Improving the U-Net Segmentation Model for Land Cover Classification in Satellite Image Processing

Ramiel Deticio
Gokongwei College of Engineering
De La Salle University
Manila, Philippines
ramiel_g_deticio@dlsu.edu.ph

Argel Bandala
Gokongwei College of Engineering
De La Salle University
Manila, Philippines
argel.bandala@dlsu.edu.ph

John Anthony Jose
Gokongwei College of Engineering
De La Salle University
Manila, Philippines
john.anthony.jose@dlsu.edu.ph

Ronnie Concepcion II
Gokongwei College of Engineering
De La Salle University
Manila, Philippines
ronnie.concepcion@dlsu.edu.ph

Mark Angelo Purio
College of Engineering
Adamson University
Manila, Philippines
markangelo.purio@adamson.edu.ph

Edwin Sybingco
Gokongwei College of Engineering
De La Salle University
Manila, Philippines
edwin.sybingco@dlsu.edu.ph

Richard Josiah Tan Ai
Gokongwei College of Engineering
De La Salle University
Manila, Philippines
richard.tanai@dlsu.edu.ph

***Abstract*—The development of machine learning methods for onboard satellite processing is important in order to facilitate the filtering of collected data samples to maximize the use of the device's limited resources. Land cover classification can be used to focus the collected data on certain terrain types by utilizing classification methods to determine the class probabilities of individual pixels in a collected satellite image. The importance of the accuracy of the segmentation model used for such a task is important in order to avoid the trashing of data samples that offer significant information and the prioritization of data samples which offer less in terms of usable information, which in the case of land cover classification is determined by which terrain features may be prioritized over others.**

This study focuses on the U-Net segmentation architecture and performs an experimental study on the effects on two aspects on the training of a segmentation model for increased performance. This includes the division of the images in the dataset into smaller patches and the replacement of the CNN encoder of the segmentation architecture. The changes made to the baseline model introduced an increase in the IoU score from 0.68 to 0.7273.

***Index Terms*—computer vision, edge computing, land cover classification, nanosatellite**

I. Introduction

There have been numerous developments in satellite technology, including the development of smaller and more cost-efficient satellites. These nanosatellites allow space research and technology to be more accessible by reducing the material requirements to create a functional satellite [3]. Unfortunately, the tradeoff is in the satellite's capabilities. With a smaller size, the physical constraints of a satellite limits its total power capacity, computing power, and size restrictions, preventing the use of technology that may be too resource-intensive [1].

One common application for satellites is its use in Earth observation. This involves the collection of satellite images from orbit for various purposes. The main problem with this is that a small satellite can't always store all the information that is collected by an imaging sensor [1]. Satellites have a limited time and bandwidth in downlinking any new data that may have been collected during this orbit, and satellites only have a very limited storage for storing this data in between downlinking sessions.

An onboard classification system intends to mitigate this problem by having a structured computational solution for the efficient allocation of resources and memory for storing any collected images [2]. Land cover classification involves the use of a segmentation algorithm that helps identify terrain features in a satellite image to provide information about these images that would be considered as important information.

In the preprocessing of satellite images, false positives could lead to the use of resources on data that is less needed, and false negatives could lead to the trashing of data samples that could have been very important. The accuracy of the model to be used in preprocessing satellite image data before transmission is important to avoid such cases, and ensure that the majority of resources are used to transmit useful data and no useful samples are lost in the process. Therefore, there is a need to consistently develop and improve the potential of onboard classification so that these nanosatellites are able to provide high-quality data with the limited resources that they have.

Previous methods involved the use of U-Net [19] or a UNet-based model for onboard image segmentation in nanosatellite applications demonstrating that the model is viable for such an application. Leong et al. developed an Image Classification Unit (ICU) intended for a CubeSat which utilized a U-Net model for cloud detection using image

979-8-3503-0219-6/23/$31.00 ©2023 IEEE

segmentation [8]. The CloudScout segmentation model used in the Φ-Sat-1 CubeSat was another segmentation model for cloud detection serving as the next iteration of the CloudScout model [5], was said to be derived from the encoder-decoder structure of U-Net [9]. Another example was the MobU-Net developed by Zhang et al. as a lightweight version of the U-Net model intended for CubeSats using depthwise separable convolutions [6].

