# ANALYSIS OF FOUR GENERATOR ARCHITECTURES OF C-GAN, LOSS FUNCTION, AND ANNOTATION METHOD FOR EPIPHYTE IDENTIFICATION

V.V. Sajithvariyar [1,*], S Aswin[1] V Sowmya [1], K.P. Soman [1], R. Sivanpillai, [2], G. K. Brown [3]

[1] Center for Computational Engineering and Networking, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, TN 641 112, India (ORCID: 0000-0003-3944-8155) - vv_sajithvariyar@cb.amrita.edu, aswins2010@gmail.com, v_sowmya@cb.amrita.edu, kp_soman@amrita.edu
[2] Wyoming GIS Center, University of Wyoming, Laramie, WY 82072, USA (ORCID: 0000-0003-3547-9464) – sivan@uwyo.edu
[3] Department of Botany, University of Wyoming, Laramie, WY 82072, USA – GKBrown@uwyo.edu

**KEY WORDS: Deep Learning, GAN, Generator Network, Loss Function, SSIM, IoU**

**ABSTRACT:**

The deep learning (DL) models require timely updates to continue their reliability and robustness in prediction, classification, and segmentation tasks. When the deep learning models are tested with a limited test set, the model will not reveal the drawbacks. Every deep learning baseline model needs timely updates by incorporating more data, change in architecture, and hyper parameter tuning. This work focuses on updating the Conditional Generative Adversarial Network (C-GAN) based epiphyte identification deep learning model by incorporating 4 different generator architectures of GAN and two different loss functions. The four generator architectures used in this task are Resnet-6. Resnet-9, Resnet-50 and Resnet-101. A new annotation method called background removed annotation was tested to analyse the improvement in the epiphyte identification protocol. All the results obtained from the model by changing the above parameters are reported using two common evaluation metrics. Based on the parameter tuning experiment, Resnet-6, and Resnet-9, with binary cross-entropy (BCE) as the loss function, attained higher scores also Resnet-6 with MSE as loss function performed well. The new annotation by removing the background had minimal effect on identifying the epiphytes.

## 1. INTRODUCTION

Neural network (NN) algorithms are used in many digital data analysis (Tefas et.al., 2013). Advancements in computational hardware, storage and software are fueling progress in digital data analysis. Deep learning-based data analysis are a part of NN algorithms and are robust for applications with data generated by numerous sources (Jia et.al., 2017 and Najafabadi et.al., 2015). These deep learning (DL) algorithms are capable of understanding data and its pattern from an experiential learning and derive the features from input data and generate learned models (Harshvardhan et.al., 2020). The performance of DL algorithms are highly dependent on the quantity and quality of data used for learning and its mathematical modelling.

DL algorithms are used for prediction and classification tasks in several disciplines (Rory et.al., 2019; Iqbal et.al., 2019). DL algorithms consist of deep neural network components and their organisation collectively referred as their architecuture. Several state of the art DL architectures are used for image classification, object detection, and image segmentation tasks (Zhao et.al., 2019; Nida et.al., 2015).

Performance of DL or any NN algorithms varies over time when the requirements change. Changes to the structure, parameters, and mathematical modelling of DL architecture are necessary for improving their performance. There are several DL architectures like VGG16, GoogleNet, ImageNet etc (Chen et al, 2018; Szegedy et.al., 2015 and Krizhevsky et.al., 2012) used for various image processing applications. The DL algorithms are not self-adaptive to the new requirements and sometimes they are computationally intensive. Hence updated concepts to DL and other NN algorithms are necessary for better learning and effective utilisation of computational resources. There are many examples like introduction of different types of convolutions in convolutional NN algorithms (Ding et.al., 2018) to improve the feature extraction with reduced computational cost. The DL architectures like Unet (Ronneberger et.al., 2015) and generative adversarial networks (GAN) (Goodfellow et.al., 2014) are specifically designed for image-to-image translation by innovative architectural components. The deep learning algorithm updates in their components aims to produce better output, improved performance and effective utilization of computational resources.

