# Enhanced Land Cover Classification: Advancing Few-Shot Semantic Segmentation with OpenEarthMap Data

Sugandha Roy Yating Wang Stuart Aldrich

## Abstract

*Motivated by the challenges of limited availability of labeled data, this project investigates the implementation of few-shot semantic segmentation in remote sensing using the OpenEarthMap dataset. We implement and incorporate multiple cutting-edge architectures, such as PSPNet with a ResNet-50 backbone and a Distilled Information Maximization (DIaM) framework. Additionally, we conduct additional experiments using DeepLabV3 and FCN-ResNet 50 models. We have adapted these models to suit specific needs of the task, optimizing performance across diverse terrains with minimal training data. We examined multiple models with aims to improve model robustness and accuracy across a variety of landscapes with minimal training instances. We addressed particular difficulties by modifying training and validation set to comply with the limitations of the OpenEarthMap Few-Shot Challenge. We found that there is significant area for improvement in data limited contexts, and while existing architectures work, they have difficulties adapting to some classes of data.*

## 1. Introduction

Our effort is focused on leveraging the OpenEarthMap dataset to address the issues related to limited labeled data in remote sensing, drawing inspiration from the OEM Few-Shot Challenge [6]. For many applications, such as urban planning, environmental monitoring, and emergency management, where accurate land cover maps are critical for efficient decision-making, semantic segmentation of such pictures is imperative. But the main obstacle is the lack of labeled data in various geographic locations, which makes it difficult to create reliable models.

By combining a PSPNet architecture with a Distilled Information Maximization (DIaM) framework [3], we successfully implemented our baseline model with a ResNet 50 backbone. Then, we applied DeepLabV3 to manage complicated image contexts at different scales [1]. In order to improve our model's predicted accuracy with the fewest possible examples, we also experimented with a 2-shot learning model to handle situations with incredibly little labeled data. Additionally, we trained our dataset on the FCN ResNet 50 [4] and Unet [5] models.

## 2. Related Work

Our work has been greatly impacted by recent developments in few-shot learning and semantic segmentation in the setting of remote sensing. Our methodology is based in part on the OpenEarthMap dataset and challenge, as reported by Xia et al. (2023), which offers a thorough framework for assessing semantic segmentation algorithms in few-shot scenarios [7].

Advances in convolutional network and training methodologies have improved the area of semantic segmentation significantly. The study by Chen et al. (2017) makes a substantial addition to this topic [1]. An enhanced technique for atrous convolution is presented in this research, allowing for effective multi-scale, resolution-preserving picture feature extraction. The network can include more context by widening the field of view of filters through the use of the atrous convolution approach, also called dilated convolution, without requiring an increase in computation or parameter count.

Our baseline approach Distilled Information Maximization (DIaM) is built on ResNet combined with the PSPNet architecture. In details, to improve the generalizability of the model with less labeled data, the DIaM by Hajimiri et al. (2023) focuses on optimizing the mutual information between features and predictions [3]. Utilizing PSPNet architecture-which is well-known for its efficacy in scene parsing tasks—this approach is incorporated into our study, as described by Zhao et al. (2017) [8].

Further important reference in our work is the Fully Convolutional Network (FCN) with a ResNet-50 backbone, as presented by Long et al. (2015) [4]. FCN benefits from strong features provided by the ResNet-50 backbone, which is renowned for its deep residual learning architecture. We are using this architecture in our project to enhance the model's learning effectiveness and flexibility when it comes to separating out various forms of land cover that are available in the OpenEarthMap resource.

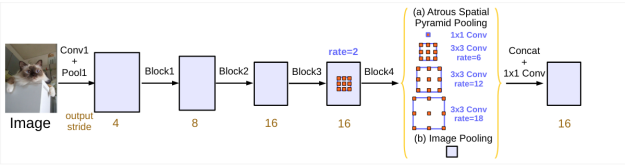



Figure 1. Illustration of the DeepLabV3 architecture, showing parallel modules with atrous convolution (ASPP), augmented with image-level feature [1]

# 3. Approach

## 3.1. DeepLabV3 Segmentation Model

For our segmentation task, we have adopted DeepLabV3 segmentation model [1]. It is built on the following modules:

**Atrous Spatial Pyramid Pooling (ASPP)**: It segments objects at multiple scales using atrous convolution at multiple rates. and consists of several parallel atrous convolutions with different rates, capturing multi-scale information. Batch normalization is also included in each branch to stabilize learning.