The use of a basic U-Net model by Leong et al. in their ICU demonstrated the viability of a U-Net architecture for onboard satellite image processing [8]. This model was implemented on an STM32 microcontroller after some compression techniques such as quantization, simulating its viability for use with the hardware restrictions of a nanosatellite. This was considered for this study after it was mentioned that a previous iteration of the ICU was implemented on a microcontroller circuit developed for the BIRDS 4 1U CubeSat. This study makes the assumption that the U-Net architecture can be used as a baseline for onboard image segmentation and therefore will focus on developing the model's performance upon a U-Net model as a baseline.

The development of this classification system will involve the development of a segmentation model that will be adapted for this type of application. The main contributions of this paper are as follows:

- The preprocessing of existing satellite data for land cover classification.
- The retraining of a UNet-based model for the specific purpose of land cover classification.
- The evaluation of this model on the DeepGlobe 2018 challenge dataset on land cover classification.

The scope of this study will be limited to the software development of an satellite image segmentation model.

II. Review of Related Literature

A. Convolutional neural networks

An artificial neural network was modeled on the structure of a biological neural network which focuses on a network of interconnected neurons sending signals to each other. This provides a computational model that can handle larger and more complex mathematical tasks that would be too much for a standard algorithmic solution. In the use of neural networks in various types of machine learning tasks, the development of the convolutional neural network significantly improved the computational efficiency of artificial neural networks with a greater focus on local connectivity allowing for more effective feature representations [16]. The convolutional layers significantly reduced the number of connections and therefore the number of computations when performing machine learning tasks.

A core concept in supervised learning is the fact that the capabilities of the network can be limited by the amount of annotated data used in training. It is highlighted that the general rule when it comes to neural networks is that the larger and deeper networks provide the ability to solve more complex computational tasks. Although, in order to compensate, a larger number of data samples would be needed to minimize the training loss of the network. The more samples that are available, the more efficient the training process [11].

B. CNNs for Image Segmentation

The task of image segmentation is similar to image classification, except the classification method is done at a pixel-level instead of the image-level [23]. Whereas the output of a classification model is a probability value or vector of the possible classes of the image, the corresponding output for a segmentation model is a map of an image with classification probabilities of each pixel. The fully convolutional network (FCN) allowed the use of convolutional neural networks in image segmentation, by using convolutional layers to generate a segmentation mask by replacing the fully connected layers in the classifier head [18].

One of the more conventional types of CNN-based segmentation are the encoder-decoder models. This usually involves an encoder backbone, which is often the first few layers of a CNN like VGG or ResNet, and a corresponding decoder, a similar group of layers that comes after the encoder but with deconvolution and upsampling instead [23]. Some examples of segmentation architecture which followed this format include DeconvNet [20], SegNet [21] and U-Net [19].

U-Net was a segmentation model that was intended for the purpose of medical image segmentation [19]. It had two paths, a contracting path to capture context and an expanding path for the precise localization. The structure of the U-Net architecture relies on data augmentation to learn from a very small set of annotated images, a similar problem which can occur in satellite image datasets due to the difficulty of collecting satellite images.

A common technique used in datasets for image segmentation, but also in other deep learning methods is the use of data augmentation to artificially increase the number of samples in a dataset [23]. Data augmentation usually involves the physical modification of the image samples such as cropping, rotation, or flipping. The use of data augmentation can help avoid the problem of overfitting, which is when there is a significant discrepancy in performance occurs when a model is introduced to new data as compared to its performance on the training set of the data. Data augmentation can help decrease the potential of overfitting, as well as leading to faster convergence and improving the overall robustness of the model [23].

C. Orbital edge computing

The physical constraints of satellites gives a limitation to how many images can be stored and eventually transmitted back to the ground station [2]. With such a small window for downlinking, there is little opportunity to send good, high-quality images that may have been captured. Because of this, there is a need to prioritize the transmission of images back to Earth, since there is a limited capacity to store and transmit. For smaller satellites in particular, you have less power which significantly limits downlink transmission.

