AN2DL - Second Homework Report
Deepmindset

Giacomo Giovanni Papa, Marta Zecchini, Leonardo Salvucci, Alessandro Torazzi

giacomopapa, martazecchini, leonsalvu, atorazzi

252855, 246790, 259585, 252725

December 14, 2024

Contents

1 Introduction

The objective of this project was to **design and train a deep learning model** for *semantic segmentation* of Mars terrain images. The dataset comprised 2,615 grayscale images (64x128 pixels) with corresponding masks classifying pixels into five distinct classes: **Background** *(Class 0)*, **Soil** *(Class 1)*, **Bedrock** *(Class 2)*, **Sand** *(Class 3)*, and **Big Rock** *(Class 4)*. The goal was to develop a model capable of accurately predicting the class of each pixel in the test set, thereby generating segmentation masks that closely resemble the ground truth.

2 Data Pre-processing

The data pre-processing pipeline was designed to ensure that the dataset was both clean and representative for training the network. The main steps included data inspection, label refinement, and addressing class imbalance.

2.1 Dataset inspection

Initially, we inspected the dataset by visualizing a selection of sample images, followed by applying *PCA* to identify potential outliers and using the *Mahalanobis distance* to detect anomalies. This analysis uncovered several images with alien artifacts, all associated with the same mask. These flawed images were removed, reducing the dataset to a total of 2,505 images.

Further analysis of the dataset revealed a significant imbalance in class occurrences, with **class 4** (*big rock*) being **highly underrepresented** and frequent label inaccuracies, primarily affecting this class.

2.2 Label refinement

Firstly, we addressed these significant label mismatches that would have hindered effective network training. We spotted approximately 75 images (30 of which contained class 4, as detailed in Section 2.3) and manually corrected their labels.
To achieve this, we used MATLAB's **roipoly()** command (see Figure 1), which allows users to manually define polygonal regions of interest (ROIs) on an image. In our process, we drew polygons only in areas where we were confident that a specific rock class was present but not properly labeled. The corresponding regions within the mask matrix were then assigned the value of the identified class. This ensured that the new labels accurately reflected the presence of specific rock types, improving both the quality and consistency of the dataset. The refined labels were subsequently integrated into the dataset to support more effective network training.

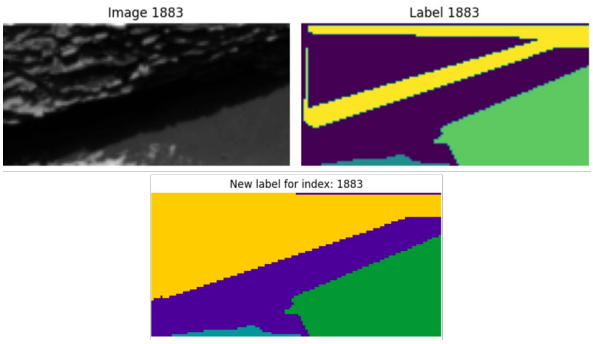

Figure 1: Comparison between the original and new label for one image

2.3 Addressing class imbalance

Following label refinement, light geometric data augmentation techniques were applied exclusively to the 30 images with corrected labels, duplicating them multiple times to expand the subset of class 4 images. This process improved the representation of the underrepresented class and enhanced the overall quality of the training samples, contributing to better dataset balance and increased the model accuracy for this class.

2.4 Other attempts: Exploring maps

To improve the detection of contours and regions of interest, we initially generated edge maps using the Sobel and Canny algorithms to outline rock boundaries. However, their effectiveness was limited by the low quality of the images. Inspired by approaches like DiffSeg for unsupervised segmentation [3], we further experimented with thresholding and watershed segmentation maps. Unfortunately, these methods also failed to yield meaningful improvements or provide additional insights for network training.

3 Data preparation and augmentation

Next, the dataset was split into training and validation sets, with images normalized to a range of [0, 1]. A light augmentation pipeline was applied exclusively to the training set to enhance the model's generalization, as it proved sensitive to heavy augmentations. After several tests, the most effective technique was the application of random horizontal and vertical flips, while other augmentations, such as contrast, zoom, and brightness adjustments, did not perform well in our specific case.
Special care was taken to apply geometric transformations consistently to both images and their corresponding labels. After these steps, the training and validation datasets were finalized for use in training the network.

4 Model construction and adopted strategy

4.1 Model Implementation

We implemented a **U-Net** architecture featuring an encoder-decoder structure with skip connections, which preserve spatial information while progressively learning hierarchical features. The encoder extracts high-level features through three convolutional blocks followed by max pooling, while the decoder restores spatial resolution using upsampling layers and merges feature maps from the encoder via concatenation. The bottleneck, composed of 256 filters with L2 regularization, captures the most abstract features while mitigating overfitting. Finally, the output layer applies a 1×1 convolution with softmax activation to predict pixel-wise class probabilities. Each convolutional block is constituted by a two-time sequence of 3x3 convolutions, batch normalization, and ReLU activation. Every convolutional layer is initialized with He-Normal initialization to facilitate training with ReLU activation.
Regarding the upsampling process, we initially experimented with transpose convolutions using a kernel size of 3×3 and a stride of 2. However, this approach was refined based on insights from [1], which highlighted that transpose convolutions often produce "checkerboard artifacts" that can degrade the quality of the predicted mask. To mitigate this issue, the article suggested splitting the transpose convolution layer into two components: an upsampling step and a standard convolution layer. For the upsampling step, a *"nearest-neighbor"* strategy was recommended instead of the default 'Bed of Nails' approach typically used in transpose convolutions. This adjustment produced more uniform results.
Consequently, we adopted this improved method, placing the upsampling block before the concatenation of the encoder and decoder feature maps.

