0

In this research, a similarity test was also carried out between the synthetic images and the authentic images. The parameters used for the image similarity test were PSNR, SSIM, and VIF. The PSNR and VIF evaluate whether the two images are the same, while the SSIM tests whether the two images are structurally similar. In the test, the PSNR value was expected to be below 20 dB [44], and the VIF score approached zero, meaning the two images were different [45]. In contrast, the SSIM value was expected to be above 50% (confidence in recognition for SSIM is 0.580) [46], meaning that both images had the same structure, that the USB and EFB images produced by PGGAN still represented an oil palm bunch but still maintained the unique characteristics that differentiate USB from EFB.

V. RESULT AND DISCUSSION

A. Training Performances of The USB Detectors

The PGGAN model that had been trained was used to generate 1000 images. Table IV depicts USB and EFB PGGAN-generated sample images. Apparently, PGGAN was assumed to be capable of producing synthetic images differently from the authentic image. Therefore, an image similarity test was carried out to prove the assumption.

TABLE IV. SAMPLES OF USB AND EFB SYNTHETIC IMAGES PRODUCED BY PGGAN AND THE REPRESENTATIVE AUTHENTIC IMAGES

Image's number	Authentic image	Synthetic image
1		
2		
3		
4		

The synthetic images were contrasted with authentic images that were used to train PGGAN. Several test parameters, including SSIM (Structural Similarity Index Measure), PSNR (Peak Signal to Noise Ratio), and VIF (Visual Information Fidelity), were conducted to evaluate the similarity of these images.

Table V presents similarity test results for the sample images. According to the PSNR scores, the average value was 13.5377 dB (under 20 dB). Whereas, the average VIF was 0.0325 (close to zero). The results of the similarity test suggested noticeable differences between the produced USB and EFB images and the authentic images.

TABLE V. SIMILARITY TEST RESULT OF THE SAMPLE IMAGES

Parameter	Image 1	Image 2	Image 3	Image 4	Average
PSNR	15.2241	12.5531	12.1678	14.2057	13.5377
VIF	0.0497	0.0235	0.0188	0.0244	0.0325
SSIM	0.8295	0.6638	0.6305	0.7310	0.7137

According to the SSIM parameters, whose average value was 0.7137 (or 71.37%, which was above 50%), it was evident that the images exhibited similarity despite their inherent disparities. Despite their distinctiveness, the images presented the defining features of a USB, which included oil palm fruitlets, as well as the defining features of an EFB, which was a lack of fruitlets. Consequently, the synthetic images portrayed an oil palm bunch effectively.

Fig. 8 to Fig. 13 show the training performance of the USB detector on different training datasets. The blue lines represent the training performance of a USB detector that was trained using a synthetic dataset (which contains only PGGAN-generated images); the red lines represent the training performance of the USB detector trained using a merged dataset (which contains authentic and synthetic images with a 1:1 ratio). The yellow lines represent the training performance of the USB detector trained on an authentic dataset (which contains only authentic images).