Shashank et.al., (2020) used Conditional-GAN algorithm (C-GAN) introduced by Philip et.al., (2017) for identifying the epiphytes (*Werauhia kupperiana*) in the drone acquired images. That study modelled the target identification task as an image-to-image translation problem and applied adversarial concepts. C-GAN identified 80% of the epiphytes from the test set. This study was not able to produce good output labels in many scenarios. Also, the algorithm was not able to perform well when the target plant differs in distance at which target is imaged, lighting conditions and distance at which the target is imaged. This motivated the current study to experiment and explore the performance of C-GAN algorithms by changing its hyper parameters and architecture components to improve the performance of the algorithm.

This study builds on the work completed by Shashank et al. (2020). First, a *new annotation method* was used for generating the images needed to train the algorithm. Next, the performance

* Corresponding author

four *generator architectures* were evaluated with two different *loss functions*. The objective of this study is to update the DL-based epiphyte identification model used by Shashank et al., (2020). The proposed model underwent an architectural change to the existing C-GAN (Philip et.al., 2017). Results from this study will provide valuable information for developing more robust DL algorithms to identify epiphytes in digital images.

## 2. MATERIALS AND METHODS

The experiment were organized in two stages where binary cross entropy (Shie et.al., 2005) was set as the loss function with 4 different residual networks. In the second stage all residual networks were coupled with MSE (Zahra et al., 2014) as the loss function. The above experiments trained the C-GAN model with the epiphyte dataset. The phython program trained the model for 200 epochs and saved the final model to the local system. The testing was done separately with 12 images which are not seen by the network during the training phase.

### 2.1 Epiphyte dataset and annotation

The epiphyte dataset used in this study was acquired in Costa Rica (Sajithvariyar et.al., 2019). The epiphyte dataset consisted of 115 drone-aquired images of which 98 were used for training, 12 for testing and remaining 5 for validation. The validation set were used during training to tune the parameters and the test set were used to assess the trained network's ability to identify the target plant.

The C-GAN architecture was implemented using python and trained in an i7 processor with 8 GB RAM and NVIDIA Quadro P5000 GPU. All the input images and labels had a dimension of 256 x 256 x 3.

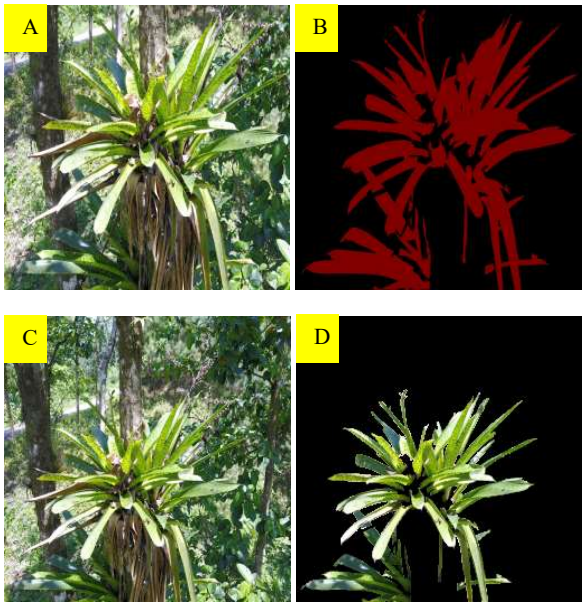



Figure 1. The input image (A) and corresponding annotation image (B) generated by the first study. The input image (C) and the annotation image (D) generated by the present study.

The new annotation method removed the background black pixels and kept the target as a true colour images. In the earlier study, the target plant was identified as red pixels (Figure 1). The annotation images generated by Shashank et al., (2020) were used to remove the background pixels and retain the original, true-colour image of the target plant (Figure 1). Retaining the RGB values of the target plant will help C-GAN to concentrate more on the foreground pixel and generate better output labels.

### 2.2 CGAN Generator and Discriminator

The C-GAN deep learning architecture used for epiphyte identification consists of two competing networks called *generators* and *discriminator* with two loss functions. The previous study used the UNET encoder-decoder architecture (Ronneberger et.al., 2015). The encoders will map the input data to a lower dimension and the decoder maps this back to original information. The dimensionality reduced data will be scale invariant and translation invariant which is very important for object identification tasks. C-GAN is a variant of GAN where it enforces the input data to derive the features based on a condition, which is our annotated image. This conditional enforcing will ensure that the algorithm will focus on the target region in the input image by referring to the annotation images.

