AN2DL - Second Homework Report
Synapses

Federico Arcelaschi, Giacomo Delera, Paolo Riva, Andrea Varesi

Kaggle federicoarcelaschi, Kaggle gdelera, Kaggle pr41482, Kaggle andreavaresi2

233301, 259904, 233240, 252584

December 14, 2024

1 Introduction

Terrain type detection and **image segmentation** are two crucial tasks involved in modern Martian missions to ensure rovers correctly follow an appropriate path and interact with the environment according to missions and experiments.
This project focuses on **image segmentation of martian soil pictures**, aiming to develop a deep learning model for segmenting images of Mars terrain into five distinct classes: **background, soil, bedrock, sand, and big rocks**.
The dataset available is composed of **gray-scale images**, paired with **masks with pixel-wise class-segmentation**. Due to class imbalance and camera perspective, several operations have to be performed on it from the get-go, in order to obtain a good level of accuracy.
The CNN architecture of choice is the **U-Net architecture**, which has imposed itself as one of the well-performing state-of-the-art neural networks regarding image segmentation problems. The robust ability of this architecture to **contract the image**, **extract context**, and **expand the image** to localize the image has proven effective in various segmentation tasks.
Due to the specific context our U-Net implementation is trained on (Martian soil segmentation), an appropriate choice of the **loss function** has to be taken into consideration in order to learn both **global and local features** of the given terrain images and associated pixel-wise classes. Our score is evaluated using the **Mean Intersection Over Union** metric.

2 Problem Analysis

- **Dataset**: The dataset used for training the model exhibited several challenges. First, it contained **110 outliers** (as shown in Figure 1) among the **2615** images it is composed by (whose shape is 64×128), which could distort the learning process and make the model struggle with generalization. Additionally, the dataset was **unbalanced**, with a disproportionate under representation of the 'big rock' class. This imbalance can cause the network to favor the overrepresented classes, leading to poor performance in predicting the underrepresented one. Moreover, the dataset used for training was relatively small in comparison to the test set, composed by **10022** images. This limited amount of training data may have further hindered the model's ability to learn generalizable features.

- **Preprocessing**: To address the challenges in the dataset, outliers were removed and data augmentation techniques were applied. The entire dataset was augmented using geometric transformation techniques, such as **horizontal and vertical flips and rotations**. A very important point to underline is that applying "aggressive" augmentation techniques produced worse results in the final score, due to the fact that particularly in image segmentation tasks, the type of augmentation applied can highly influence the model ability in identifying the right classes. The fact that the big rocks class was highly underrepresented was handled later in the (dice) loss function.

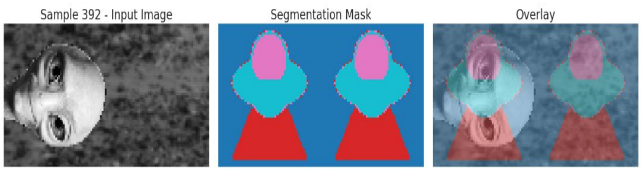

Figure 1: Outliers in the dataset

- **Main challenges**
  1. **Architecture selection**: after starting with a basic UNet, we tried to increase the complexity of the model both in the basic UNet and also using architectures such as **UNet++, UNet3+ or WNet**.
  2. **Parameters selection**: during the challenge we tried to identify the best hyper-parameters for our model, such as **learning rate, epochs, batch size** and so on. In particular we fund that we needed a high number of epochs to let our model train. In fact starting from 100 epochs, the model in some cases reached that limit, and so we decided to increase that number to let the model learn as much as possible.
  3. **Choice and tuning of the loss function**: after starting with Sparse Categorical Crossentropy, we implemented also **Dice, Focal and Boundary losses**. In particular with the dice loss, we used weights associated to each of the 5 classes, giving high relevance to the big rocks class, and this produced incredibly high improvements in our score (+10%).
  4. **Ensemble**: we tried also model ensemble, and we found that this addition produced improvements in the score.