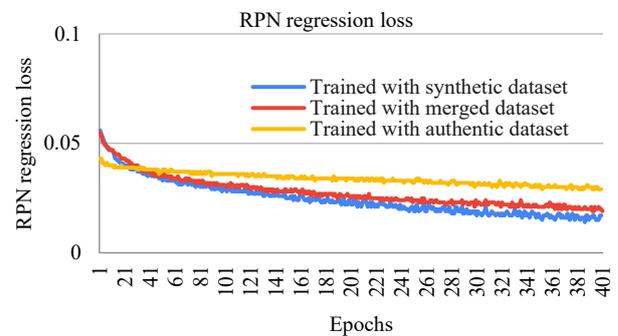


Fig. 8. RPN regression loss

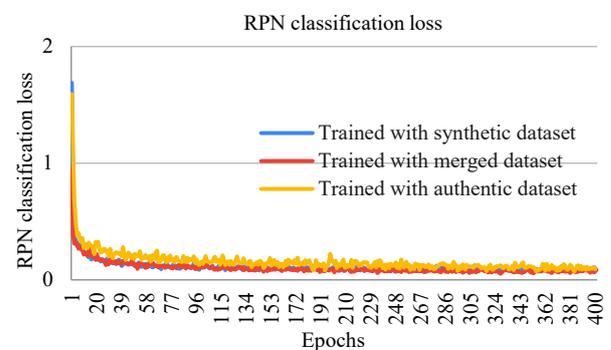


Fig. 9. RPN classification loss

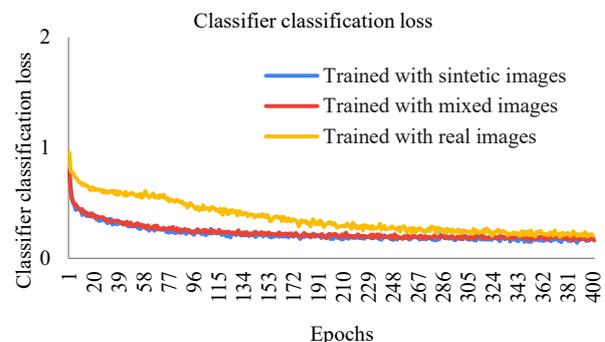


Fig. 10. Classifier classification loss

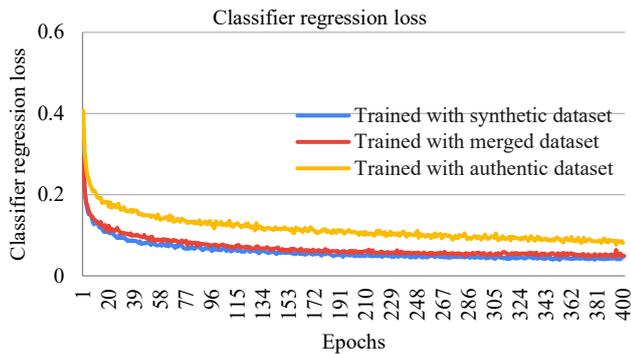


Fig. 11. Classifier regression loss

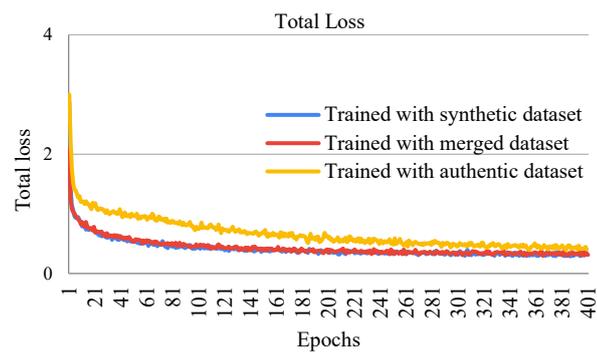


Fig. 12. Total loss

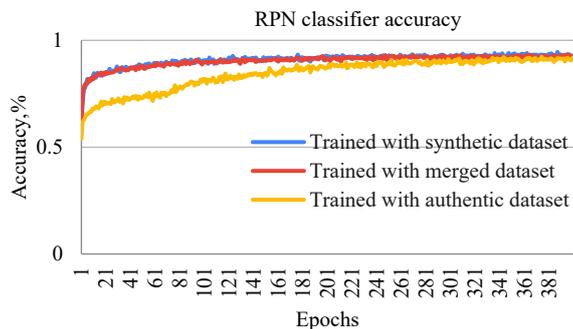


Fig. 13. RPN classifier accuracy

As seen in Fig. 8 to Fig. 13, all the datasets used to train the USB detector could create a convergent model. From Fig. 8 to Fig. 13, the USB detector trained using a synthetic dataset displayed better training performance, where the total loss was 0.337 instead of 0.492, which was the total loss resulting from model training using an authentic dataset. The same trend was also demonstrated by the USB detector trained with a merged dataset, whose results were also reasonably better than using an authentic dataset. A summary of the training performance parameters of USB detectors trained on different datasets is presented in Table VI.

Table VI also reveals that the utilization of synthetic images could increase RPN layer accuracy (92.6% for training with the synthetic dataset, 92.1% for training with the merged dataset) compared to using only authentic images (89.9%).

Regarding RPN classification loss, model training using synthetic and merged datasets provided smaller values (0.092

and 0.078, respectively) compared to using an authentic dataset (0.115). The trend was similar for RPN regression loss. The regression loss of synthetic and merged datasets was smaller (0.019 and 0.023, respectively) compared to the authentic dataset (0.249).

As with the loss in the RPN network, the loss during classifier network training also showed similar trend results. The classification loss and regression loss were smaller when the USB detector was trained using synthetic images compared to when trained with only authentic images. This causes the total loss for the proposed method to be smaller (0.336 and 0.348 for synthetic and merged datasets, respectively) than conventional training with only authentic images (0.492). Regarding the RPN layer accuracy as a training parameter, using synthetic images for training resulted in increased accuracies (92.6% for training with a synthetic dataset, 92.1% for training with the merged dataset) compared to using authentic images (89.9%).

