

## EFFECT OF HYPERPARAMETERS ON DEEPLABV3+ PERFORMANCE TO SEGMENT WATER BODIES IN RGB IMAGES

Onteddu Chaitanya Reddy<sup>1</sup>, Illa Dinesh Kumar<sup>1</sup>, Pingali Sathvika<sup>1</sup>, Sajith Variyar V.V<sup>1\*</sup>, Sowmya V<sup>1</sup>, Ramesh Sivanpillai<sup>2</sup>

<sup>1</sup>Amrita School of Artificial Intelligence, Coimbatore, Amrita Vishwa Vidyapeetham, India.

(chaitanyareddy0702@gmail.com, ORCID: 0009-0001-1532-7124; idineshkumar12321@gmail.com, ORCID: 0009-0008-1560-8496; pingalisathvika@gmail.com, ORCID: 0009-0001-8437-010X; vv\_sajithvariya@cb.amrita.edu, ORCID: 0000-0003-3944-8155; v\_sowmya@cb.amrita.edu, ORCID: 0000-0003-3745-6944)

<sup>2</sup>Wyoming GIS Center, School of Computing, University of Wyoming, Laramie, WY, 82071, USA (sivan@uwyo.edu, ORCID: 0000-0003-3547-9464).

**KEY WORDS:** Hyperparameters, Epochs, Batch size, Image size, DeepLabv3+, Segmentation.

### ABSTRACT:

Deep Learning (DL) networks used in image segmentation tasks must be trained with input images and corresponding masks that identify target features in them. DL networks learn by iteratively adjusting the weights of interconnected layers using backpropagation, a process that involves calculating gradients and minimizing a loss function. This allows the network to learn patterns and relationships in the data, enabling it to make predictions or classifications on new, unseen data. Training any DL network requires specifying values of the hyperparameters such as input image size, batch size, and number of epochs among others. Failure to specify optimal values for the parameters will increase the training time or result in incomplete learning. The rationale of this study was to evaluate the effect of input image and batch sizes on the performance of DeepLabV3+ using Sentinel 2 A/B RGB images and labels obtained from Kaggle. We trained DeepLabV3+ network six times with two sets of input images of 128 x 128-pixel, and 256 x 256-pixel dimensions with 4, 8 and 16 batch sizes. The model is trained for 100 epochs to ensure that the loss plot reaches saturation and the model converged to a stable solution. Predicted masks generated by each model were compared to their corresponding test mask images based on accuracy, precision, recall and F1 scores. Results from this study demonstrated that image size of 256x256 and batch size 4 achieved highest performance. It can also be inferred that larger input image size improved DeepLabV3+ model performance.

### 1. INTRODUCTION

Machine learning and deep learning are both subfields of artificial intelligence (AI) that involve training networks to learn from data and make predictions or decisions (Mahesh et al., 2020). While they share some similarities, deep learning represents a more advanced and specialized approach within the broader scope of machine learning. On the other hand, Deep learning is a specific subset of machine learning that focuses on using neural networks with multiple layers to process and analyse complex data. Deep learning algorithms, also known as artificial neural networks, are inspired by the structure and function of the human brain.

The key difference between deep learning and traditional machine learning lies in the level of abstraction and feature engineering required. In traditional machine learning (ML), the task of manually identifying and extracting relevant features from the data falls upon domain experts, making it a time consuming and challenging process. Deep learning (DL), on the other hand, aims to automate this feature engineering step by directly learning hierarchical representations from raw data. DL algorithms leverage neural networks to extract and analyse features at various levels of abstraction, enabling them to capture complex patterns. However, it is important to note that in certain scenarios, DL algorithms may still require manual annotation or feature identification. By incorporating manual annotations, DL algorithms can leverage domain experts' knowledge and improve the efficiency and accuracy of the learning process. Therefore, while DL strives to automate feature extraction, there are instances where manual annotation remains essential for achieving optimal model

performance. Forward and backward propagation are fundamental processes in deep learning that enable the training of neural networks by iteratively adjusting the model's parameters to minimize the difference between predicted outputs and ground truth values. During forward propagation, input data is fed into the neural network, and computations are performed layer by layer, moving from the input layer to the output layer. Each layer consists of interconnected neurons that apply weighted sums and activation functions to produce outputs. The outputs from one layer serve as inputs to the next layer until the final output is generated. This forward pass allows the neural network to make predictions based on the current values of its parameters.