In the present study, we evaluated four different types of deep convolutional neural architecture for the generator networks. The following subsection gives the details of the deep neural networks used for constructing the generator network.

#### 2.2.1 Residual Generator Networks

The Generator networks architecture was replaced with a deep convolutional neural architecture called *Residual networks* (He et.al., 2016). The residual networks were designed by a group of researchers in Microsoft during 2015. The major contribution of the network is to remove the vanishing / exploding gradient problem in deep networks. The residual networks are implemented with shortcut connections between layers and exhibit efficient training from residual functions as shown in (Figure 2).

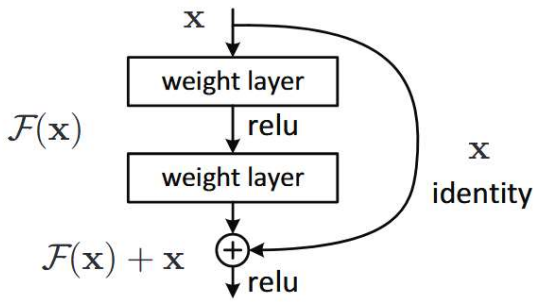



Figure 2. The Residual network learning residual functions by referring input layers (He et.al., 2016).

The residual networks are designed in various layer depths and they are named in such a way that the decimal number indicates the layer depth. In this study we tested four different variants of the Resnet architectures after replacing the C-GAN generator with following variants a) Resnet-6, b) Resnet-9, c) Resnet-50, and d) Resnet-101.

The generator network plays an important role in C-GAN algorithms. The generator network is responsible for generating the fake sample by looking at the conditional parameter that is the annotated image. The performance of generator is good when it can produce fake samples as close to the original images. At this stage, the discriminator network will fail to differentiate the

original and from the fake samples. Since this study is mainly focused on the generator networks, we retained the original discriminator network that was used in the previous study. The discriminator network is a patchGAN architecture with a patch size of  which is the best window size experimentally proved by (Isola et.al., 2017).

Most of the studies and research in deep convolutional networks states that "the deeper the better" (Bekele et.al., 2019). Hence we attempted to improve the performance of deep CNN networks by adding more layers. On the other hand, by making the networks deeper the computational cost and run time will increase. The major issue in a deep neural network is the vanishing Gradient problem. This occurs when the network is not able to learn anything from the input data. When the network is too deep, the gradients from the loss functions will map the values to zeros. This will result in no further updates to the weight matrix of the model and the learning rate of the network will decline. To overcome this huddle we need an architecture which is deeper but also free from the vanishing gradient problem.

The residual networks enforce efficient learning by mapping residual functions between layers and there improve the training process. The residual network learns the residuals to match the input with the predicted weights. This process makes sure that the deeper networks will learn better without degrading the process.

### 2.3 The CGAN loss functions

The loss functions are vital in any neural network training to keep track of the model's learning of the weights. The proposed methods consist of a binary classification where the C-GAN must classify the background and target pixels. Generator loss depends on the ability of the discriminator to identify fake as real samples. Discriminator loss penalizes itself for misclassifying a real instance as fake and vice versa. In this study, we used two loss functions a) Mean Squared Error (MSE) b) Binary cross Entropy (BCE) for the generator network. The experiments were conducted for all four generator networks with two loss functions. The output generated by 4 different generator networks with two loss functions are reported using structural similarity index (SSIM) and intersection over union score (IoU).

## 3. RESULT AND DISCUSSION

The SSIM and IoU scores were computed for the predicted label and ground truth label from different models trained in this study. The SSIM will look for the structural similarity and IoU will find the maximum overlap between the predicted labels and ground truth labels. A Python script was developed to iteratively compute the IoU and SSIM score for all the test images and their average.