TABLE VI. TRAINING PERFORMANCE OF USB DETECTORS TRAINED WITH DATASETS OF AUTHENTIC, SYNTHETIC, AND COMBINATION OF BOTH IMAGES

Parameter	Synthetic dataset	Merged dataset	Authentic dataset
RPN classifier accuracy	0.926	0.921	0.899
RPN classification loss	0.092	0.078	0.115
RPN regression loss	0.019	0.023	0.032
Classifier classification loss	0.180	0.193	0.249
Classifier regression loss	0.046	0.055	0.095
Total loss	0.337	0.348	0.492

B. Validation Test

Validation tests were performed to verify the functionality of the USB detector. Fig. 14 illustrates the detection results of each USB detector trained using a synthetic dataset Fig. 14(a), a merged dataset Fig. 14(b), and an authentic dataset Fig. 14(c). Moreover, Fig. 14 actually depicted an ESB.

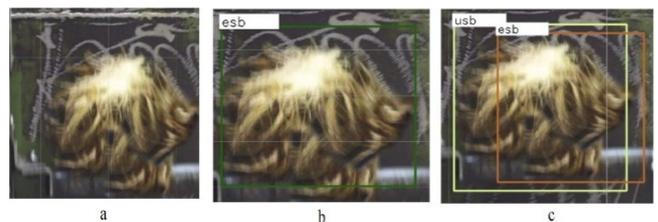


Fig. 14. Object detection accuracy per image during the validation test

Although the USB detector trained using the synthetic dataset failed to recognize an ESB Fig. 14(a), combining synthetic images with authentic images in a 1:1 ratio to form a merged dataset was able to improve the performance of the USB detector so that it could detect an ESB Fig. 14(b). In fact, it also corrected erroneous detection by the USB detector trained with authentic images as shown in Fig. 14(c).

Fig. 15 shows the results of the USB detector validation test. The validation test results revealed that the performance of USB detectors trained on authentic and synthetic datasets was nearly identical. However, the USB detector trained with a merged dataset had better performance.

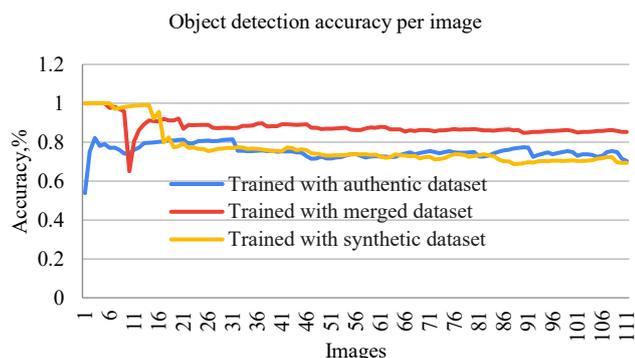


Fig. 15. Object detection accuracy per image during the validation test

Table VII lists the validation test results of USB detectors trained using different datasets. According to the results, a merged dataset used to train the USB detector resulted in the best results. A USB detector trained with a dataset of synthetic images gave a slightly lower mAP (75.3%) than authentic (real) images (77.1%). The accuracy was slightly smaller because only some of the detailed distinctive features of USB and EFB could be captured by PGGAN. The results of the training validation test using a merged dataset in a ratio of 1:1 gave a better mAP of 87.9%, or an increment of approximately 46% than the mAP obtained when using authentic images as training data.

TABLE VII. VALIDATION TEST RESULTS OF USB DETECTORS TRAINED WITH SYNTHETIC, MERGED AND AUTHENTIC DATASETS

Dataset	mAP	Difference (%)
Authentic	77.1%	Benchmark
Merged (1:1 ratio)	87.9%	10.1%
Synthetic	75.3%	-2.4%

VI. CONCLUSION

This paper proposes synthetic data images to train a VGG16-based Faster R-CNN to detect USB and EFB. The new data samples were obtained using PGGAN with the highest resolution of 512×512. The synthetic image-generating system using PGGAN could produce new images while maintaining the semantics of authentic images. Increasing the number and variety of sample images impacts the accuracy of USB and EFB detection. Combining authentic and synthetic datasets yielded the best detection results, increasing the ability to detect (mAP) by 10.1% compared to using only an authentic dataset.

Even though this research succeeded in showing an increase in object detection performance trained using additional data in the form of synthetic images, training only using synthetic images produced by PGGAN was not able to improve object detection performance; one of the reasons was because PGGAN was trained using a relatively small dataset. Using a larger dataset would enable PGGAN to extract USB and ESB features more comprehensively.

This research shows promising results for further study in real applications of USB monitoring in palm oil mills where the data obtained was videos rather than still images. Further research for generating synthetic data to train detection objects might utilize other GAN architectures, such as a

super-resolution GAN, since it could generate better-resolution images.