Next, backward propagation, also known as backpropagation, takes place. Backward propagation calculates the gradients of the loss function with respect to the network's parameters (Li et al., 2012). It starts at the output layer and iteratively works backward through the network, adjusting the weights and biases based on the computed gradients. The gradients are computed using the chain rule of calculus, which allows for efficient computation of gradients at each layer. By propagating the gradients backward, the neural network updates its parameters in a way that reduces the difference between predicted outputs and the actual targets. This iterative process continues until the model converges to a state where the prediction accuracy is optimized, and the network has learned to generalize well to unseen data. To successfully train a Deep Learning network, an analyst must specify certain constraints or Hyperparameters. These Hyperparameters determine the behaviour and performance of a DL network. Specifying appropriate hyperparameters values is essential for achieving optimal model performance.

\* Corresponding author

## 1.1 Hyperparameters

Hyperparameters are constraints that specify or limit the network's ability to execute forward and backward propagation steps (Hertel et al., 2020). The common hyperparameters include learning rate, batch size, image size, number of epochs, activation function, optimizer, regularization, and initial weights. There are different hyperparameters as listed below:

**1.1.1 Learning rate:** The learning rate is a critical hyperparameter that controls the step size of parameter updates during training. It influences the balance between convergence speed and accuracy. Choosing an appropriate learning rate is crucial to ensure efficient and effective model training.

**1.1.2 Image size:** Image size refers to the dimensions of the input images fed into the neural network (Probst et al., 2018). The choice of image size can significantly impact the performance of the segmentation model. Larger image sizes may capture more detailed information but can also increase computational requirements. On the other hand, smaller image sizes may sacrifice some details but can lead to faster processing times.

**1.1.3 Batch size:** Batch size, on the other hand, pertains to the number of training samples processed together in a single step of the training process where a batch of training samples is processed together. It influences the trade-off between computational efficiency and model convergence. A larger batch size can accelerate training by parallelizing computations, but it may also require more memory resources. Conversely, a smaller batch size can provide more accurate weight updates but can be computationally expensive.

**1.1.4 Number of epochs:** Number of epochs is a significant hyperparameter in training models. It determines the number of times the dataset is iterated during training. Finding the right balance is important, as too few epochs may result in underfitting, while too many can lead to overfitting. Proper selection requires experimentation and monitoring to achieve optimal model performance.

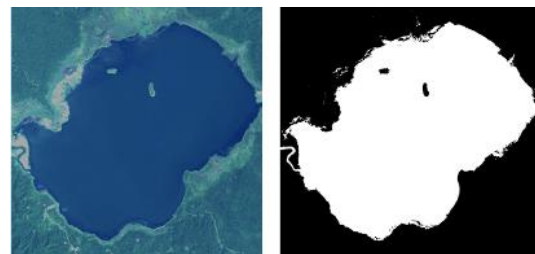
**1.1.5 Activation Function:** Activation functions are essential hyperparameters in neural networks, introducing non-linearity. They affect the model's capacity to learn and approximate complex functions. Popular choices include sigmoid, tanh, and ReLU, each with unique properties that impact network performance. Proper selection and experimentation are crucial to maximize model capabilities.

In this study, we evaluated the role of image and batch sizes for training DeepLabV3+ (Chen et al., 2018), a DL network, for segmenting waterbodies in satellite images (Liu et al., 2021) (George et al., 2023). By investigating the effects of varying image sizes and batch sizes on water body segmentation, this study aims to optimize these hyperparameters for achieving accurate and efficient segmentation results. Understanding the impact of these hyperparameters can provide insights into the trade-offs between computational resources and segmentation performance (Yuan et al., 2021), ultimately contributing to the advancement of water body segmentation techniques in remote sensing and environmental monitoring applications.

## 2. MATERIALS AND METHODS

### 2.1 Dataset Description

The dataset used in this research was obtained from Kaggle, a website that provides access to a diverse collection of open datasets. Specifically, the dataset comprises satellite images of water bodies along with their corresponding masks in greyscale format (Figure 1). The satellite images were captured by the Sentinel-2 (Escobar, n.d.) satellite, a remote sensing platform used for Earth observation.