**Encoder-Decoder Structure**: It captures context and reduces spatial dimension. Recovers object details and spatial dimensions, often using transposed convolutions for upsampling.

**Cascade and Parallel Modules**: Atrous convolutions are applied in sequence to capture broader context progressively. Several atrous convolutions with different rates run in parallel to capture features at various scales simultaneously.

In our experimental setup, the DeepLabV3 model used a dataset partitioned into 250 training images and a validation set (pseudo-test set in this case) of 58 images, all equipped with ground truth masks for performance evaluation. Since all the ground truths from validation set were not publicly available, we went with using 58 images from training set as a pseudo-test set, so that we could compare our prediction maps visually and quantitatively. We resized the input images to $512 \times 512$ to standardize input size. Converted images to PyTorch tensors and normalized images using mean and standard deviation (of the RGB channels from the ImageNet dataset), which helps in faster convergence. Resized using nearest neighbor interpolation to prevent changing the class labels in the mask and applied custom transformation to convert the mask into a tensor directly from the PIL Image format, preserving exact label values. We utilized DeepLabV3 which is a pre-configured DeepLabV3 model from the segmentation models pytorch library. Specified the backbone of the model (e.g., resnet101), crucial for feature extraction. We chose pre-trained weights (imagenet), facilitating transfer learning which leverages learned features from a large and diverse dataset. Then we configured the model to accept 3-channel images and output the segmentation maps for 8 classes. Employed learning rate $1e-3$ and Adam optimizer. Moreover, we ensured the model runs on GPU if available for faster training. We used cross-entropy loss, ideal for multi-class classification problems like semantic segmentation.

## 3.2. Baseline Model: Distillied Information Maximization (DIaM)

DIaM uses the well-known PSPNet architecture (Figure 2) with a ResNet-50 backbone for feature extraction. This is a powerful combination for semantic segmentation, leveraging ResNet-50's deep residual learning and PSPNet's pyramid pooling to capture features at various scales. The ResNet-50 backbone serves as the feature extractor, converting input images into rich feature maps. The linear classifier, trained in a supervised manner on these features, is responsible for pixel-wise class predictions. The model is trained using standard cross-entropy loss, focusing on the base classes. This process does not require prior knowledge of novel classes, setting a solid foundation for the model's adaptability. Unlike other approaches, DIaM does not discard images containing novel classes during training, integrating them as background. This inclusion is key to DIaM's approach, allowing the model to generalize better to unseen classes during inference . At the core of DIaM's adaptation process is the principle of maximizing mutual information. This principle guides the optimization of the classifier during inference to ensure high-confidence predictions and a balanced distribution of class predictions, avoiding biases towards particular classes . DIaM employs KL divergence for knowledge distillation, ensuring that the updated classifier maintains consistency with the base model's predictions. This is critical for preventing catastrophic forgetting, where the model would otherwise lose its ability to recognize base classes after adapting to novel classes . By not removing novel classes during training, the model learns to consider these as part of the complex background. This means during inference, the model is better equipped to recognize and differentiate between truly novel objects and the typical background, leading to better generalization capabilities when encountering unseen classes.