ACKNOWLEDGMENT

The authors thank UMPSA (Universiti Malaysia Pahang Al-Sultan Abdullah), Pahang, Malaysia and UAD (Universitas Ahmad Dahlan), Yogyakarta, Indonesia, for supporting this research through equipment and data.

REFERENCES

- [1] U. K. H. Mohd Nadzim, R. Yunus, R. Omar, and B. Y. Lim, "Factors Contributing to Oil Losses in Crude Palm Oil Production Process in Malaysia: A Review," *International Journal of Biomass and Renewables*, vol. 9, no. 1, pp. 10-24, 2020.
- [2] O. Walat and N. S. Bock, "Palm Oil Mill OER and Total Oil Losses," *Palm Oil Engineering Bulletin*, vol. 108, pp. 11-16, Sep. 2013.
- [3] Y. D. Tan and J. S. Lim, "Feasibility of palm oil mill effluent elimination towards sustainable Malaysian palm oil industry," *Renewable and Sustainable Energy Reviews*, vol. 111, pp. 507-522, 2019.
- [4] W. S. Aji, K. H. B. Ghazali, and S. A. Akbar, "Oil palm unstripped bunch detector using modified faster regional convolutional neural network," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 1, p. 189, 2022.
- [5] A. Hassan, N. H. Muhammad, Z. Ab Rahman, R. M. Halim, H. Alias, and M. Sabtu, "Improving mill oil extraction rate under the Malaysian National Key Economic Area," *Palm Oil Engineering Bulletin*, vol. 103, pp. 33-47, 2012.
- [6] J. W. Lai, H. R. Ramli, L. I. Ismail, and W. Z. W. Hasan, "Real-Time Detection of Ripe Oil Palm Fresh Fruit Bunch Based on YOLOv4," in *IEEE Access*, vol. 10, pp. 95763-95770, 2022, doi: 10.1109/ACCESS.2022.3204762.
- [7] A. Paullada, I. D. Raji, E. M. Bender, E. Denton, and A. Hanna, "Data and its (dis)contents: A survey of dataset development and use in machine learning research," *Patterns*, vol. 2, no. 11, p. 100336, Nov. 2021, doi: 10.1016/j.patter.2021.100336.
- [8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, May 2015, doi: 10.1038/nature14539.
- [9] J. Deng, W. Dong, R. Socher, L. -J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248-255, 2009, doi: 10.1109/CVPR.2009.5206848.
- [10] T. Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740-755, 2014.
- [11] A. Gupta, P. Dollár, and R. Girshick, "LVIS: A Dataset for Large Vocabulary Instance Segmentation," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5351-5359, 2019, doi: 10.1109/CVPR.2019.00550.
- [12] A. Kuznetsova *et al.*, "The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale," *Int. J. Comput. Vis.*, vol. 128, no. 7, pp. 1956-1981, Jul. 2020, doi: 10.1007/s11263-020-01316-z.
- [13] Y. Lu and S. Young, "A survey of public datasets for computer vision tasks in precision agriculture," *Comput. Electron. Agric.*, vol. 178, p. 105760, Nov. 2020, doi: 10.1016/j.compag.2020.105760.
- [14] A. Tomczak, A. Gupta, S. Ilic, N. Navab, and S. Albarqouni, "What can we learn about a generated image corrupting its latent representation?," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 505-515, 2022.
- [15] M. Abedi, L. Hempel, S. Sadeghi, and T. Kirsten, "GAN-Based Approaches for Generating Structured Data in the Medical Domain," *Appl. Sci.*, vol. 12, no. 14, p. 7075, Jul. 2022, doi: 10.3390/app12147075.
- [16] M. Jang *et al.*, "Image Turing test and its applications on synthetic chest radiographs by using the progressive growing generative adversarial network," *Sci. Rep.*, vol. 13, no. 1, p. 2356, Feb. 2023, doi: 10.1038/s41598-023-28175-1.

