AN2DL - Second Homework Report
Alpaca

Ernesto Natuzzi, Flavia Nicotri, Luca Pagano, Giuseppe Vitello

ernesto, flanico, lp1807, peppisparrow

251284, 251671, 249359, 251129

December 14, 2024

1 Introduction

This project focuses on multiclass semantic segmentation of **64x128** grayscale images of Mars terrain, with each pixel labeled into one of five categories: background, soil, bedrock, sand, or bigrock. The goal is to generate a pixel-wise mask for each image. The dataset includes a `training_set` of 2615 labeled images and a `test_set` of 10022 unlabeled images.

2 Data Analysis

Firstly, we decided to remove the outliers in the dataset, specifically images of aliens. These were identified and filtered out based on their masks, which were identical across all such images. In the end we had a dataset composed of 2505 labeled images with the following distribution:

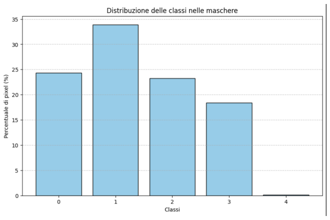


Figure 1: Distribution of classes

3 Experiments

3.1 First model and its evolution

We began with a standard **U-Net** [1] to establish a baseline for comparison, which, however, yielded poor results. Subsequently, we worked to enhance it by incorporating well-known techniques used in modern architectures. After multiple iterations, validating and testing on a subset of data, we decided to implement a U-Net with **residual blocks** [2] not only in the bottleneck but also in all the layers of the encoder and decoder. For upsampling (implemented using `Conv2DTranspose` instead of `UpSampling2D`), we applied **attention** [3] gates between symmetric upsampling and downsampling paths. We trained this model using an **hybrid loss** composed of `CategoricalCrossEntropy` and a `DiceLoss`.

3.2 Cut-and-Paste augmentation

To address the underrepresentation of the **"bigrock" class** in our dataset, we employed a Cut-and-Paste augmentation technique [4]. We used an algorithm to extract patches containing the "bigrock" objects and their corresponding masks, which were then inserted into different portions of the original images and masks. Although this approach improved the model's ability to recognize "bigrock", it did not lead to performance improvements on the test set. This suggested that the this class remained underrepresented in the test data as well, prompting us to discontinue this strategy.

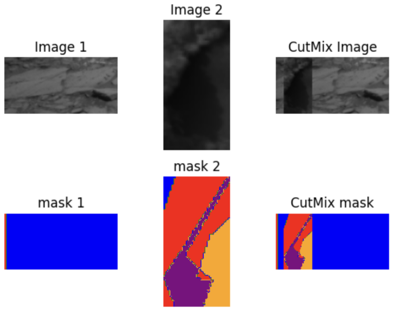


Figure 2: Cut-and-Paste Image

3.3 Post-Processing on predictions

Since the **MeanIoU** formula ignores class zero, we applied post-processing on X_test predictions for this initial model. Replacing the background with the most frequent label improved results by **2%** locally and online but reduced mask accuracy in some areas. To address this, we used SciPy's binary dilation to expand background-adjacent pixels, achieving better results (+**11%**), reaching a **67.09%** MeanIoU.

3.4 Double Enhanced U-net

Based on the results from non-parametric background removal, we designed a second network to refine the masks from the first model by replacing the background with another class. This second model, identical to the first, was tested with various connection strategies.

Our initial attempt involved providing the second model with the mask produced by the first model after the `Softmax` layer and then concatenating the bottleneck of the first model with that of the second to provide the latter with additional contextual information from the source image. This initial configuration achieved a MeanIoU score of **68.45%**, already outperforming the best non-parametric method.

Subsequently, we experimented with different modifications to the connection points between the two models. In particular, we explored various techniques for merging the two bottlenecks; however, in our case, no alternative proved superior to simple concatenation. Ultimately, the best-performing 2-model configuration included three connection points:

- **Input:** The input to the second model was the mask produced by the first model after an `Argmax` layer. Notably, the `Softmax` output probabilities introduced excessive noise and did not help the second model make better decisions regarding alternative background classes.

- **Bottleneck:** As previously mentioned, the bottleneck connection consisted of a simple concatenation of the encoder outputs from both models.

- **Skip Layers:** In the decoder of the second model, we incorporated attention gates for connections to both the encoder of the first model and that of the second. The outputs of the two attention blocks were then concatenated.

To train this model, we initialized its first component with pre-trained weights obtained from a separate training phase. The second component was then trained independently while keeping the first component frozen to preserve its learned representations. For the second phase of training, we employed a weighted `FocalLoss`, designed to ignore the background class and assign increased importance to the underrepresented "big rock" class, thereby addressing the class imbalance effectively. Any change in this configuration resulted in same or worse performance. With this final configuration, we achieved a MeanIoU score of **74.39%**.