The collected data is sent to a place with more computing power to analyze the data, but an argument can be made for the onboard processing [1]. Akin to conventional sensor systems, local processing can be used over sending raw data back to an external location. While the latter utilize faster computing processes, it is severely limited by the capacity of the network used for transmission.

D. Machine learning for onboard satellite processing

One of the earlier uses of machine learning was the onboard image classification used in the IPEX CubeSat mission. The IPEX CubeSat mission involved the use of complex operations for onboard instrument processing [7]. This took on the form of two applications, the classification of satellite images and salience image analysis. The image classification made use of a random forest classifier for its machine learning component.

Deep learning is known for the heavy computational requirements, but also for their capability to perform complex operations. One study implements a two step process involving a JPEG2000 image compression scheme and the processing of these compressed images in optimized versions of segmentation networks using lightweight layers [6]. MobU-Net and MobDeconv-Net were the products of these solutions which utilized depthwise separable convolutions utilized in MobileNets to the segmentation networks of U-Net and Deconv-Net. The networks have reduced memory costs and increased inference speeds at the cost of slight decreases in accuracy.

The CubeSatNet networks were ultralight CNNs intended for use in 1U CubeSats [10]. A Cubesat imagery dataset was created from CubeSat images from all sources, nadir-pointed Google Earth thumbnail images, and ISS Horizon images. The network is constructed using a conv layer, FC layer and output layer. Two versions of the network were used with version 2 using a global pooling layer between the convolutional layers and the FC layers instead of the flatten layer used in version 1.

The CloudScout algorithm was implemented to perform onboard cloud detection on hyperspectral images [5]. It was developed to be implemented on the Eyes of Things board using a Myriad 2 Vision Processing Unit (VPU). It was used for binary image classification in which it determines whether a given satellite image was cloudy or not. The CloudScout algorithm was used again in the Φ-Sat-1 mission where it was adapted into a customized segmentation network [9].

One study focused on the implementation of an Image Classification Unit (ICU) for onboard use of a segmentation model [8]. The ICU focused on the development of a microcontroller based circuit intended for running an image segmentation algorithm to be used onboard a satellite. This study utilized a U-Net segmentation model with quantization trained for cloud detection using a custom dataset made from LandSat 8 images while using the SPARCS cloud assessment dataset for evaluation. The model was then compressed and quantized to be uploaded onto an STM32F746BGT6 microcontroller.

III. Methodology

A. Hardware Restrictions and Feasibility of Study

Cubesats are composed cube modules measuring around 10x10x10 cm3 and weighing around 1.3 kg [3]. A 1U CubeSat would have one of these cubes, a 2U CubeSat would have two and would measure around 10x10x20 cm3, and so on. However, the restricted size of these cubes, despite being more cost-efficient, leads to limited functionality compared to conventional satellites [4].

The aim of this study is to simulate the machine learning methods that were used in a previous study intended for a CubeSat application. The Image Classification Unit developed by Leong et al. [8] involved the use of a microcontroller circuit with a usable memory capacity of 2 MB. The model that was developed in that study was a U-Net, converted into a C++ char array and quantized, reducing the model size from 2.25 MB to 867 KB.

Dealing with the uncompressed versions of the model will lead to model sizes that will be larger than appropriate for practical usage. Although, this study aims to test the effects of changing model architectures to the model size before it has undergone compression techniques. Future studies involve the compression of already smaller models to potentially discover smaller and even more feasible model sizes.

B. Dataset

The dataset used is the land cover classification dataset from the DeepGlobe 2018 challenge [24]. This involves satellite images with corresponding segmentation masks highlighting which parts of the image belong to which terrain type including: urban, agriculture, rangeland, forest, water, barren and unknown. Some sample images and their corresponding segmentation masks can be seen in Figure 1. The dataset was processed with a 90/10 training/validation split. A separate dataset was derived from this base dataset, in which each image was separated into four smaller patches, from a size of 2048x2048 into 1024x1024 pixels.

