POLITECNICO
MILANO 1863

AN2DL - Second Homework Report
Spanish Inquisition

Miguel Planas, Manuel del Carmen Fernández, Rayan Emara, Emanuele Paesano

miguelplanas, manuelferfer, rayanemara, emanuelepaesano

276442, 276383, 260145, 221974

December 14, 2024

1 Introduction

This project involves building a semantic segmentation model to classify Mars terrain images into five classes, including *background, soil, bedrock, sand,* and *big rock*, emphasizing pixel-level precision and model development from scratch.

2 Problem analysis

Our first day was spent analyzing the data, utilizing a combination of t-SNE [6] and a one-class SVM [5] to detect outliers.

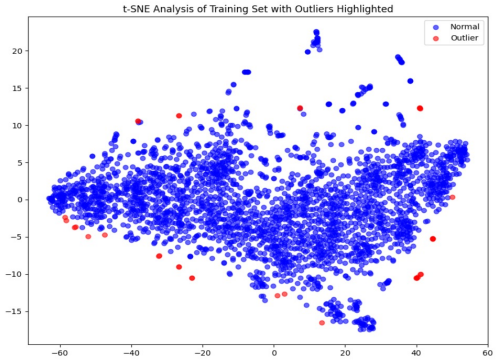

Figure 1: Outliers.

The one-class SVM, an unsupervised model, estimates the relative distance of each sample from the decision hyperplane using its decision function. This distance effectively acts as a threshold, allowing us to sort the images based on their proximity to the hyperplane. By identifying those samples furthest from the decision plane, we flagged the most distinct outliers for removal. This approach ensured that anomalous images, such as those containing alien-like patterns, were efficiently identified and prioritized, while the terrain data remained untouched.

Hash Computation and Duplicate Removal: Perceptual hashes[4] were computed using the `imagehash` library to detect duplicates, which were then removed.

The cleaned dataset, reduced from 2,615 to 2505 unique, noise-free images, including **image masks** and **original images**, was saved as `training_data_clean.npz` for reproducibility. (**Aliens are not real, well in the dataset at least**)

3 Experiments

3.1 First ideas

Our initial experiment served as a baseline for further development, using a basic U-Net architecture with cross-entropy loss and minimal data augmentation. The encoder-decoder design included two downsampling blocks, a bottleneck, and two upsampling blocks, achieving a score of 0.41. This

framework set the foundation for subsequent improvements.

We also tested a custom U-Net-inspired architecture that initially achieved a mean IoU of 0.43. However, it proved suboptimal as the project progressed, leading us to adopt a more scalable Atrous Spatial Pyramid Pooling (ASPP) architecture.

After experimenting with various loss functions, we adopted a combined loss of structural similarity score (SSIM)[7], focal loss, and dice loss. Focal loss parameters were tuned to address class imbalance, and weighted dice loss leveraged class frequencies to prioritize rare classes. Assigning near-zero weight to the background class (ignored during evaluation) significantly boosted performance. Attempts to use learnable loss weights led to inconsistent convergence, so we manually set weights based on model behavior.

3.2 Atrous Spatial Pyramid Pooling

When inspecting the predictions of this model we found regions where many different classes were being predicted in a small space. This made us look for an approach to control the scale at which features are extracted, at first we implemented a post processing layer, this helped class compactness by essentially blurring the output. This solution didn't satisfy us as it wasn't a learnable parameter and therefore would create a sort of bottleneck we couldn't improve upon. This led us to DeepLab-like implementation using an Atrous Spatial Pyramid Pooling (ASPP) [2][1] block, which is a technique that helps to capture features at different scales. This block includes several branches that process the image: The first branch, a 1x1 convolution without dilation, captures local features since it doesn't have gaps between the filters. The other three branches apply 3x3 convolutions with different dilation (spaces are left between processed pixels by the filters) rates (6, 12, and 18), allowing the model to capture features at different scales without reducing the image's spatial resolution.

These dilated convolutions "expand" the filter, covering a larger area without the need to downsample the image. Additionally, a global average pooling branch is included, which performs a global summary of the image by averaging all the values in each channel, helping to capture global contextual information.

Then, their outputs are combined using concatenation. The result is an image representation that combines local, medium-scale, and global features. To ensure the outputs are the same size and can be combined properly, the output from global average pooling is resized to match the dimensions of the other branches. Afterward, a 1x1 convolution is applied to reduce the number of channels, followed by Batch Normalization. Finally, a ReLU function is applied to introduce non-linearity in the combination of features. Thanks to this the model went up in performance achieving a 0.51.

We also tested an implementation based on BasNet which did yield slightly better results but, being a multi output model originally meant for binary semantic segmentation on much bigger images, ended up being incredibly slow on both training and inference, rendering quick iteration difficult and costly given our limited resources.

3.3 Augmentation

We improved our data pipeline with an augmentation step using `Albumentations`, starting with simple techniques like flips and rotations. Later, we added advanced augmentations like scaling and shift-scale-rotate operations. This also helped with overfitting.