**Figure 1:** Example of an RGB image and the corresponding mask that highlights water pixels in white color.

The images are in RGB (Red, Green, Blue) format, representing the visual spectrum. Each image is accompanied by a black and white mask, where pixels in white color indicate water and black represents areas other than water. Normalized Difference Water Index (NDWI) was used for identifying water pixels in the image (McPheters, 1996). NDWI is a commonly used index for detecting water in satellite imagery. However, for this study, a higher threshold was employed to specifically detect and delineate water bodies. The images were pre-processed using Rasterio, a Python library for handling geospatial data. The pre-processing step ensured the compatibility and appropriate formatting of the satellite images for further analysis.

### 2.2 Data Pre-Processing

In the data pre-processing step of this work, quality control measures were implemented to ensure the accuracy and reliability of the dataset. This included identifying and removing images with incorrect masks, such as green vegetation or bare ground misclassified as water. Further, images containing turbid water or water with varying colors were excluded to ensure data uniformity. After this step only the good water quality images are retained.

Next, a significant emphasis was placed on changing the image sizes and creating subsets for training, validation, and testing. The original dataset consisted of varying image sizes, which were divided into smaller patches of size 128x128 and 256x256. This division into patches offered computational efficiency, localized analysis, and the ability to capture both fine-grained and contextual information, thereby enhancing the effectiveness of subsequent analyses and model training.

The resulting subsets of 128x128 and 256x256 image patches, along with the corresponding allocation into training, validation, and test sets as shown in the table 1, formed the foundation for conducting rigorous and unbiased water body segmentation experiments.

Image Size	Training set	Validation set	Test set
128x128	14252	1860	1861
256x256	2916	364	353

**Table 1:** Number of images of two different dimensions used for training, validating, and testing the trained models.

### 2.3 DeepLabV3+ for Water body Segmentation

Satellite images are widely used for water body extraction (Rithin Paul Reddy et al., 2018). The water body segmentation in this study was performed using the DeepLabv3+ architecture, a highly effective convolutional neural network (CNN) model for semantic segmentation (Chen et al., 2018). CNN based network is a potential approach for satellite image processing (Thanga Manickam et al., 2021). The training dataset consisted of image patches of sizes 128x128 and 256x256, allowing the model to capture both fine-grained details and contextual information. During the training process, batch sizes of 5, 8, and 16 were used, with the corresponding masks guiding the learning process. The DeepLabv3+ architecture incorporates an encoder-decoder structure with atrous spatial pyramid pooling (ASPP) modules, enabling multi-scale feature fusion and accurate predictions on objects of different sizes (Sunandini et al., 2023). By utilizing dilated convolutions at multiple rates, the model captures features at various scales, while skip connections preserve fine-grained details. This combination of architecture and dataset facilitated precise and reliable water body segmentation.

### 2.4 Technical Requirements

The study required specific hardware and software configurations to facilitate the computational tasks involved in deep learning experiments. For hardware, the High-Performance Computation (HPC) Server was utilized. The server comprised a total of 5 nodes, with each node being equipped with 28 CPUs and 254 GB of memory. This hardware infrastructure provided substantial computational power and memory capacity necessary for processing large-scale datasets and training complex deep learning models. In terms of software requirements, Python, a widely adopted programming language in the field of machine learning and deep learning, was used as the primary programming language. Python offers extensive libraries and frameworks that supports various aspects of deep learning implementation. Furthermore, TensorFlow, a popular open-source deep learning framework, was employed as the core software tool. TensorFlow provides a comprehensive ecosystem for designing, training, and evaluating deep neural networks. Its scalability and flexibility made it well-suited for conducting the deep learning experiments in the research study.

### 2.5 Hyperparameter Configurations

To optimize the performance of the deep learning models for water body segmentation, (Tsai et al., 2020). specific hyperparameter configurations were employed in this study. Two key hyperparameters under consideration were the image size and batch size. The image size, chosen as 128x128 and 256x256, determined the input dimensions of the neural networks and played a crucial role in capturing the desired level of detail from the satellite images. By selecting appropriate image sizes, a balance was achieved between computational efficiency and the ability to capture fine-grained features. The batch size, on the other hand, determined the number of training samples processed

in each iteration during model training. For this study, batch sizes of 4, 8, and 16 were tested to investigate their impact on model convergence and computational efficiency. Different combinations of image size and batch size were examined, resulting in separate trained models for each combination. The deep learning models, specifically the DeepLabv3+ architecture, were trained for a fixed number of epochs, with 100 epochs chosen as the training duration. This configuration allowed the models to iterate through the training data, refining their weights and optimizing their performance over time. As a result, trained models were obtained for each combination of image size and batch size, serving as the outputs of the training process.