3 Method

In all our tests we used **Early Stopping and ReduceOnPlateau** callbacks to improve the training phase and, in addition, we identified in **AdamW** the optimizer that produced the best results.
As first thing, to have a baseline we tried to use Sparse Categorical Crossentropy as loss and a basic UNet, composed by 3 levels in downsampling and upsampling and the dataset without augmentations but only cleaned. Each level was characterized by **Conv2D, Batch Normalization and Activation** layers, repeated twice. As we expected the result was bad, with only 17% accuracy score.
Then we applied data augmentation to our dataset with the techniques described before. All the results in the following are obtained using the augmented version of the dataset.
Training our simple model with the augmented dataset, we achieved a rather better result than before, hitting on average a 50% accuracy score, depending on the small changes that we applied to the basic UNet. During our tests we applied different combinations of the dice, focal and boundary losses, reaching the best results giving more weight to dice and focal losses.
After many tests, adding for example **squeeze and excitation** blocks, **shortcut connections** and **dropout layers**, we could not reach more improvements. We also tried to implement different architectures, but also with UNet++ and WNet we remained more or less stuck at the same point, hitting a maximum of 56% accuracy.
At this point we decided to implement a different version of the dice loss, to assign different weights to classes during training. This produced an important improvement in the score, achieving 66% accuracy executing the same code that reached 56% accuracy, whose architecture was a WNet.
In the final phase of the competition we focused our work to try to get better results starting from this model. For example we tried to add complexity in the bottleneck, leveraging **cross branch attention and global context modules**, and we tried to add **test time augmentation**.

4 Experiments

Several experiments were conducted on the model architecture and hyperparameters.

Exploring Different Architectures: we tried to change our architecture, training different models. The principal four models that we used are: **UNet, UNet++, UNet3+ and WNet**. We achieved the best result with WNet.
Hyperparameter tuning: after a testing phase we decided the best hyperparameters were using 200 epochs, initial learning rate of 0.0001 and batch size of 64.

5 Results

Results from testing different models can be seen in Table 1). The reported results were obtained during the development phase, with the same optimal configurations that we found and that we exposed in the previous sections.

Model	**Test conditions**	**Accuracy result**
U-Net	Base network	0.17
U-Net	Dataset augmentation	0.55
U-Net++	Dataset augmentation	0.52
U-Net3+	Dataset augmentation	0.51
W-Net	Dataset augmentation	0.56
W-Net	Dataset augmentation + class balancing	0.66

Table 1: Performance w.r.t. models

Model	**Loss function**	**Accuracy**
W-Net	Dice Loss	0.56
W-Net	Dice Loss + Focal Loss + Boundary Loss	0.55
W-Net	Dice Loss + class weighting	0.66

Table 2: Performance w.r.t. loss functions

6 Discussion

In the first phase of the development of our work we focused on the architecture, trying to increase the complexity over and over, but we understood that this way of work limited the possibility of a high score. So we understood that we had to change objective and so we focused on the loss function and the dataset. The dataset was one important point to focus on, since all the training is based on that and its augmentations. The final boost was given by the introduction of the weights in the loss functions.
We think that the model could be improved more adding the right combination of complexity in the modules of the architecture and the right test time augmentation, that was not done totally.

7 Conclusions

Among all the experiments conducted, the W-Net model, enhanced with a custom loss function and calibrated class weights, achieved the highest score of 66% on the leaderboard. It is important to highlight that while more complex models often yield better performance, proper class balancing remains essential for designing an efficient neural network. To further enhance the performance of the models, several areas of improvement can be explored. First, refining test-time augmentation techniques could significantly improve robustness. The DeepLabv3 model, with pretrained backbones like EfficientNet and optimizing the Atrous Spatial Pyramid Pooling module could yield better feature extraction and segmentation accuracy. Additionally, delving deeper into the selection and combination of various optimizers may reveal more effective optimization pathways.

The project was a collective effort: the entire team contributed to training, testing various approaches outlined in this report. Giacomo and Paolo focused on implementing and training W-Net models, with Paolo emphasizing the optimization of the loss function. Federico and Andrea worked on more advanced U-Net architectures, with Federico focusing on data augmentation and Andrea on modules to add complexity to the architectures.

References

[1] ”W-Net: A Deep Model for Fully Unsupervised Image Segmentation” Xide Xia, Brian Kulis (arXiv:1711.08506)

[2] ”Reproducing a Deep Learning Paper” Asror Walia, Erwin Russel (https://aswali.github.io/WNet/)

[3] ”Albumentations documentation” (https://albumentations.ai/docs/)

[4] ”UNet++: A Nested U-Net Architecture for Medical Image Segmentation” Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, Jianming Liang (arXiv:1807.10165)

[5] ”UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation” Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, Jian Wu (arXiv:2004.08790)

[6] ”Loss Functions in the Era of Semantic Segmentation: A Survey and Outlook” (arXiv:2312.05391v1)

[7] ”Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations Carole H Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, M. Jorge Cardoso” (arXiv:1707.03237)