Evaluation of the created model would involve the use of a pixel-wise Jaccard index (Intersection over Union), which can be seen in Eq. 1,

$$IoU_j = \frac{\sum_{i=1}^{n} TP_{ij}}{\sum_{i=1}^{n} TP_{ij} + \sum_{i=1}^{n} FP_{ij} + \sum_{i=1}^{n} FN_{ij}}, \quad (1)$$

where IoU_j refers to the IoU score for each pixel belonging to class j to a total number of n images. TP_{ij}, FP_{ij}, and FN_{ij} refers to every true positive, false positive and false negative pixels of class j in each image i [24]. The final score is the average IoU among classes, which is expressed in Eq. 2,

$$mIoU = \frac{1}{k}\sum_{j=1}^{k} IoU_j, \quad (2)$$

where k is the total number of classes [24].

C. Model Architecture and Training

The model used is a U-Net segmentation model with an encoder model of VGG16 [13] or MobileNet v2 [15], both of which has been pretrained on ImageNet. It was implemented in PyTorch using the Segmentation Models library for PyTorch [25]. The images were cropped to a size of 1024x1024 and the training images were augmented with horizontal and vertical flips. The model is trained over 5 epochs using Adam as an optimizer. Dice loss, which can be seen in Eq. 3, is used as the training loss function [22]. Dice loss is equivalent to the F-score between the ground truth and the predicted result.

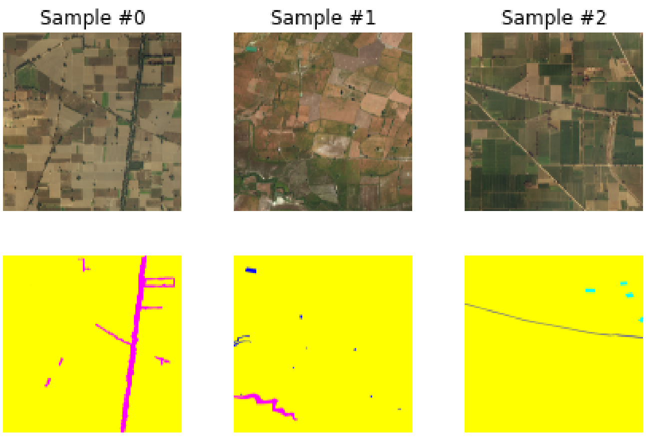

Fig. 1. Samples from the land cover classification dataset in the DeepGlobe 2018 challenge.

$$DiceLoss\left(y,\overline{p}\right) = 1 - \frac{(2y\overline{p}+1)}{(y+\overline{p}+1)} \tag{3}$$

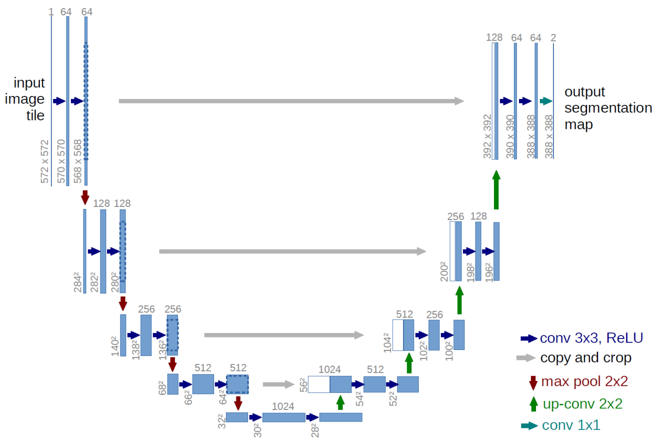

Fig. 2. Diagram of the U-Net architecture.

IV. Results and Discussion

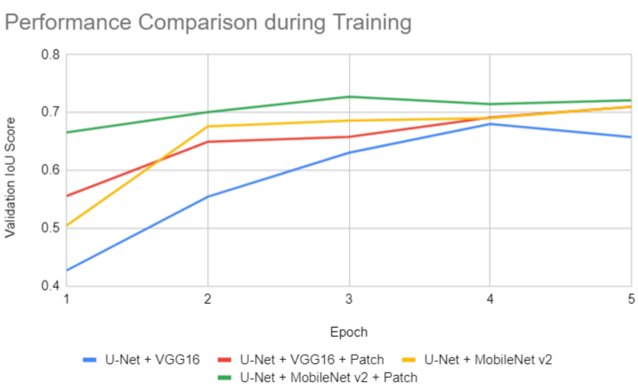

Fig. 3. Graph showing measure of IoU after each epoch of training.