## 3.3. FCN Resnet

For the FCN Resnet 50 approach, we experimented with both finetuned and training from scratch methods. Our finetuned model was the one provided by PyTorch, which was trained on COCO. While the OEM challenge provided some example testing code, we found it was inadequate for our needs and overspecialized for the challenge requirements. In both cases we adapted example code from PyTorch for

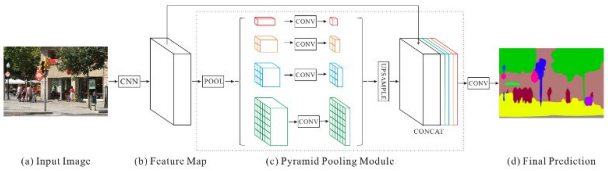



Figure 2. Spatial Pyramid Pooling Module: one of the building blocks of DIaM [8]

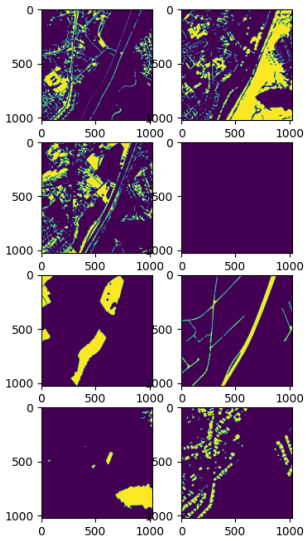


Figure 3. Ground truth for Resnet 50 test image

segmentation and adapted it for our specific dataset, model, and hyperparameter choices. We experimented with multiple optimisers, learning rate schedules, and epoch count. For providing the model training and test data on our small dataset, we used 208 images as training and withheld 50 as a test set from the provided baseline images from the challenge. For measuring the test set results, we used built-in IoU calculation functions to ensure a correct result consistent with other model measurements.

One of our initial tasks was getting the baseline code working for the OEM Challenge baseline example. We found that it was somewhat rigid to the changes we wanted to make and it was not well designed for testing locally. Namely that it was set up assuming no withheld data. Our approach to get something working was modifying it from 5-shot to 2-shot, as we had very limited data.

### 3.4. Unet

Unet is renowned for its effective encoder-decoder structure, which is particularly advantageous for tasks requiring precise localization and detailed contextual understanding from limited training data. Its architecture starts with a con-

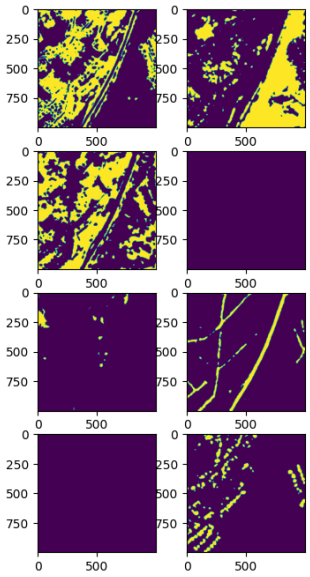


Figure 4. Thresholded predictions for Resnet 50 from scratch test image

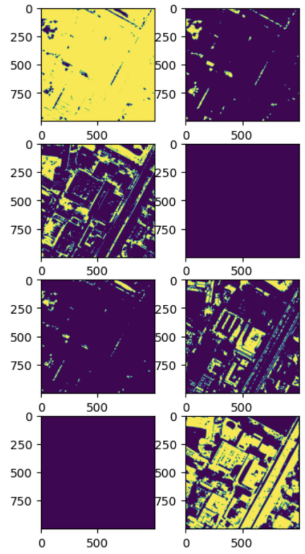


Figure 5. Predictions for Unet test image

tracting path to minimize dimensionality and capture context, and then a symmetric expanding path to allow for accurate localization. Unet is particularly well-suited for segmenting complex geographical features in aerial data because of its design, which enables it to capture both fine-grained details and high-level semantic information. We adapted example code of PyTorch in training a single epoch and modified the criteria and loss function to our dataset.

| Index | Name | Deep | ResFine | Res |
|---|---|---|---|---|
| 0 | background | 0.407 | 0.431 | 0.444 |
| 1 | tree | 0.357 | 0.480 | 0.484 |
| 2 | rangeland | 0.164 | 0.363 | 0.357 |
| 3 | bareland | 0.0 | 0.0 | 0.0 |
| 4 | agric land type 1 | 0.094 | 0.237 | 0.093 |
| 5 | road type 1 | 0.148 | 0.322 | 0.388 |
| 6 | sea, lake, & pond | 0.0 | 0.0 | 0.0 |
| 7 | building type 1 | 0.168 | 0.390 | 0.422 |
| | mean | 0.167 | 0.278 | 0.274 |

