Image Segmentation: A Case Study on the TCD Dataset

Visal KAO, Sebora MEMOLLA

IMT Mines Alès

Abstract

This report presents a comprehensive study of image segmentation techniques applied to the TCD dataset :a curated collection of forest and ecological imagery provided by Hugging Face Restor. The dataset, annotated with detailed segmentation masks, geospatial metadata, and supplementary information, serves as a valuable resource for advancing ecological analysis and monitoring. Our approach leverages state-of-the-art deep learning models to perform pixel-level segmentation, enabling precise extraction of structural and semantic information from complex natural scenes. The experimental results demonstrate the potential of these techniques for applications in forest monitoring, biodiversity assessment, and land use management, and they highlight avenues for future research in automated ecological observation.

1 Introduction

Image segmentation, the process of partitioning an image into meaningful segments, is a fundamental task in computer vision with widespread applications ranging from medical imaging to remote sensing. In recent years, deep learning-based segmentation methods have significantly improved the ability to extract intricate details from complex images, making them well-suited for analyzing ecological data.

The TCD dataset, available on Hugging Face [1], is an ideal testbed for such studies. It comprises a diverse set of images depicting forested landscapes, accompanied by extensive annotations that include segmentation masks, bounding boxes, and metadata such as geolocation coordinates, biome classification, and image dimensions. These multimodal annotations not only support the development of robust segmentation models but also facilitate deeper insights into environmental structures and patterns.

In this project, our main objective is to develop and evaluate a segmentation pipeline that can accurately delineate key features within the TCD dataset. By applying advanced deep learning architectures, our goal is to capture fine-grained details that are critical for ecological analysis. This report details our methodology, including data preprocessing, model design, training strategies, and evaluation metrics, and discusses the implications of our findings for future research in environmental monitoring.

2 Related Work

This work is not new to this field. There are many published papers on image segmentation tasks. For instance, the ”OAM-TCD: A globally diverse dataset of high-resolution tree cover maps [1]” paper introduced by Veitch-Michaelis et al., authors introduced multiple methods to train segmentation models to work on the same dataset. Their works were exceptional, high-quality and reliable. However, they were training some heavy models such as UNet ResNet34, UNet ResNet50, SegFormer mit-b0, SegFormer mit-b5, etc. Although those models archieve such high accuracy, they are quite heavy. For example, according to ”SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers” paper[2], the smallest model of SegFormer mit series, the SegFormer mit-b0 has upto 3.7M parameters, while the heaviest one, the SegFormer mit-b5 has approximately 84.7 parameters.

The purpose of this paper is to train another custom model, which lighter weight, lesser parameters with UNET with the hope that we can reduce training costs, as well as reducing the size of the model, while trying to keep the accuracy and the quality of the model on par with the prior models.

3 Methodology

Our research focuses on developing an efficient segmentation framework for processing high-resolution images, with particular attention to robust data standardization, a tailored network architecture, and an adaptive training regimen. This section outlines the core methodological components that underpin our work.

3.1 Data Standardization and Preprocessing

The initial phase of our study centered on establishing a consistent and reliable dataset. Given the inherent variability in image sources and annotation quality, we employed a systematic preprocessing strategy. Images were uniformly converted to an RGB format and resized to 256×256 pixels. This resizing, performed with an interpolation method optimized for image quality preservation, ensured homogeneity across all samples. In parallel, annotation maps were converted to a grayscale format and resized using a nearest-neighbor approach, which maintained the integrity of the discrete label values. This preprocessing step was crucial, as it established a controlled environment that minimized data variability and facilitated effective learning.

3.2 Model Architecture

At the heart of our segmentation framework lies a modified UNet architecture from MONAI [3], which has been widely adopted in medical and natural image segmentation tasks due to its capacity to capture both fine and coarse features. Our implementation leverages a five-level encoder-decoder structure with progressively increasing feature channels. On the encoder side, Beginning with 16 channels at the shallowest level and expanding to 256 at the deepest layer, the network is designed to learn a hierarchy of representations that are critical for accurate segmentation. It's the exact same channels, except in reverse order for the decoder path. So in short, the architecture for the encoder is channels=(16, 32, 64, 128, 256), while on the decoder is channels=(256, 128, 64, 32, 16).