### 2.6 Evaluation Metrics

In assessing the performance of the water body segmentation models, multiple evaluation metrics were considered, including accuracy, precision, recall, and F1 score. Accuracy quantifies the proportion of correctly classified samples, taking into account both the water and non-water classes. A higher accuracy indicates a higher level of agreement between the predicted labels and the ground truth across all classes.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Samples} \quad (1)$$

Precision measures the proportion of correctly classified water pixels out of all pixels predicted as water. It provides insights into the model's ability to correctly identify true water pixels, minimizing false positives.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (2)$$

Recall calculates the proportion of correctly classified water pixels out of all actual water pixels. It highlights the model's capability to capture all relevant water pixels, reducing false negatives.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (3)$$

F1 score is the harmonic mean of precision and recall, providing a balanced measure that considers both metrics. It combines precision and recall into a single value, reflecting the overall performance of the model in correctly identifying water pixels.

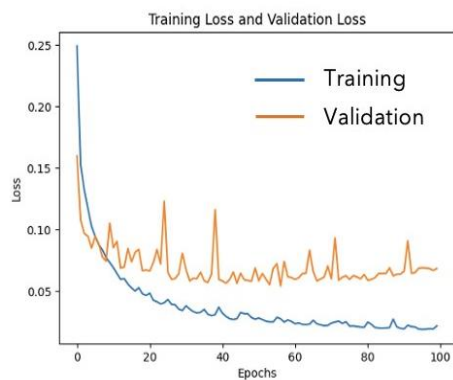
$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

## 3. RESULTS

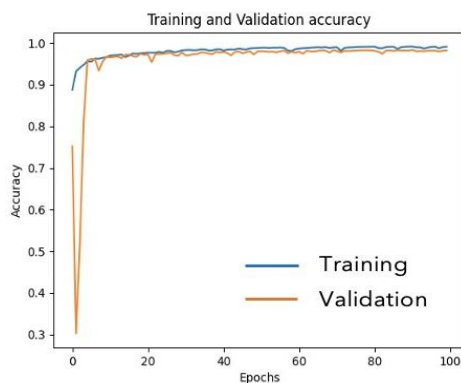
### 3.1 Accuracy Plots & Loss Plots

In the study, it was observed that the accuracy of the water body segmentation models consistently improved as the number of training epochs increased. The accuracy plots exhibited an upward trend, indicating that the models became progressively more proficient in accurately classifying water and non-water regions. This improvement in accuracy can be attributed to the models' ability to learn and capture the relevant features of water bodies over time. Additionally, it was observed that the loss plots showed a decreasing trend as epochs increased. The decline in loss values indicates that the models effectively minimized errors

and discrepancies between the predicted and actual labels. As the models converged, the loss plot gradually flattened, suggesting that the models reached a state of stability and achieved optimal segmentation performance. Accuracy and loss plots for obtained when DeepLabV3+ was trained with images of dimension 128 x 128 (Figures 2 and 3), and 256 x 256 (Figures 4 and 5) show the importance of training the models over an adequate number of epochs to enhance accuracy and minimize loss, ultimately leading to improved water body segmentation results. Plots obtained for other parameter combinations are presented in Appendix (Figures 6 – 9).



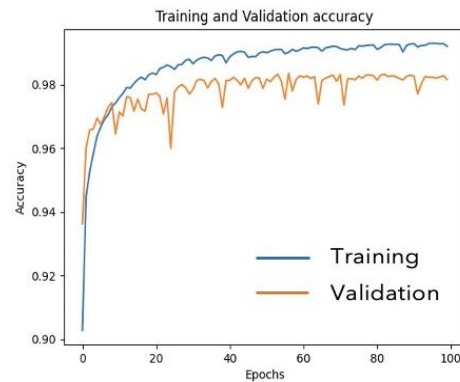
**Figure 2:** Accuracy plot obtained when DeepLabV3+ was trained with 128x128 images with a batch size 16.