Table 1. IoU from DeepLabV3 (20 epochs), finetuned Resnet 50, and Resnet from scratch at 15 epochs

## 4. Results

For the DeepLabV3, the training process (Figure 6) was monitored by observing the loss over 20 epochs, as depicted in the provided graph. The training started with a loss of approximately 1.45. Throughout the epochs, there was a noticeable downward trend in loss, indicating that the model was effectively learning from the training data. Post the initial sharp decline, the loss plateaued around epoch 10 and showed minor fluctuations towards the later epochs.
The effectiveness of the model can also be visually assessed through the segmentation results: Figure 8a displays a high-resolution aerial photograph, detailing complex urban layouts. The ground truth (Figure 8b) provides a clear delineation of various segments like roads and buildings, serving as a benchmark for evaluating the predicted mask. Predicted Mask (Figure 8c) shows the segmentation results as predicted by the model. The predicted mask reveals that the model has been able to distinguish between various areas but with some areas of misclassification, particularly in densely built-up or overlapping regions.

In the evaluation of our DeepLabV3 segmentation model, a confusion matrix (Figure 7) was generated to visualize the model's performance across the various classes. Notably, the matrix revealed that Class 4 achieved the highest number of correct predictions (119,160), indicating a strong model performance for this class. Conversely, Class 0, while having a substantial number of correct predictions (83,723), also exhibited significant misclassifications, particularly being confused with Class 1, where 4,847 instances were incorrectly predicted as Class 1. The matrix further highlighted challenges in correctly predicting Classes 3 and 6. Other classes showed varied performance. The overall mean IoU of 0.1676 (Table 1), while indicative of the model's ability to segment certain classes, also highlights the challenges in achieving high-quality segmentation across all classes. This underlines the necessity for further model training optimizations.

For the Resnet 50 model, we found that some optimizers,

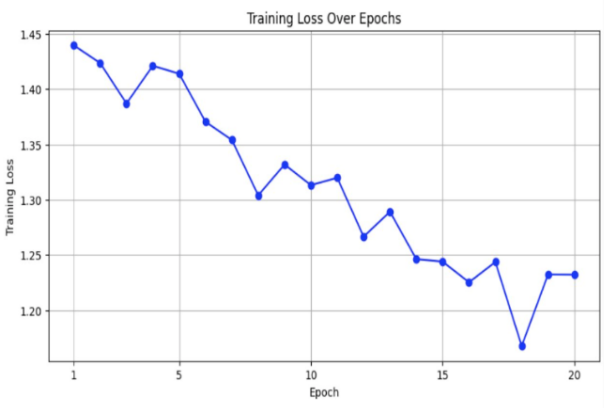



Figure 6. Training loss of DeepLabV3 model decreasing over 20 epochs, showing some fluctuations indicating the need for hyperparameter tuning