The architecture incorporates residual connections and dropout regularization to improve gradient flow and prevent overfitting, thereby enhancing the model's robustness in handling complex image features.

3.3 Training Strategy

A train-validation split of 80:20 was adopted to ensure that the model's performance was rigorously evaluated on unseen data. Our loss function of choice was the Dice Loss, selected for its direct correlation with the segmentation quality metric, thereby aligning the optimization objective with the evaluation criterion.

Initially, we set the maximum number of epoches to 150 epoches, with the option of early stopping. This is to prevent overfitting and the great number of epoches is to prevent in case the model can't converge fast enough.

We utilized Adam optimizer with an initial learning rate of **1e-3** to achieve a balance between rapid convergence and stability. Additionally, a learning rate scheduler was integrated into the training loop to dynamically adjust the learning rate based on validation loss trends, ensuring that the model could adapt to plateauing performance.

Early stopping was also implemented, with patience parameter = 7, to prevent overtraining and to retain the best model weights identified during the validation phase.

3.4 Evaluation and Metrics

A comprehensive evaluation framework was established to assess model performance across multiple dimensions. In addition to the Dice coefficient, which was used as the primary metric for segmentation accuracy, we also computed accuracy, F1 score, and AUC. The inclusion of these additional metrics provided a multidimensional view of the model's performance, ensuring that improvements in one metric did not come at the expense of others. During the validation phase, a sliding window inference strategy was employed to manage memory usage and to aggregate predictions across image patches, thereby yielding a robust performance assessment on full-resolution images.

4 Experimental Results

While the obtained results do not surpass state-of-the-art models, they demonstrate a reasonable trade-off given the computational constraints. The model achieved a validation loss of 0.7655, an accuracy of 0.7223, an F1 score of 0.6043, and an AUC of 0.7472. The best Dice score, recorded at epoch 23, was 0.5052, indicating moderate segmentation performance.

It is important to highlight that the model was trained in less than five hours using only

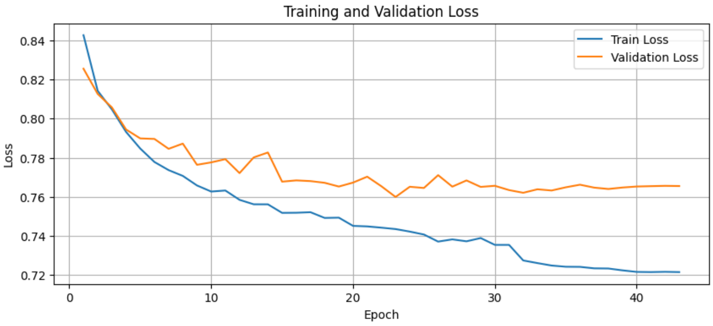

Figure 1: Training and validation loss

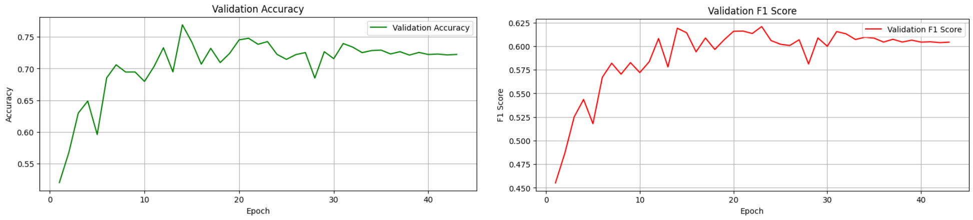

Figure 2: Validation accuracy and f1 score

a CPU, making it computationally efficient compared to more complex deep learning architectures that require extensive GPU resources. Furthermore, the relatively small model size ensures its practical applicability in resource-limited environments.

Despite these advantages, further hyperparameters tuning and architectural refinements are necessary to enhance segmentation performance. The graph below illustrates the validation loss, accuracy, and F1 score over the training process, providing insights into the model's learning behavior and potential areas for improvement.