The new annotation method proposed in this study contributed more towards the predicted label analysis for the analyst. The new annotation images are not contributing anything new while comparing them to the annotation used in the previous study. This reveals that masked annotation with false color will be sufficient for epiphyte identification task.

Analyses of the output images generated by various models trained with the new annotation method revealed no major difference in generated output labels. The major advantage for analyst with the new annotation is that after predicting the labels it is easy to understand the portion of the epiphyte where the model failed to predict. The effect of loss function on predicted labels like blurring is evident from the annotation. This also helped to understand that the prediction on epiphyte leaf edges and overlapped leaves are more blurred.

Replacing the generator networks with residual networks and two generator loss functions MSE and BCE generated different output labels. Table 1 summarizes the results obtained from 4 different residual networks with BCE and MSE loss functions. From the results obtained, generator networks with Resnet-6, Resnet-9 and BCE as the generator loss function scored maximum IoU and SSIM score. Also, from Table 1 it is evident that when MSE was set as the loss function Resnet-6 generated output labels with high SSIM and IoU score. The Resnet-50 and -101 underperformed due to a smaller number of training samples. The deeper the networks, the more data required for training. The Resnet-50 consist of 50 layers and Resnet-101 consist of 101 layers, when this many layers iterate over fewer number of training samples there will not be further improvement in training. The performance will be degraded, and the network will generate a poor model. This resulted in generating poor output labels which gives low SSIM, and IoU scores compared to ground truth.

| Average SSIM and IoU score for 4 different architecture with BCE as loss function | | | | |
|---|---|---|---|---|
| | **Resnet-6** | **Resnet-9** | **Resnet-50** | **Resnet-101** |
| SSIM | **0.60** | **0.61** | 0.56 | 0.56 |
| IoU | **0.38** | **0.41** | 0.27 | 0.25 |
| Average SSIM and IoU score for 4 different architecture with MSE as loss function | | | | |
| | **Resnet-6** | **Resnet-9** | **Resnet-50** | **Resnet-101** |
| SSIM | **0.74** | **0.60** | 0.60 | 0.60 |
| IoU | **0.56** | **0** | 0 | 0 |

Table 1. The average SSIM and IoU score computed for the residual networks with two different loss functions.

The SSIM scores obtained with Resnet-6 and MSE as the loss function was higher for Resnet-6 and remained the same for the remaining networks (Table 1). These scenarios indicate the limitations of SSIM scores for evaluating the output labels. The SSIM score looks for the maximum similarity between the predicted and ground truth labels. In this study all label images consisted of two classes: 1) all non-target pixels belong to background (black), and 2) the target pixels in its original colour space. Also, in many test images the target plant occupied a small area in each frame compared to background information. Under these circumstances, though the model fails to correctly predict the target plant, the similarities in the background pixels will lead to higher SSIM scores (Figure 3).

The IoU scores associated with Resnet-6 and MSE loss function was 0.56 (Table 1) and was 0 for the rest (Table 1). The IoU is a method to quantify the percentile of overlap between the predicted label and ground truth label. IoU metric measures the number of pixels common between the ground truth and prediction label divided by the total number of pixels present across both labels. The value of IoU score is ranging between 0 and 1 where value close to 1 indicate predicted label is closer to ground truth and 0 indicate they are dissimilar.


Figure 3. The high SSIM score of an input image (A) with minimal target occupancy and incorrectly predicted label (B) due to high overlap between black regions.

The SSIM limitations can be easily replaced while computing the IoU score. The SSIM score is helpful when we need to evaluate the predicted label with some epiphyte pixels and compare the structural similarity. This shows that SSIM score along with IoU gives a better clarity on the output labels predicted.

Results obtained from various models evaluated in this study shows that the output predicted labels are more blurred when we have BCE as the loss function (Table 2). The model trained with Resnet-6 and MSE loss produce sharp images with less blurring effect. The objective of the BCE loss function is to reduce the error to zero. This results in blurring effect. Unlike BCE, MSE computes the error based on the squared distance. This results in less blurring effect, when compared to the output obtained using BCE loss function.

Table 2 gives some sample output labels with high IoU and SSIM scores predicted by Resnet-6 and 9 with BCE and Resnet-6 with MSE as the loss function.