- [17] S. B. Alam, M. Hossain, and S. Kobashi, "Synthetic Brain Image Generation for ADHD prediction based on Progressive Growing Generative Adversarial Network," in *International Symposium on Affective Science and Engineering ISASE2020*, pp. 1-5, 2020.
- [18] C. Bowles *et al.*, "GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks," *arXiv preprint arXiv:1810.10863*, 2023.
- [19] K. C. Chan, M. Rabaev, and H. Pratama, "Generation of synthetic manufacturing datasets for machine learning using discrete-event simulation," *Prod. Manuf. Res.*, vol. 10, no. 1, pp. 337–353, Dec. 2022, doi: 10.1080/21693277.2022.2086642.
- [20] R. Sun, M. Zhang, K. Yang, and J. Liu, "Data Enhancement for Plant Disease Classification Using Generated Lesions," *Appl. Sci.*, vol. 10, no. 2, p. 466, Jan. 2020, doi: 10.3390/app10020466.
- [21] H. Bhagwani, S. Agarwal, A. Kodipalli, and R. J. Martis, "Targeting class imbalance problem using GAN," *2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT)*, pp. 318–322, 2021, doi: 10.1109/ICEECCOT52851.2021.9708011.
- [22] R. Ma, J. Lou, P. Li, and J. Gao, "Reconstruction of Generative Adversarial Networks in Cross Modal Image Generation with Canonical Polyadic Decomposition," *Wirel. Commun. Mob. Comput.*, vol. 2021, pp. 1–9, Apr. 2021, doi: 10.1155/2021/8868781.
- [23] B. Prisantono and H. Santoso, "CycleGAN and SRGAN to Enrich the Dataset," *Sinkron*, vol. 7, no. 2, pp. 495–503, Apr. 2022, doi: 10.33395/sinkron.v7i2.11384.
- [24] M. Yurt, S. U. Dar, A. Erdem, E. Erdem, K. K. Oguz, and T. Çukur, "mustGAN: multi-stream Generative Adversarial Networks for MR Image Synthesis," *Medical image analysis*, vol. 70, p. 101944, May 2021, doi: 10.1016/j.media.2020.101944.
- [25] A. Gonzales, G. Guruswamy, and S. R. Smith, "Synthetic data in health care: A narrative review," *PLOS Digital Health*, vol. 2, no. 1, p. e0000082, Jan. 2023, doi: 10.1371/journal.pdig.0000082.
- [26] S. Kumari and K. Aggarwal, "Scope of generative adversarial networks (GANs) in image processing: A review," *Int. J. Health Sci.*, vol. 6, no. S6, pp. 724–733, Jun. 2022, doi: 10.53730/ijhs.v6nS6.9664.
- [27] D. Mukherjee, P. Saha, D. Kaplun, A. Sinitca, and R. Sarkar, "Brain tumor image generation using an aggregation of GAN models with style transfer," *Sci. Rep.*, vol. 12, no. 1, p. 9141, Jun. 2022, doi: 10.1038/s41598-022-12646-y.
- [28] J. Islam and Y. Zhang, "GAN-based synthetic brain PET image generation," *Brain Inform.*, vol. 7, no. 1, p. 3, Dec. 2020, doi: 10.1186/s40708-020-00104-2.
- [29] A. Aljohani and N. Alharbi, "Generating Synthetic Images for Healthcare with Novel Deep Pix2Pix GAN," *Electronics*, vol. 11, no. 21, p. 3470, Oct. 2022, doi: 10.3390/electronics11213470.
- [30] L. Abady, J. Horváth, B. Tondi, E. J. Delp, and M. Barni, "Manipulation and generation of synthetic satellite images using deep learning models," *J. Appl. Remote Sens.*, vol. 16, no. 4, Nov. 2022, doi: 10.1117/1.JRS.16.046504.
- [31] H. Mansourifar, A. Moskovitz, B. Klingensmith, D. Mintas, and S. J. Simske, "GAN-Based Satellite Imaging: A Survey on Techniques and Applications," *IEEE Access*, vol. 10, pp. 118123–118140, 2022, doi: 10.1109/ACCESS.2022.3221123.
- [32] P. Gao, T. Tian, L. Li, J. Ma, and Y. Tian, "DE-CycleGAN: An Object Enhancement Network for Weak Vehicle Detection in Satellite Images," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 3403–3414, 2021, doi: 10.1109/JSTARS.2021.3062057.
- [33] X. Rui, Y. Cao, X. Yuan, Y. Kang, and W. Song, "DisasterGAN: Generative Adversarial Networks for Remote Sensing Disaster Image Generation," *Remote Sens.*, vol. 13, no. 21, p. 4284, Oct. 2021, doi: 10.3390/rs13214284.
- [34] H. Liu, G. Yang, F. Deng, Y. Qian, and Y. Fan, "MCBAM-GAN: The Gan Spatiotemporal Fusion Model Based on Multiscale and CBAM for Remote Sensing Images," *Remote Sens.*, vol. 15, no. 6, p. 1583, Mar. 2023, doi: 10.3390/rs15061583.
- [35] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, and J. Jiang, "Edge-Enhanced GAN for Remote Sensing Image Superresolution," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5799–5812, Aug. 2019, doi: 10.1109/TGRS.2019.2902431.