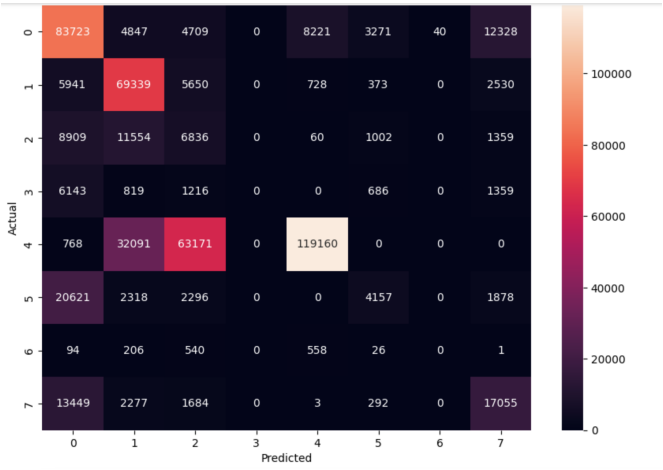


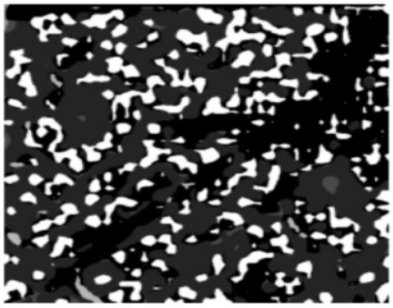

Figure 7. Confusion matrix

| | |
|---|---|
| 1 Tree | 0.0 |
| 2 Rangeland | 0.0 |
| 3 Bareland | 0.0 |
| 4 Agric land type 1 | 0.0 |
| 5 Road type 1 | 0.0 |
| 6 Sea, lake, pond | 0.0 |
| 7 Building type 1 | 0.0 |
| 8 Road type 2 | 0.0 |
| 9 River | 0.0356 |
| 10 Boat, ship | 0.00 |
| 11 Agric land type 2 | 0.0393 |
| Base mIoU | 0.00 |
| Novel mIoU | 0.0187 |
| Average of Base-and-Novel mIoU | 0.0094 |

Table 2. Results for baseline 2-shot

(a) Input image

(b) Ground Truth



(c) Prediction map

Figure 8. Visualization from DeepLabV3 segmentation model. (a) Input image, (b) Ground truth, (c) Prediction map obtained over 20 epochs of training paradigm

will not train at all for the task of segmentation. We additionally found that a very high learning rate of 1.0 is actually useful for retraining a network not trained on satellite imagery. We suspect that given the network was trained on ground imagery, and we reused the existing network head, it required fairly strong changes to the network, even if the convolutionial filters stayed the same. Additionally, it required quite a few epochs to get a good result. However interestingly, we found that the from scratch and finetuned had comparable performance within the first few epochs. Our best setup for producing a good IoU was using the Adadelta optimizer with a learning rate of 1.0, decreasing by 10x every 3 epochs.

We found that for the 2-shot learning from the baseline model it did learn, but only slightly. With overall very poor results with only two classes having an IoU greater than zero, and only in the single digits. Our supposition is that it is both limited by a small number of epochs and a very small dataset. Unfortunately reproducing results from the github repo for the OEM challenge is not possible at this time, as the ground truth labels have not been released for the novel classes outside the 5-shot training data.

We also have a tried on the Unet model. The mean IoU of Unet on the validation set is 0.31. The visual results (Figure 5) suggest that while Unet can capture significant areas of interest, such as roads and building structures, it struggles with consistency and finer details, as seen in the scattered and misclassified pixels. This discrepancies might stem from insufficient training data, inadequate model complexity, or suboptimal tuning of hyperparameters. Channels that require high-resolution detail retention seem to underperform, highlighting a possible need for enhancing the model's depth or employing more sophisticated feature extraction techniques like atrous convolutions.

## 5. Discussion and Conclusions

For the DeepLabV3 model, the experiment was designed to evaluate the model's ability to accurately segment aerial images into relevant classes, such as roads, buildings etc.

Comparing the predicted mask with the actual mask, the model demonstrates substantial accuracy in identifying and segmenting various structures within the image. However, discrepancies in the form of misclassified regions or blurred boundaries are evident, suggesting that the model may benefit from further training or more complex data augmentation to improve its ability to handle spatial intricacies. The high accuracy observed for Class 4 suggests that the model features and training data for this class are well-optimized, leading to effective learning and prediction. However, the substantial misclassifications for other classes raise concerns about the feature overlap or the model's sensitivity to distinguishing between these classes (Figure 7). The DeepLabV3 model has shown potential in the semantic segmentation of aerial imagery as part of the OpenEarthMap challenge. The training process, evidenced by the decreasing loss trend, indicates successful learning, albeit with room for improvement in model stability and prediction precision.