4.2 Model training

The model was trained using the *Adam optimizer*, which proved to be the most effective compared to other optimizers we tested, such as SGD and AdamW. To ensure efficient convergence and prevent overfitting, we implemented two callbacks: EarlyStopping and ReduceLROnPlateau. These adjustments were particularly beneficial, as we observed that smaller learning rates improved the network's performance metrics on the validation set.
A key aspect of our project was the choice of the loss function. The adoption of *Categorical Focal Crossentropy*, combined with a careful selection of class weights, resulted in a nearly 20-percentage-point improvement in the test set score.
Initially, our model delivered unsatisfactory results. The turning point came when we assigned a weight of 0 to background's class. We observed that this class contained external elements, such as machinery and portions of terrain that were difficult to classify accurately, requiring more precise labeling than the one we were provided.
Furthermore, the use of the mean IoU metric, which excludes the background class from evaluation, reinforced the decision to remove it from the loss computation.
Starting with the weights computed using the *class weights function*, we conducted an extensive search to identify the optimal values, fine-tuning them to maximize validation accuracy. Ultimately, we settled on the final weights of [0, 4, 5, 5.5, 50]. This adjustment had a substantial impact, increasing the mean IoU from 0.49 to 0.65.

4.3 Second Model and Ensembling

Pleased with the results of our first model, we tried building a different one with the aim of adopting a strategy inspired by Stacking Ensembling. Our

second model main characteristic was the implementation of an Atrous Spatial Pyramid Pooling block in the bottleneck, taking inspiration from [2]. Characterized by parallel Atrous convolutional layers with different dilation rates followed by Global Average Pooling, we thought that ASPP could have been useful to capture more generalized informations, allowing the bottleneck to work with different receptive fields. Clearly, this addition changed the whole model, thus requiring other minor modifications, such augmented regularization, as well as hyper-parameters fine tuning.
With the predicted probabilities from the two models on the test set, we computed the final output by summing the probabilities from both models and applying the *np.argmax* function to determine the final class predictions.
This approach, despite missing the additional meta model usually present when adopting stacking ensemble, still brought visible improvements, allowing the capture of different informations about our terrains, thus increasing prediction robustness, enhancing prediction robustness and achieving our top score of 72.7%.

4.4 Other Attempts

In our search for the optimal model we had many unsuccessful findings.
Having read about the generalizing power of Atrous Convolutions [4], useful to increase the receptive field of the net, we effectively applied them in the ASPP block, but we weren't able to apply them outside of the bottleneck, despite several trials.
Moreover, we decided to try implementing Test Time Augmentation, a technique that applies augmentations to images during the testing phase, generating multiple augmented versions of each image. The predictions for the pixels of these augmented images were then averaged to produce the final probabilities. Unfortunately TTA didn't give us significant results.

5 Discussion

Despite achieving satisfactory results for the semantic segmentation task, there is still room for improvement. One of the main limitations lies in the imprecision of the training labels. More accurate and detailed masks would likely enable the network to better extract and learn meaningful features (As demonstrated in: [5]); despite our efforts, the limited time at disposal wasn't sufficient to improve the dataset whole. Additionally, the limited size of the dataset posed a significant challenge, constraining the model's ability to generalize effectively. Given the ambiguous nature of class zero and the limited number of images, it would be beneficial to analyze the dataset further to attempt extracting valuable information from the uncertain regions labeled as class zero.

6 Conclusions

The most impactful improvements in our solution were achieved by adjusting the class weights, modifying the dataset labels and implementing our two model solution with the add of ASPP, leading to a final mean IoU of 0.73 on the test set.
Future directions for improving the performance and expanding the scope of this work include:

- **Map Applications:** Exploring how the generated segmentation maps can be utilized to enhance the dataset or improve model performance. These maps could be used to augment the dataset by providing additional annotated regions, or they could serve as supplementary inputs to the network, offering enriched spatial or contextual information to guide the learning process.
- **Creating More Accurate Masks:** Implementing algorithms like k-clustering to refine the training masks, thereby improving data quality and model performance.

Contributions

All team members contributed to the project, with **Marta Zecchini** and **Giacomo Giovanni Papa** focusing primarily on data pre-processing and enhancement, while **Leonardo Salvucci** and **Alessandro Torazzi** concentrated on model development, architecture exploration, and post-processing. Everyone collaborated on report writing and finalizing the results.

References

[1] C. O. Augustus Odena, Vincent Dumoulin. Deconvolution and checkerboard artifacts. *Distill*, 2016. https://distill.pub/2016/deconv-checkerboard/.

[2] N. Das, S. Saha, M. Nasipuri, S. Basu, and T. Chakraborti. Deep-fuzz: A synergistic integration of deep learning and fuzzy water flows for fine-grained nuclei segmentation in digital pathology. *PLoS ONE*, 18(6):e0286862, 2023.

[3] K. Kejriwal. Diffseg: Unsupervised zero-shot segmentation using stable diffusion, 2023.

[4] Y. Lv, H. Ma, J. Li, and S. Liu. Attention guided u-net with atrous convolution for accurate retinal vessels segmentation. *IEEE Access*, 8:32826–32839, 2020.

[5] L. Torresin. Sviluppo ed applicazione di reti neurali per segmentazione semantica a supporto della navigazione di rover marziani. Tesi di laurea magistrale, Università degli Studi di Padova, 2019. Relatori: Stefano Debei, Marco Pertile, Sebastiano Chiodini.