- [36] A. Abdollahi, B. Pradhan, G. Sharma, K. N. A. Maulud, and A. Alamri, "Improving Road Semantic Segmentation Using Generative Adversarial Network," *IEEE Access*, vol. 9, pp. 64381–64392, 2021, doi: 10.1109/ACCESS.2021.3075951.
- [37] A. Abdollahi, B. Pradhan, S. Gite, and A. Alamri, "Building Footprint Extraction from High Resolution Aerial Images Using Generative Adversarial Network (GAN) Architecture," *IEEE Access*, vol. 8, pp. 209517–209527, 2020, doi: 10.1109/ACCESS.2020.3038225.
- [38] H. Alqahtani, M. Kavakli-Thorne, and G. Kumar, "Applications of Generative Adversarial Networks (GANs): An Updated Review," *Arch. Comput. Methods Eng.*, vol. 28, no. 2, pp. 525–552, Mar. 2021, doi: 10.1007/s11831-019-09388-y.
- [39] M. B. Bejiga, G. Hoxha, and F. Melgani, "Improving Text Encoding for Retro-Remote Sensing," in *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 4, pp. 622–626, April 2021, doi: 10.1109/LGRS.2020.2983851.
- [40] M. Fuentes Reyes, S. Auer, N. Merkle, C. Henry, and M. Schmitt, "SAR-to-Optical Image Translation Based on Conditional Generative Adversarial Networks—Optimization, Opportunities and Limits," *Remote Sens.*, vol. 11, no. 17, p. 2067, Sep. 2019, doi: 10.3390/rs11172067.
- [41] H. Gao *et al.*, "A Hyperspectral Image Classification Method Based on Multi-Discriminator Generative Adversarial Networks," *Sensors*, vol. 19, no. 15, p. 3269, Jul. 2019, doi: 10.3390/s19153269.
- [42] R. Qin and J. Zhao, "High-Efficiency Generative Adversarial Network Model for Chemical Process Fault Diagnosis," *IFAC-Pap.*, vol. 55, no. 7, pp. 732–737, 2022, doi: 10.1016/j.ifacol.2022.07.531.
- [43] P. Peng, Y. Wang, W. Zhang, Y. Zhang, and H. Zhang, "Imbalanced Process Fault Diagnosis Using Enhanced Auxiliary Classifier GAN," *2020 Chinese Automation Congress (CAC)*, pp. 313–316, 2020, doi: 10.1109/CAC51589.2020.9327104.
- [44] X. Hu, G. Li, P. Niu, J. Wang, and L. Zha, "A generative adversarial neural network model for industrial boiler data repair," *Appl. Soft Comput.*, vol. 104, p. 107214, Jun. 2021, doi: 10.1016/j.asoc.2021.107214.
- [45] J. D. C. Mumbelli, G. A. Guarneri, Y. K. Lopes, D. Casanova, and M. Teixeira, "An application of Generative Adversarial Networks to improve automatic inspection in automotive manufacturing," *Appl. Soft Comput.*, vol. 136, p. 110105, Mar. 2023, doi: 10.1016/j.asoc.2023.110105.
- [46] J. Cao, J. Ma, D. Huang, P. Yu, J. Wang, and K. Zheng, "Method to enhance deep learning fault diagnosis by generating adversarial samples," *Appl. Soft Comput.*, vol. 116, p. 108385, Feb. 2022, doi: 10.1016/j.asoc.2021.108385.
- [47] J. E. Choi, D. H. Seol, C. Y. Kim, and S. J. Hong, "Generative Adversarial Network-Based Fault Detection in Semiconductor Equipment with Class-Imbalanced Data," *Sensors*, vol. 23, no. 4, p. 1889, Feb. 2023, doi: 10.3390/s23041889.
- [48] L. Huo, H. Qi, S. Fei, C. Guan, and J. Li, "A Generative Adversarial Network Based a Rolling Bearing Data Generation Method Towards Fault Diagnosis," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–21, Jul. 2022, doi: 10.1155/2022/7592258.
- [49] Y. Lu, D. Chen, E. Olaniyi, and Y. Huang, "Generative adversarial networks (GANs) for image augmentation in agriculture: A systematic review," *Comput. Electron. Agric.*, vol. 200, p. 107208, Sep. 2022, doi: 10.1016/j.compag.2022.107208.
- [50] E. Olaniyi, D. Chen, Y. Lu, and Y. Huang, "Generative adversarial networks for image augmentation in agriculture: a systematic review," *arXiv preprint arXiv:2204.04707*, 2022.
- [51] J. J. Bird, C. M. Barnes, L. J. Manso, A. Ekárt, and D. R. Faria, "Fruit quality and defect image classification with conditional GAN data augmentation," *Sci. Hortic.*, vol. 293, p. 110684, Feb. 2022, doi: 10.1016/j.scienta.2021.110684.
- [52] H. Madokoro *et al.*, "Semantic Segmentation of Agricultural Images Based on Style Transfer Using Conditional and Unconditional Generative Adversarial Networks," *Appl. Sci.*, vol. 12, no. 15, p. 7785, Aug. 2022, doi: 10.3390/app12157785.
- [53] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN Comput. Sci.*, vol. 2, no. 6, p. 420, Nov. 2021, doi: 10.1007/s42979-021-00815-1.