Further steps to enhance the model's performance could include: more training data could help the model generalize better over diverse scenarios. Performing hyperparameter optimization by tweaking learning rates or employing learning rate schedules could address the observed fluctuations in loss. Implementing advanced DeepLabV3+ [2] might help in getting better segmentation results.

The model's current training and validation strategy does not account for the potential variability and diversity of real-world scenarios where unseen/novel classes might emerge. This lack of exposure can hamper the model's ability to generalize well beyond its training conditions. To mitigate this, incorporating techniques such as few-shot learning, or domain adaptation could be explored that is tailored to handle novel classes. While the DeepLabV3 model has shown good potential, aforementioned adjustments could unlock higher levels of accuracy and make the model more robust against diverse and complex geographic features present in aerial images.

For both models we trained, while we did not achieve the intended five-shot task, we still managed to get decent performance out of models with only a very small training dataset of around 200 items. While we would have attempted training on the OEM validation set, due to difficulties in getting our models working until very close to the deadline, resulted in limited time for additional work. However we are happy with getting results that are comparable with the OEM baseline of approximately 0.30.

In retrospect, it might have been easier for us to choose a project based on a better established dataset without intentionally withheld data. We ended up spending a lot of time on getting data just in the format we wanted and had very limited documentation. As a result, there was very limited model development time beyond just debugging basic issues with not learning due to data formatting problems. Overall we somewhat bit off more than we could chew given our prior experience with computer vision machine learning. The upside of this project choice was a lot of learning the basics in detail, for example, how some common presets for learning rate don't result in learning and the model is just stuck in some contexts. If we had more time, we would have added an attention mechanism for the DeepLabV3 model, as that might have allowed it to focus on the more important parts of images for creating the segmentation map. Additionally, we would have wanted to experiment with diffusion models, as their ability to transform input into completely visually different output might have allowed us to show something new and different from other segmentation architectures, especially since we would have tried using multiple output channels for a diffusion model, which seems somewhat uncommon. We believe that exploring these aforementioned ideas would be interesting and could potentially lead to better segmentation results.

## 6. Statement of Individual Contribution

Our project essentially split into two to three subprojects (with each focused on one model type) each subproject handled by one person with cross collaboration when someone got stuck.

- Data collection: Stuart and Sugandha
- Report writing: Everyone
- Related work/literature review: Yating
- Group coordination: Stuart and Sugandha
- DeepLabV3 coding/training: Sugandha
- FCN Resnet coding/training: Stuart
- Unet coding/training: Yating

## 7. External Resources Used

For running code and generating results, we ended up using Google Colab for all final results. There was some limited usage of WUSTL HPC for prototyping. For running the OEM baseline code, we utilised one of our own desktop computers. For libraries we used PyTorch for training networks. Matplotlib for visualations. Some Numpy, PIL, and OpenCV features for data processing. For external code, we used the example OEM Challenge starter code for the 2-shot, but with a few modifcations to support 2-shot rather than the default 5-shot. For the Resnet 50 we used existing Pytorch examples and modified them a bit to support our dataset correctly. For the DeepLabV3, we followed the coding structure and implementation guidelines provided in the PyTorch documentation on DeepLabV3. For the U-Net we referred to the code of the original paper and modified the loss and criteria function. Our GitHub

## References


[1] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1, 2

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611:http://arxiv.org/abs/1802.02611, 2018. 5

[3] Sina Hajimiri, Malik Boudiaf, Ismail Ben Ayed, and Jose Dolz. A strong baseline for generalized few-shot semantic segmentation. In *CVPR*, pages 11269–11278, 2023. 1

[4] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 1

[6] Junshi Xia, Naoto Yokoya, Bruno Adriano, and Clifford Broni-Bediako. Openearthmap: A benchmark dataset for global high-resolution land cover mapping. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6254–6264, January 2023. 1

[7] Junshi Xia, Naoto Yokoya, Bruno Adriano, and Clifford Broni-Bediako. Openearthmap: A benchmark dataset for global high-resolution land cover mapping. In *WACV*, pages 6243–6253, 2023. 1

[8] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, pages 6230–6239, 2017. 1, 3