**Figure 3:** Loss plot obtained when DeepLabV3+ was trained with 128x128 images with a batch size 16.



**Figure 4:** Accuracy plot obtained when DeepLabV3+ was trained with 256x256 images with a batch size 16.



**Figure 5:** Loss plot obtained when DeepLabV3+ was trained with 256x256 images with a batch size 16.

### 3.2 Evaluation Scores

Image size	Batch size 4	Batch size 8	Batch size 16
128x128	95.44	95.60	95.48
256x256	96.54	96.38	96.37

**Table 2:** Accuracy scores obtained when DeepLabV3+ network was trained with 2 image sizes and 3 batch sizes.

Accuracies achieved by the models are presented in the table. From table 2, it can be inferred that increasing the image size from 128x128 to 256x256 led to a slight improvement in accuracy across all batch sizes. This suggests that larger image sizes provided more detailed information, allowing the models to better capture and classify water bodies. Large batch sizes can result in increased processing time and resource requirements, while also potentially leading to limited generalization.

On the other hand, there was no clear trend to indicate that batch size influenced the accuracy of the model's ability to predict water pixels. The accuracies achieved with batch sizes 4, 8, and 16 were relatively close, with minor variations observed. This indicates that the choice of batch size had a minimal effect on the overall performance of the models in terms of accuracy.

Image size	Batch size 4	Batch size 8	Batch size 16
128x128	91.46	92.46	91.23
256x256	97.33	96.21	97.92

**Table 3:** Precision scores obtained when DeepLabV3+ network was trained with 2 image sizes and 3 batch sizes.

The precision values achieved by the model are presented in Table 3. From the results, it can be observed that the precision values vary for different image sizes and batch sizes. When considering image size, the precision values for water body segmentation are consistently higher for the larger image size of 256x256 compared to the smaller image size of 128x128. This suggests that the larger image size provides more accurate

identification and classification of true water pixels, resulting in higher precision values. Regarding batch size, no significant trend is evident. The precision values for different batch sizes (4, 8, and 16) do not exhibit a consistent pattern. This indicates that the choice of batch size does not have a substantial impact on the precision of the water body segmentation models.

Image size	Batch size 4	Batch size 8	Batch size 16
128x128	70.65	71.59	70.63
256x256	76.63	74.04	75.11

**Table 4:** Recall scores obtained when DeepLabV3+ network was trained with 2 image sizes and 3 batch sizes.

Recall values obtained are presented in the Table 4. From the results, it can be inferred that the recall values varied for different image sizes and batch sizes. For the 128x128 image size, the recall values ranged from 70.65% to 71.59% for batch sizes 4 and 8, respectively. Batch size 16 achieved a slightly lower recall of 70.63%. This suggests that the models had a moderate ability to correctly capture and identify water pixels compared to the total number of actual water pixels. For the 256x256 image size, the recall values ranged from 74.04% to 76.63% for batch sizes 8 and 4, respectively. Batch size 16 achieved a recall of 75.11%. These higher recall values indicate a relatively better performance in correctly identifying water pixels for the larger image size compared to the smaller image size.

Image size	Batch size 4	Batch size 8	Batch size 16
128x128	76.94	78.04	76.95
256x256	83.39	81.08	82.05

**Table 5:** F1-scores obtained when DeepLabV3+ network was trained with 2 image sizes and 3 batch sizes.

The F1-scores obtained are presented in the Table 5. From the results, it can be inferred that the F1 scores varied for different image sizes and batch sizes. For the 128x128 image size, the F1 scores ranged from 76.94% to 78.04% for batch sizes 4 and 8, respectively. Batch size 16 achieved a slightly lower F1 score of 76.95%. These scores indicate a reasonably balanced performance in terms of precision and recall for water body segmentation. For the 256x256 image size, the F1 scores ranged from 81.08% to 83.39% for batch sizes 8 and 4, respectively. Batch size 16 achieved an F1 score of 82.05%. These higher F1 scores suggest a relatively better balance between precision and recall for the larger image sizes.

Increasing batch size does not consistently lead to improved metric scores. Specifically, for images sized 128 x 128 pixels, a batch size of 8 performs the best, while for 256 x 256-pixel images, a batch size of 4 yields optimal results. Notably, the combination of a 256 x 256 image size and a batch size of 4 consistently outperforms other configurations in terms of achieved metrics. These insights contribute to the ongoing exploration of efficient deep learning techniques and offer valuable guidance for enhancing model performance in image-related tasks.