- [54] K. Choudhary *et al.*, "Recent advances and applications of deep learning methods in materials science," *Npj Comput. Mater.*, vol. 8, no. 1, p. 59, Apr. 2022, doi: 10.1038/s41524-022-00734-6.
- [55] X. Li, M. Luo, S. Ji, L. Zhang, and M. Lu, "Evaluating generative adversarial networks based image-level domain transfer for multi-source remote sensing image segmentation and object detection," *Int. J. Remote Sens.*, vol. 41, no. 19, pp. 7343–7367, Oct. 2020, doi: 10.1080/01431161.2020.1757782.
- [56] S. M. A. Bashir and Y. Wang, "Small Object Detection in Remote Sensing Images with Residual Feature Aggregation-Based Super-Resolution and Object Detector Network," *Remote Sens.*, vol. 13, no. 9, p. 1854, May 2021, doi: 10.3390/rs13091854.
- [57] C. D. Prakash and L. J. Karam, "It GAN Do Better: GAN-Based Detection of Objects on Images With Varying Quality," in *IEEE Transactions on Image Processing*, vol. 30, pp. 9220–9230, 2021, doi: 10.1109/TIP.2021.3124155.
- [58] L. Posilović, D. Medak, M. Subašić, M. Budimir, and S. Lončarić, "Generative adversarial network with object detector discriminator for enhanced defect detection on ultrasonic B-scans," *Neurocomputing*, vol. 459, pp. 361–369, Oct. 2021, doi: 10.1016/j.neucom.2021.06.094.
- [59] R. Liu *et al.*, "Anomaly-GAN: A data augmentation method for train surface anomaly detection," *Expert Syst. Appl.*, vol. 228, p. 120284, Oct. 2023, doi: 10.1016/j.eswa.2023.120284.
- [60] J. Zhang *et al.*, "Shrimp egg counting with fully convolutional regression network and generative adversarial network," *Aquac. Eng.*, vol. 94, p. 102175, Aug. 2021, doi: 10.1016/j.aquaeng.2021.102175.
- [61] H. Lee, S. Kang, and K. Chung, "Robust Data Augmentation Generative Adversarial Network for Object Detection," *Sensors*, vol. 23, no. 1, p. 157, Dec. 2022, doi: 10.3390/s23010157.
- [62] R. Barth, J. Hemming, and E. J. Van Henten, "Optimising realism of synthetic images using cycle generative adversarial networks for improved part segmentation," *Comput. Electron. Agric.*, vol. 173, p. 105378, Jun. 2020, doi: 10.1016/j.compag.2020.105378.
- [63] E. Bellocchio, G. Costante, S. Cascianelli, M. L. Fravolini, and P. Valigi, "Combining Domain Adaptation and Spatial Consistency for Unseen Fruits Counting: A Quasi-Unsupervised Approach," in *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1079–1086, April 2020, doi: 10.1109/LRA.2020.2966398.
- [64] Z. Fei, A. Olenskyj, B. N. Bailey, and M. Earles, "Enlisting 3D Crop Models and GANs for More Data Efficient and Generalizable Fruit Detection," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 1269–1277, 2021, doi: 10.1109/ICCVW54120.2021.00147.
- [65] B. N. Bailey, "Helios: A Scalable 3D Plant and Environmental Biophysical Modeling Framework," *Front. Plant Sci.*, vol. 10, p. 1185, Oct. 2019, doi: 10.3389/fpls.2019.01185.
- [66] C. Douarre, C. F. Crispim-Junior, A. Gelibert, L. Tougne, and D. Rousseau, "Novel data augmentation strategies to boost supervised segmentation of plant disease," *Comput. Electron. Agric.*, vol. 165, p. 104967, Oct. 2019, doi: 10.1016/j.compag.2019.104967.
- [67] M. Zeng, H. Gao, and L. Wan, "Few-Shot Grape Leaf Diseases Classification Based on Generative Adversarial Network," *J. Phys. Conf. Ser.*, vol. 1883, no. 1, p. 012093, Apr. 2021, doi: 10.1088/1742-6596/1883/1/012093.
- [68] D. M. Ibrahim and N. M. Elshennawy, "Improving Date Fruit Classification Using CycleGAN-Generated Dataset," *Comput. Model. Eng. Sci.*, vol. 131, no. 1, pp. 331–348, 2022, doi: 10.32604/cmescs.2022.016419.