3.5 Initialization with Autoencoder

Since we had access to a test set of 10,000 images—nearly **20 times** the number of labeled images—we aimed to exploit this advantage. An autoencoder [5] was trained on 12,527 images in an unsupervised manner, ensuring a reconstruction accuracy of $\geq 85\%$ before discarding the decoder. Encoder weights were then used to initialize the model and reduce overfitting. While training was accelerated, predictions did not improve over the previous model.

3.6 Pseudo-Labeling

Drawing inspiration from an approach studied in another course, *Recommender Systems*, we leveraged our dual-network setup to train the second U-Net on its predicted outputs from the test set. This technique takes the name of **Pseudo-Labeling** [6]. Although this method seemingly introduces significant noise, it actually resulted in a two-percentage-point improvement in both validation and public test scores, achieving approximately **76.10%** on the public test set. We then repeated this process by retraining the model on the dataset augmented with the test set predictions and iterated this process for a few epochs. Finally, we re-predicted the test set with the updated model and reintegrated these predictions into the training data, continuing this cycle until the performance reached a plateau.

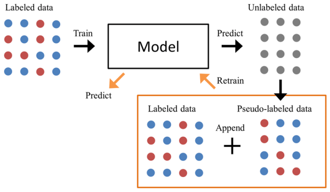
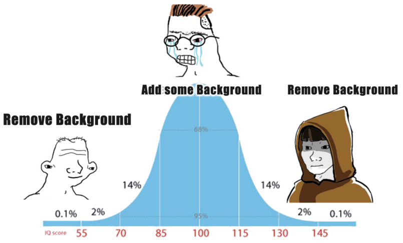


Figure 3: Pseudo Labeling

3.7 Filtering on background

Although the mean IoU formula ignores the background class, replacing the background with other labels can reduce the score by increasing the union's denominator. To address this, we introduced a post-processing step where the first model's background prediction was retained if it exceeded a certain threshold. However, despite tuning the threshold, this approach consistently underperformed compared to the second model.



3.8 Test Time Augmentation

Test-Time Augmentation (**TTA**) [7] improves predictions by applying transformations (e.g., flips, rotations, scaling) to the input image during inference. The model generates predictions for each augmented version, which are then transformed back and aggregated (e.g., averaged) to produce a final, more robust output. We decided to employ this technique to further improve our results locally and online test. We employed only three augmentations since others were not beneficial for our case: identity (to keep the same images), horizontal and vertical flip. Then we predicted each transformed X_test, used the model to predict it, and finally averaged all results. We achieved through this technique a final public score of **77.48**%.

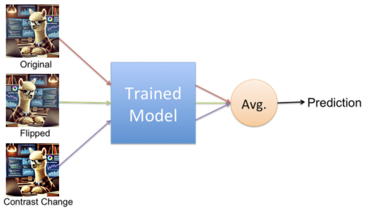


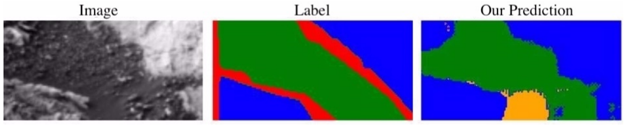
Figure 4: Test Time Augmentation

4 Results

Model/Configuration	**MeanIoU**
First model	42.64%
Augmented model	55.11%
Binary Dilation	67.09%
First Double U-Net	68.45%
Final Double U-Net	74.39%
PseudoLabel	76.10%
TTA	77.48%

Table 1: Model Performance Comparison

5 Conclusion



We achieved excellent results by fully leveraging all the data available to us. Further improvements could be made by refining the pseudo-labels, perhaps by filtering them based on a confidence value, such as averaging the softmax outputs, and applying a threshold to retain only the most reliable predictions. This way we could have use only the pseudo-labels that don't introduce noise on the dataset, in order to have a resilient model.

References

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

[4] Author(s). Evaluating the efficacy of cut-and-paste data augmentation in semantic segmentation for satellite imagery. *arXiv*, 2024.

[5] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. 1986.

[6] Carla P. Cascante-Bonilla, Adrien Gaidon, Kevin Murphy, Ubong U. Inyang, and Teng-Yok Lee. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6670–6678, 2021.

[7] Divya Shanmugam, Davis W. Blalock, Guha Balakrishnan, and John V. Guttag. When and why test-time augmentation works. *CoRR*, abs/2011.11156, 2020.