5 Conclusion and Future Work

This study demonstrated the feasibility of training a lightweight model for segmentation with limited computational resources. The results, while not surpassing state-of-the-art models, indicate that the approach is on the right track. The model achieved a validation accuracy of 72.23% and an F1 score of 0.6043, with a best Dice score of 0.5052. Given that training was completed in under five hours on a CPU and with just 1,625,420 trainable parameters (a half of the size of the most lightweight model on the same dataset), this work highlights the potential for efficient segmentation models in resource-constrained environments.

Moving forward, the primary focus will be on increasing model depth. With only five layers, the current architecture may have reached its performance ceiling, leaving limited room for improvement through further tuning alone. Future work will explore deeper architectures to enhance feature extraction and segmentation accuracy while maintaining a balance between computational efficiency and model complexity, but we will leave it for the future improvement.

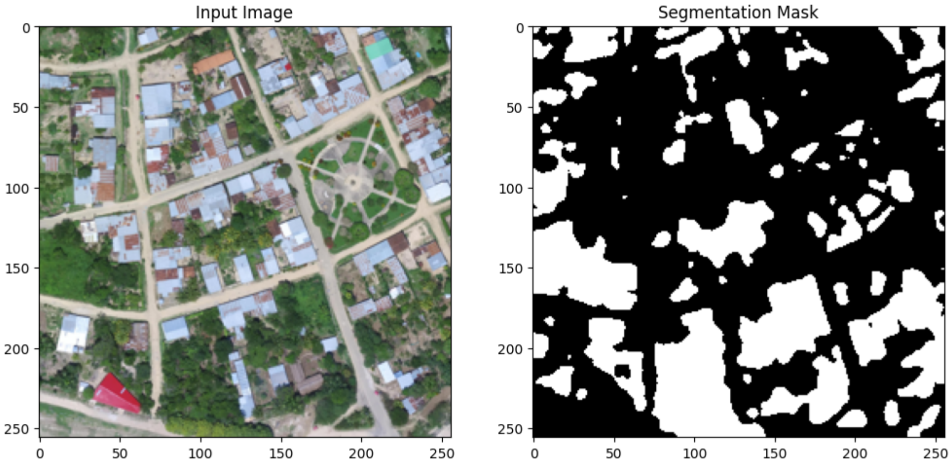

Figure 3: Masking results

Acknowledgments

We would like to express our gratitude to Mr. Binbin XU, our professor for insightful discussion without which this paper would not be possible.

References

[1] J. Veitch-Michaelis, A. Cottam, D. Schweizer, E. N. Broadbent, D. Dao, C. Zhang, A. A. Zambrano, and S. Max, "Oam-tcd: A globally diverse dataset of high-resolution tree cover maps," 2024. [Online]. Available: https://arxiv.org/abs/2407.11743

[2] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," 2021. [Online]. Available: https://arxiv.org/abs/2105.15203

[3] M. J. Cardoso, W. Li, R. Brown, N. Ma, E. Kerfoot, Y. Wang, B. Murrey, A. Myronenko, C. Zhao, D. Yang, V. Nath, Y. He, Z. Xu, A. Hatamizadeh, A. Myronenko, W. Zhu, Y. Liu, M. Zheng, Y. Tang, I. Yang, M. Zephyr, B. Hashemian, S. Alle, M. Z. Darestani, C. Budd, M. Modat, T. Vercauteren, G. Wang, Y. Li, Y. Hu, Y. Fu, B. Gorman, H. Johnson, B. Genereaux, B. S. Erdal, V. Gupta, A. Diaz-Pinto, A. Dourson, L. Maier-Hein, P. F. Jaeger, M. Baumgartner, J. Kalpathy-Cramer, M. Flores, J. Kirby, L. A. D. Cooper, H. R. Roth, D. Xu, D. Bericat, R. Floca, S. K. Zhou, H. Shuaib, K. Farahani, K. H. Maier-Hein, S. Aylward, P. Dogra, S. Ourselin, and A. Feng, "Monai: An open-source framework for deep learning in healthcare," 2022. [Online]. Available: https://arxiv.org/abs/2211.02701