- [69] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with C-GAN synthetic images," *Comput. Electron. Agric.*, vol. 187, p. 106279, Aug. 2021, doi: 10.1016/j.compag.2021.106279.
- [70] Y. Tian, G. Yang, Z. Wang, E. Li, and Z. Liang, "Detection of Apple Lesions in Orchards Based on Deep Learning Methods of CycleGAN and YOLOV3-Dense," *J. Sens.*, vol. 2019, pp. 1–13, Apr. 2019, doi: 10.1155/2019/7630926.
- [71] Z. Luo, H. Yu, and Y. Zhang, "Pine Cone Detection Using Boundary Equilibrium Generative Adversarial Networks and Improved YOLOv3 Model," *Sensors*, vol. 20, no. 16, p. 4430, Aug. 2020, doi: 10.3390/s20164430.
- [72] C. Dewi, R.-C. Chen, Y.-T. Liu, and S.-K. Tai, "Synthetic Data generation using DCGAN for improved traffic sign recognition," *Neural Comput. Appl.*, vol. 34, no. 24, pp. 21465–21480, Dec. 2022, doi: 10.1007/s00521-021-05982-z.
- [73] G. Hu, H. Wu, Y. Zhang, and M. Wan, "A low shot learning method for tea leaf's disease identification," *Comput. Electron. Agric.*, vol. 163, p. 104852, Aug. 2019, doi: 10.1016/j.compag.2019.104852.
- [74] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.
- [75] X. Guo, Y. Liu, K. Tan, M. Jin, and H. Lu, "PGGAN: Improve Password Cover Rate Using the Controller," *J. Phys. Conf. Ser.*, vol. 1856, no. 1, p. 012012, Apr. 2021, doi: 10.1088/1742-6596/1856/1/012012.
- [76] A. A. Rusu *et al.*, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.
- [77] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed. Tools Appl.*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023, doi: 10.1007/s11042-022-13644-y.
- [78] W. Liu *et al.*, "Ssd: Single shot multibox detector," in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 21–37, 2016.
- [79] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digit. Signal Process.*, vol. 126, p. 103514, Jun. 2022, doi: 10.1016/j.dsp.2022.103514.
- [80] J. P. Vasconez, J. Delpiano, S. Vougioukas, and F. Auat Cheein, "Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation," *Comput. Electron. Agric.*, vol. 173, p. 105348, Jun. 2020, doi: 10.1016/j.compag.2020.105348.
- [81] S. Wan and S. Goudos, "Faster R-CNN for multi-class fruit detection using a robotic vision system," *Comput. Netw.*, vol. 168, p. 107036, Feb. 2020, doi: 10.1016/j.comnet.2019.107036.
- [82] Z. Wang *et al.*, "An improved Faster R-CNN model for multi-object tomato maturity detection in complex scenarios," *Ecol. Inform.*, vol. 72, p. 101886, Dec. 2022, doi: 10.1016/j.ecoinf.2022.101886.
- [83] Z. Li *et al.*, "A high-precision detection method of hydroponic lettuce seedlings status based on improved Faster RCNN," *Comput. Electron. Agric.*, vol. 182, p. 106054, Mar. 2021, doi: 10.1016/j.compag.2021.106054.
- [84] L. Quan *et al.*, "Maize seedling detection under different growth stages and complex field environments based on an improved Faster R-CNN," *Biosyst. Eng.*, vol. 184, pp. 1–23, Aug. 2019, doi: 10.1016/j.biosystemseng.2019.05.002.
- [85] Z. Song, L. Fu, J. Wu, Z. Liu, R. Li, and Y. Cui, "Kiwifruit detection in field images using Faster R-CNN with VGG16," *IFAC-Pap.*, vol. 52, no. 30, pp. 76–81, 2019, doi: 10.1016/j.ifacol.2019.12.500.
- [86] M. M. Taye, "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions," *Computation*, vol. 11, no. 3, p. 52, Mar. 2023, doi: 10.3390/computation11030052.
- [87] G. Iyer, B. Hanin, and D. Rolnick, "Maximal initial learning rates in deep relu networks," in *International Conference on Machine Learning*, pp. 14500–14530, 2023.