The results of the experimental trials show that from the lowest peak accuracy of 0.68, the model can be improved to up to 0.7273 by opting to use MobileNet v2 [15] as the encoder instead of VGG16 [13] and the images be processed into smaller patches. A sample result through inference can be seen in Figure 4. The accuracy comparisons for the various setups can be seen in Figure 3. The scores presented by this figure were the validation scores obtained after each epoch of training through an evaluation process on the validation set. The overall graph shows how this metric changes between epochs.

Fig. 4. Sample accurate result from model inference with the actual image (left), the ground truth mask (middle) and the predicted segmentation map (right)

It is important to note that the training was done through Google Colaboratory for the immediate availability of GPU computing power. Because of this, each session could have varied training times. Although, any stated differences between training times between training setups was a trend that was observed after multiple trials and would require a more controlled experimental setup to be strongly conclusive.

MobileNet v2 [15] as an encoder has resulted in reduced training times compared to VGG16 [13]. This is probably because the use of bottleneck residuals in MobileNet v2 architecture. This development made use of the depthwise separable convolutions in MobileNet v1, which had a significant effect on increasing computational efficiency [14], and also utilizing inverted residual connections, which were shown to be more memory efficient [15].

Although, while the patches have yielded higher accuracies, it would also lengthen the total training time. This shows how accuracy and training time would both increase if the images were separated into patches rather than simply resizing the raw images, which overall reduces the number of samples and causes a loss in information. While it is possible the diversity in the sample images persists in simply resizing the images, it is worth considering if the small accuracy increase is worth sacrificing for a faster training time, especially if the model is to be trained on limited hardware.

Aboard a satellite controlled environment, another thing important to note that the model will be trained on the ground and that the only necessary components for the implementation would be the model itself and its weights after training. In this environment, it will be subject to a number of other physical factors which could affect its operation and would also depend on the quantity and quality input images provided by the corresponding satellite. To evaluate this, further testing would be required on a simulated environment with a compressed version of the model.

V. Conclusion

The study has shown that both aspects have resulted in improvements in accuracy. Three things could be shown in the result, the encoder CNN affects the effectiveness of the segmentation model, the size in which the images are processes affects the accuracy of evaluation while lengthening the training time, and both of these factors could have a significant effect when considering the cost to accuracy ratio of a developed model for processing satellite images. The dataset used was intended to be representative of the satellite images that could be collected while maintaining a healthy diversity of terrain types, but could be different in terms of clarity and resolution of images that may be collected in real-time so this could be a point of consideration for future research.

VI. Recommendation

For future studies, the refinement of the U-Net encoder can be further explored through the inclusion of other CNN architectures. The use of other segmentation architectures could be explored and doesn't have to be limited to an encoder-decoder structure. The use of these other models can also lead to other evaluation metrics such as memory usage and inference speed. Other datasets can be used to train and evaluate the proposed models as mentioned before, in order to focus on the robustness of any future proposed models.

The model can be expanded to scale up to the full application. Future studies could explore methods that would make the model more lightweight for edge applications, like quantization, and make the model more appropriate for implementation in hardware suitable for nanosatellites. In addition to this, a full-fledged control system could be developed that could extract information from the generated segmentation maps and be used to facilitate an image storage system for any collected satellite images.

A method could be determined to pool the results from patches belonging to the same image to represent the evaluation of a full-sized image with the pooled result. This could be done to retain the potential effect on the weight of the positional relationship between certain images on the final segmentation result. While the image could be analyzed more effectively, there is a possibility that splitting them up could affect the results at the edges in which the images were separated. Other patching constraints could be adjusted, like expanding the patch dimensions to include overlap between neighboring patches to avoid this problem.