The results obtained from the experiments also reveals that when the network is going deeper from number of layers 6 to 101 in residual networks the SSIM and IoU scores are declined. The deeper the networks a greater number of images is required for training. This also shows that the potential of improving the results with more training data.

## 4. CONCLUSIONS AND RECOMMENDATIONS

Resnet-6 and Resnet-9 with BCE and MSE loss functions were able to generate output labels with higher SSIM and IoU scores. The SSIM scores can be higher if the target plant occupies a small area in the images used for testing. Resnet-50 and Resnet-101 did yield output labels with lower SSIM and IoU scores due to smaller number of images for training.

The output labels were more blurred for BCE compared to MSE loss function. Choice of selecting appropriate loss function for reducing the blur in the output labels. Current DL architecture demands more changes in the system loss function.

The new annotation by removing the background had no significant improvement in label prediction.

Incorporation of hybrid models might be necessary to make improvements to the epiphyte identification model. This work also highlights the opportunities for further improvement by making changes to hyper parameters like loss function in addition to incorporating new architectures.

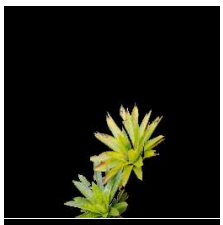
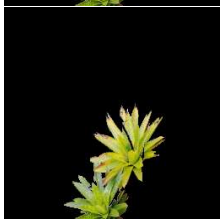
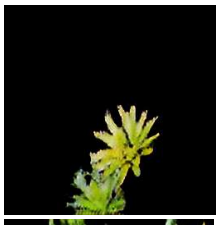
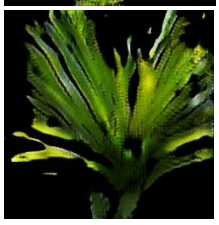
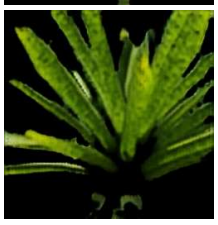
| Generator network & Loss function | Ground Truth | Predicted |
|---|---|---|
| Resnet-6 & BCE |  |  |
| Resnet-9 & BCE |  |  |
| Resnet-6 & MSE |  |  |
| Resnet-6 & BCE |  |  |
| Resnet-9 & BCE |  |  |
| Resnet-6 & MSE |  |  |

Table 2. The labels predicted by the trained model with highest SSIM and IoU score.

## ACKNOWLEDGEMENTS

UAV images used in this study were acquired through a seed grant from the UW College of Arts & Sciences. We thank Dr. Mario Blanco, University of Costa Rica, and Dr. Carlos de la Rosa, Director, La Selva Research Station for logistic support.

Authors thank Dr. Poornachandran, Centre for Cyber Security Systems, Amritapuri Campus, Kollam, Kerala for supporting the computational platforms. We thank Mr. Shashank Anivila for the initial implementation of epiphyte identification model.

## REFERENCES

Bekele, E., Lawson, W., 2019: The Deeper, the Better: Analysis of Person Attributes Recognition,*14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France*, pp. 1-8. doi.org/10.1109/FG.2019.8756526.

Chen, L., Zhu, L., Papandreou, G., Schroff, G., and Adam, H., 2018: Encoder-decoder with atrous separable convolution for semantic image segmentation. *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 801-818. doi.org/ doi.org/10.1007/978-3-030-01234-2_49.

Ding-Xuan., Zhou., 2019: Universality of deep convolutional neural networks. *In Applied and Computational Harmonic Analysis*. doi.org/10.1016/j.acha.2019.06.004.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014: Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680). doi.org/10.5555/2969033.2969125.

Harshvardhan, G.M., Mahendra, K.G., Manjusha, P., Siddharth S.R., 2020: A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*, Volume 38, 100285, ISSN 1574-0137, doi.org/10.1016/j.cosrev.2020.100285.