We attempted to implement Cutmix [2][8]. However, due to possible implementation errors or incompatibility with the model, it led to a decrease in accuracy.

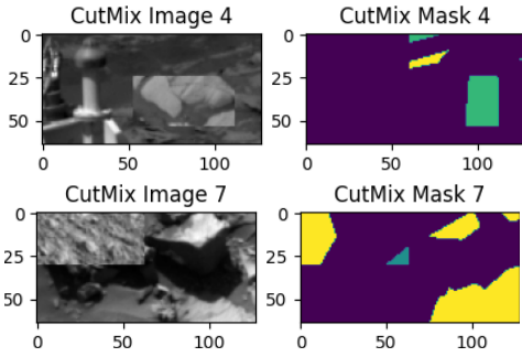

Figure 2: Example of Cutmix Augmentation.

3.4 Conditional Random Fields

To further refine the segmentation results, we incorporated a Conditional Random Field (CRF) [3] as a post-processing step. CRFs are probabilistic

graphical models that improve pixel-level segmentation by considering the spatial and contextual relationships between pixels. In our case, the CRF used both a Gaussian pairwise term to enforce spatial smoothness and a bilateral term to preserve fine details, such as object boundaries, by incorporating both spatial and intensity information from the original image. This helped reduce noise in the predictions and improved the compactness of segmented regions. While CRF is not a learnable layer, its integration as a post-processing step significantly enhanced the accuracy of the final segmentation maps, particularly in areas with ambiguous class predictions.

4 Final Model Architecture

Finally we came up with the model in Figure 3. In which we can highlight that it is an image segmentation network based on the U-Net architecture with an additional ASPP block to improve feature capture at different scales. The architecture can be divided into these separated parts:

Encoder: Reduces the image resolution through convolutional layers and max pooling.

ASPP: Uses convolutions with different dilation rates to capture features at various scales.

Decoder: Increases the image resolution using transposed convolutions and skip connections to preserve details.

Output: The output is a pixel-by-pixel segmentation with a defined number of classes (5), using softmax.

The biggest improvement in our model's performance came from assigning a class weight of zero to the background class during training. This decision was based on the fact that the evaluation metric ignores the background class, making it unnecessary for the model to focus on it. By redistributing the emphasis to the meaningful terrain classes, the model was able to better learn and segment relevant features, significantly boosting the mean IoU from 0.51 to 0.70.

5 Results

After trying different implementations, we can summarize our work in a table like the following:

Table 1: Model Comparison

Model	Mean IoU
First U-Net Model	0.41
Custom U-Net Architecture (skip connections)	0.43
Atrous Spatial Pyramid Pooling (ASPP)	0.51
ASPP label 0 ignored	0.70

To check how our models were doing, we plotted the predictions the model made and the original dataset:

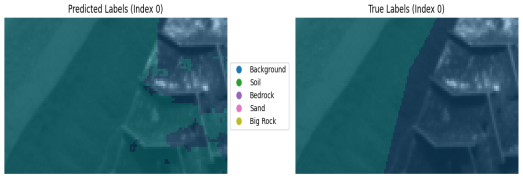

Figure 3: Model Architecture.

We started from a simple U-Net model and with intense work and research managed to increase in both ways our knowledge regarding image segmentation and our models performance finishing with . This homework was

6 Conclusions

Our final model achieved a mean IoU of 0.70, primarily due to assigning a class weight of zero to the background, enabling focus on meaningful terrain features. The ASPP block improved multi-scale feature capture but added computational overhead, slowing experimentation. While data augmentation enhanced generalization, it could have been more diverse if (f.e.) Cutmix worked properly, sadly. The model's performance relies on the assumption that the training data accurately reflects the target domain.

7 Work

Rayan worked on the implementation of the architectures, loss functions and research. Emanuele Paesano worked on the ASPP architecture. Manuel worked on the report, data augmentation and architectures. Miguel worked on u-net implementations and on the report.

References

[1] S. AI. Atrous spatial pyramid pooling (aspp). `https://serp.ai/blogs/aspp/`, 2023. Accessed: 2024-12-14.

[2] W. Chen, J. Liang, H. Xie, and B. Yu. Attentive cutmix: An enhanced data augmentation approach for deep learning based image classification. *arXiv preprint arXiv:2003.13048*, 2020.

[3] N. Plath, M. Toussaint, and S. Nakajima. Multi-class image segmentation using conditional random fields and global classification. In *Proceedings of the 26th annual international conference on machine learning*, pages 817–824, 2009.

[4] C. Prathima and N. B. Muppalaneni. A novel framework for handling duplicate images using hashing techniques. In *2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pages 984–991. IEEE, 2023.

[5] Scikit-learn. Support vector machines, 2024. Accessed: 2024-12-14.

[6] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[7] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[8] D. Yun, S. Gupta, K. Sohn, S. Yun, and H. Lee. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6023–6032, 2019.