References

[1] B. Denby and B. Lucia, "Orbital Edge Computing: Nanosatellite Constellations as a New Class of Computer System," in Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, New York, NY, USA, Mar. 2020, pp. 939–954. doi: 10.1145/3373376.3378473

[2] G. Furano et al., "Towards the Use of Artificial Intelligence on the Edge in Space Systems: Challenges and Opportunities," IEEE Aerospace and Electronic Systems Magazine, vol. 35, no. 12, pp. 44–56, Dec. 2020, doi: 10.1109/MAES.2020.3008468.

[3] S. Liu et al., "A Survey on CubeSat Missions and Their Antenna Designs," Electronics, vol. 11, no. 13, Art. no. 13, Jan. 2022, doi: 10.3390/electronics11132021.

[4] D. Selva and D. Krejci, "A survey and assessment of the capabilities of Cubesats for Earth observation," Acta Astronautica, vol. 74, pp. 50–68, May 2012, doi: 10.1016/j.actaastro.2011.12.014.

[5] G. Giuffrida et al., "CloudScout: A Deep Neural Network for On-Board Cloud Detection on Hyperspectral Images," Remote Sensing, vol. 12, no. 14, Art. no. 14, Jan. 2020, doi: 10.3390/rs12142205.

[6] Z. Zhang, G. Xu, and J. Song, "CubeSat cloud detection based on JPEG2000 compression and deep learning," Advances in Mechanical Engineering, vol. 10, no. 10, p. 1687814018808178, Oct. 2018, doi: 10.1177/1687814018808178.

[7] S. A. Chien et al., "Onboard Autonomy on the Intelligent Payload EXperiment (IPEX) Cubesat Mission: A pathfinder for the proposed HyspIRI Mission Intelligent Payload Module," 2014.

[8] T. I. Leong, Y. M. O. Abbas, M. A. C. Purio, and H. A. Elmegharbel, "Image Classification Unit: A U-Net Convolutional Neural Network for On-Orbit Cloud Detection Aboard CubeSats," in 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Jul. 2021, pp. 2807–2810. doi: 10.1109/IGARSS47720.2021.9553156.

[9] G. Giuffrida et al., "The Φ-Sat-1 Mission: The First On-Board Deep Neural Network Demonstrator for Satellite Earth Observation," IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1–14, 2022, doi: 10.1109/TGRS.2021.3125567.

[10] A. Maskey and M. Cho, "CubeSatNet: Ultralight Convolutional Neural Network designed for on-orbit binary image classification on a 1U CubeSat," Engineering Applications of Artificial Intelligence, vol. 96, p. 103952, Nov. 2020, doi: 10.1016/j.engappai.2020.103952.

[11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in Neural Information Processing Systems, 2012, vol. 25.

[13] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv, Apr. 10, 2015. doi: 10.48550/arXiv.1409.1556.

[14] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv, Apr. 16, 2017. doi: 10.48550/arXiv.1704.04861.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks." arXiv, Mar. 21, 2019. doi: 10.48550/arXiv.1801.04381.

[16] Z. Li, W. Yang, S. Peng, and F. Liu, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects." arXiv, Apr. 01, 2020. doi: 10.48550/arXiv.2004.02806.

[17] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." arXiv, Sep. 11, 2020. doi: 10.48550/arXiv.1905.11946.

[18] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation." arXiv, Mar. 08, 2015. doi: 10.48550/arXiv.1411.4038.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation." arXiv, May 18, 2015. doi: 10.48550/arXiv.1505.04597.

[20] H. Noh, S. Hong, and B. Han, "Learning Deconvolution Network for Semantic Segmentation," presented at the Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1520–1528.

[21] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[22] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, "Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations," in Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Cham, 2017, pp. 240–248. doi: 10.1007/978-3-319-67558-9_28.

[23] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 7, pp. 3523–3542, Jul. 2022, doi: 10.1109/TPAMI.2021.3059968.

[24] I. Demir et al., "DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Jun. 2018, pp. 172–17209. doi: 10.1109/CVPRW.2018.00031.

[25] P. Iakubovskii, "Segmentation Models Pytorch," GitHub repository. GitHub, 2019. [Online]. Available: https://github.com/qubvel/segmentation_models.pytorch