He, K., Zhang, X., Ren, S., and Sun, J., 2016: Deep residual learning for image recognition, Proc. *IEEE Conf. Comput. Vis. Pattern Recognition*, pp. 770-778. doi.org/ 10.1109/CVPR.2016.90

Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A., 2017: Image-to-image translation with conditional adversarial networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134). doi.org/10.1109/CVPR.2017.632.

Jia, X., 2017: Image recognition method based on deep learning, *29th Chinese Control And Decision Conference (CCDC), Chongqing, China*, pp. 4730-4735. doi.org/ 10.1109/CCDC.2017.7979332.

Krizhevsky, A., Sutskever, I., Hinton, H., 2012: Imagenet classification with deep convolutional neural networks. *Proc. Adv. Neural Inf. Process. Syst*., pp. 1097-1105.

Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., 2015. Deep learning applications and challenges in big data analytics. *Journal of Big Data 2*, 1. doi.org/10.1186/s40537-014-0007-7

Nirmal, S., Sowmya, V., Soman K.P., 2020: Open Set Domain Adaptation for Hyperspectral Image Classification Using Generative Adversarial Network. *In Lecture Notes in Networks and Systems.* doi.org/10.1007/978-981-15-0146-3_78.

Patil, S.O., Sajithvariyar, V.V., Soman, K.P., 2020: Speed Bump Segmentation an Application of Conditional Generative Adversarial Net- work for Self-driving Vehicles. In *2020 Fourth International Conference on Computing Methodologies and Communication.* doi.org/ 10.1109/ICCMC48092.2020.ICCMC-000173.

Ronneberger, O., Fischer, P., Brox, T., 2015: U-Net: Convolutional networks for biomedical image segmentation. *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.,* pp. 234-241.

Rory, P.B., Fadi, T., 2019: A machine learning framework for sport result prediction. *In Applied Computing and Informatics,* 15(1), pp. 27-33, ISSN 2210-8327.

Sajithvariyar, V.V., Sowmya, V., Gopalakrishnan, E.A., Soman, K.P., Bupathy, P., Sivanpillai, R., & Brown, G.K., 2019: Opportunities and challenges of launching UAVs within wooded areas. In *Proceedings of the 2019 ASPRS Annual Conference* (pp. 27-31).

Sarker, I.H., Kayes, A.S.M., Watters, P., 2019: Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. *Journal of Big Data* 6, 57. doi.org/10.1186/s40537-019-0219-y.

Shashank, A., Sajithvariyar, V. V., Sivanpillai, R., Brown, G. K., Sowmya, V.,2020: Identifying epiphytes in drone photos with a conditional generative adversarial network (C-GAN). In *ASPRS 2020 Annual Conference Virtual Technical Program*. doi.org/10.5194/isprs-archives-XLIV-M-2-2020-99-2020.

Shie, M., Dori, P., Reuven, R., 2005: The cross entropy method for classification. In *Proceedings of the 22nd international conference on Machine learning (ICML '05). Association for Computing Machinery, New York*, NY, USA, 561–568. doi.org/10.1145/1102351.1102422.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., 2015: Going deeper with convolutions. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., pp. 1-9.

Tefas A., Iosifidis A., Pitas I., 2013: Neural Networks for Digital Media Analysis and Description. In Iliadis L., Papadopoulos H., Jayne C. (eds). *Engineering Applications of Neural Networks. Communications in Computer and Information Science*, vol 383. Springer, Berlin, Heidelberg. doi.org/10.1007/978-3-642-41013-0_1

Wei, D., Zeyu, H., Zunkai, H., Li, T., Hui, W., Songlin, F., 2019: Designing efficient accelerator of depthwise separable convolutional neural network on FPGA. *Journal of Systems Architecture*, 97, Pages 278-286.

Zahra, M., Essai, A., Mohamed, H., Refaee, A., 2014: Performance Functions Alternatives of Mse for Neural Networks Learning. *International Journal of Engineering Research & Technology (IJERT)*. 3. pp- 967.

Zhao, Z., Zheng, P., Xu, S., Wu, X., 2019: Object Detection with Deep Learning: A Review. *In IEEE Transactions on Neural Networks and Learning Systems,* 30(11), pp. 3212-3232. doi.org/10.1109/TNNLS.2018.2876865.