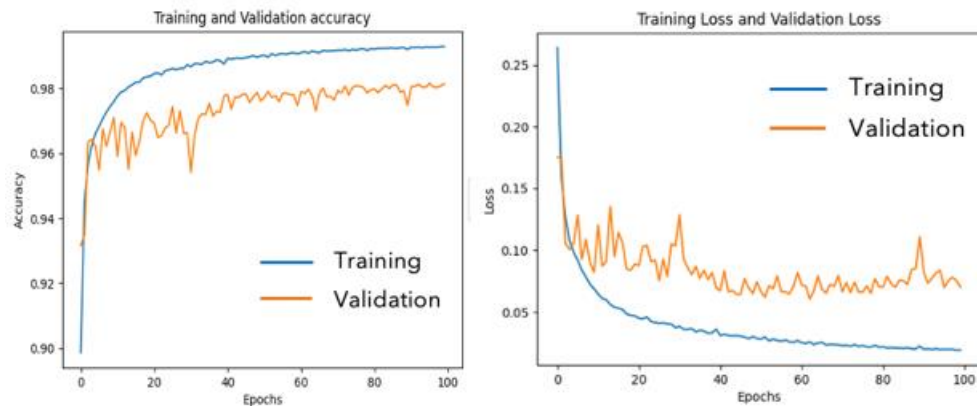
## REFERENCES

- Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European conference on computer vision (ECCV) pp. 801-818.
- Escobar, F., (n.d). "Satellite Images of Water Bodies", <https://www.kaggle.com/datasets/franciscoescobar/satellite-images-of-water-bodies>, CC BY-NC-SA 4.0, (6 June 2022).
- George, B., Sajith, V. V., Sowmya, V. and Sivanpillai, R., 2023. Performance improvement of water body segmentation by deeplabv3+ using two-dimensional variational mode decomposition. In: 2023 10th International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, pp. 603–608.
- Lars Hertel, Daniel Gillen, and Michele Guindani. 2020. On Hyperparameter Optimization for Deep Learning. Ph.D. Dissertation. University of California, Irvine. Advisor(s) Baldi, Pierre. Order Number: AAI27741857.
- Li, J., Cheng, J.-h., Shi, J.-y. and Huang, F., 2012. Brief introduction of back propagation (bp) neural network algorithm and its improvement. In: Advances in Computer Science and Information Engineering: Volume 2, Springer, pp. 553–558.
- Liu, M., Fu, B., Xie, S., He, H., Lan, F., Li, Y., Lou, P. and Fan, D., 2021. Comparison of multi-source satellite images for classifying marsh vegetation using deeplabv3 plus deep learning algorithm. Ecological Indicators 125, pp. 107562.
- Mahesh, B., 2020. Machine learning algorithms-a review. International Journal of Science and Research (IJSR) 9, pp.381-386.
- McFeeters, S. K., 1996. The use of the normalized difference water index (ndwi) in the delineation of open water features. International journal of remote sensing 17(7), pp. 1425–1432.
- Probst, P., Bischl, B., & Boulesteix, A. L. (2018). Tunability: Importance of hyperparameters of machine learning algorithms. arXiv pp arXiv:1802.09596.
- Rithin Paul Reddy, K., Srija, S.S., Karthi, R. and Geetha, P., 2020. Evaluation of water body extraction from satellite images using open-source tools. In Intelligent Systems, Technologies and Applications: Proceedings of ISTA 2018. pp. 129-140. Springer Singapore.
- Sunandini, G., Sivanpillai, R., Sowmya, V. and Variyar, V. S., 2023. Significance of atrous spatial pyramid pooling (aspp) in deeplabv3+ for water body segmentation. In: 2023 10th International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, pp. 744–749.
- Thanga Manickam, M., Karthik Rao, M., Barath, K., Shree Vijay, S. and Karthi, R., 2022. Convolutional neural network for land cover classification and mapping using landsat images. In Innovations in Computer Science and Engineering: Proceedings of the Ninth ICICSE, 2021. pp. 221-232. Singapore: Springer Singapore.
- Tsai, C.W., Hsia, C.H., Yang, S.J., Liu, S.J. and Fang, Z.Y., 2020. Optimizing hyperparameters of deep learning in predicting

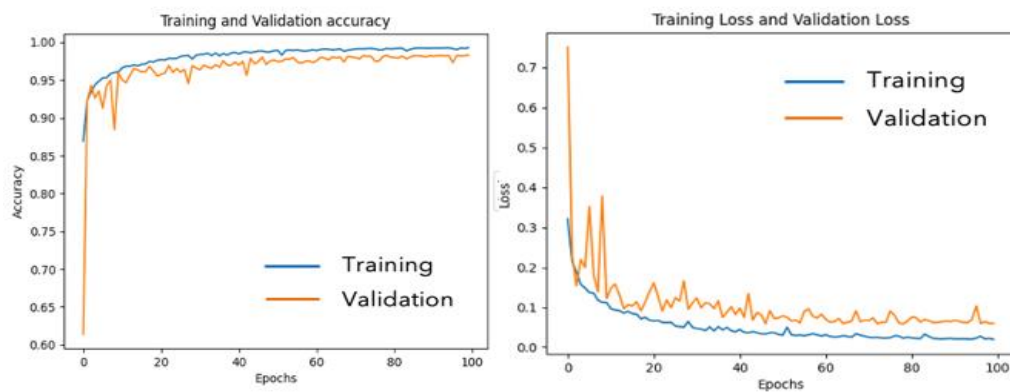
bus passengers based on simulated annealing. Applied soft computing, 88, p.106068.

Yuan, K., Zhuang, X., Schaefer, G., Feng, J., Guan, L. and Fang, H., 2021. Deep-learning-based multispectral satellite image segmentation for water body detection. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14, pp.7422-7434.

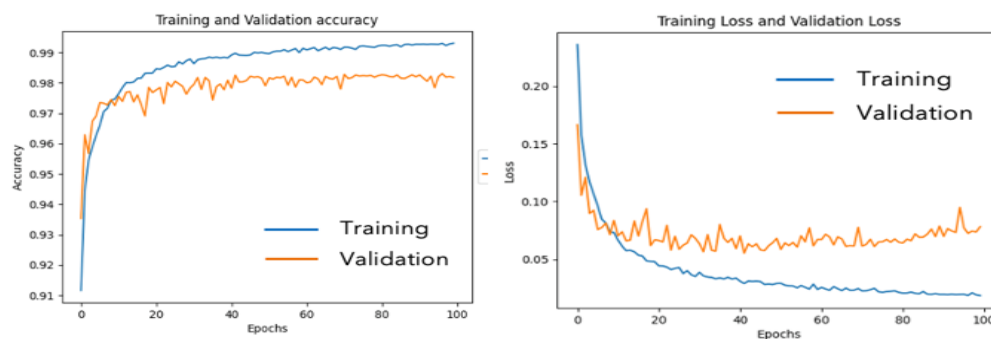
## APPENDIX



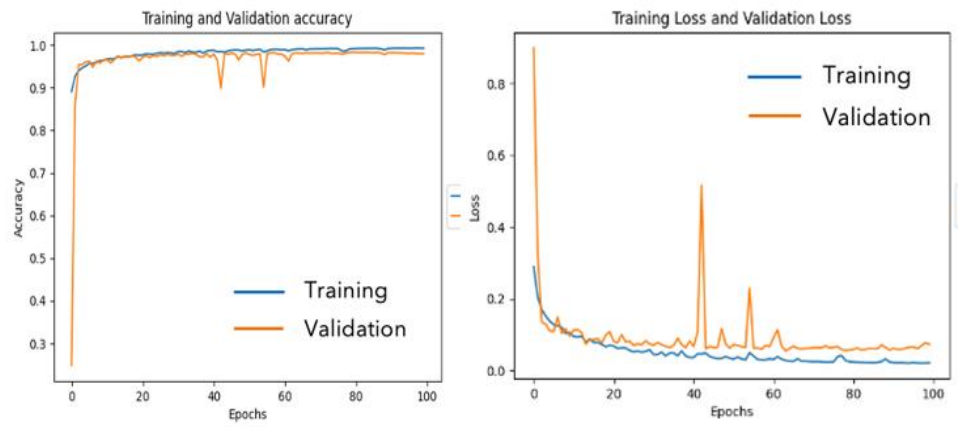
**Figure 6:** Accuracy (left) and loss (right) plots obtained when DeepLabV3+ was trained with image size 128x128 and batch size 4.



**Figure 7:** Accuracy (left) and loss (right) plots obtained when DeepLabV3+ was trained with image size 256x256 and batch size 4.



**Figure 8:** Accuracy (left) and loss (right) plots obtained when DeepLabV3+ was trained with image size 128x128 and batch size 8.



**Figure 9:** Accuracy (left) and loss (right) plots obtained when DeepLabV3+ was trained with image size 256x256 and batch size 8.