AN2DL - Second Homework Report
DL Bois

Alessandro Lorenzini, Manuela Benini, Daniel Giuseppe Boi

Kaggle alelore6, Kaggle nunu02, Kaggle danielboi

273343, 273307, 278900

December 14, 2024

1 Introduction

In this project of semantic segmentation, we were given a dataset containing images of five Mars terrains, where pixels were categorized by classes. The aim was to correctly assign a class label to each pixel with the **highest possible MIoU**[1].

2 Problem Analysis

Dataset characteristics

The first thing we did was analyze the given dataset to check the presence of outliers and the distribution of the classes.
At first, we scanned the dataset and we found alien images, which were removed.

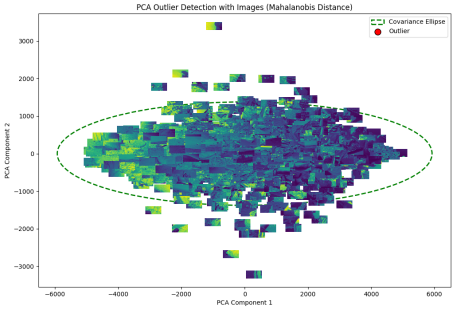

Figure 1: Visualization of the outliers based on the Mahalanobis' distance

Next, we used, as shown in figure 1, the Mahalanobis distance to find non-trivial outliers to get the final and cleaned dataset. Then, we analyzed the distribution of the classes image-wise and pixel-wise (2), finding out that it was very unbalanced.

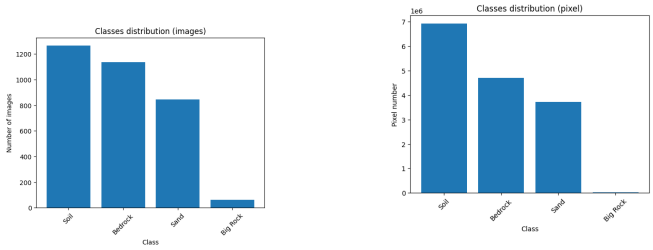

Figure 2: Image-wise distribution (left) and pixel-wise distribution (right)

Main challenges

Since the beginning, the main problem we have encountered was the fact that the class *Big Rock* was not enough represented.
Moreover, we had difficulty to raise the MIoU above a certain point. Indeed, when we reached a MIoU of 50% on validation set, all the improvements we brought seemed to be ineffective.

Initial assumptions

At the beginning, we tried to face the classes unbalance by calculating the class weights, in order to

[1] Mean Intersection over Union

redefine the importance of each class. We, also, implemented a weighted loss, based on the categorical cross-entropy, but this measures were not sufficient to correctly balance the dataset.
Afterwards, we built our first model as follows:

- **Downsampling**
  Constituted by four encoders: two simple encoders with basic convolution and two more complex encoders.

- **Bottleneck**
  With a dropout layer between two convolutions.

- **Upsampling**
  Constituted by four decoders: two simple decoders (where upsampling was performed by the nearest neighbor technique) and two more complex encoders (where upsampling was performed by the transpose convolution).

With this model, we obtained a MIoU of **40%**. Hence, we proceeded to improve this result.

3 Our procedure

We tried to raise the MIoU of *Big Rock*, which was close to zero, by implementing **Data Augmentation** and **Oversampling** techniques. We started softly with the first, applying zoom and horizontal flip both to images and masks. Then, the oversampling was done by selecting images with Big Rock and putting as background every pixel not belonging to this class to reduce overfitting. Then, these images have been duplicated a number of times and augmented. This procedure actually increased the MIoU of Big Rock but didn't bring the hoped improvements on test set.
We continued by enriching our model with inception and residual encoder/decoder and parallel, dual and residual bottleneck. Later on, we tried the UNet++, UNet3+ and **WNet** models. Eventually, we opted for the last one, which consists in two concatenated UNets: the first one extracts global features and the second one local details. In our model, we have also tried parallel dilated convolutions, pyramid pooling, attention modules, gating and squeeze-and-excitation mechanisms but with no significant result.
Then, we have implemented dice and focal loss and we wrapped them together with the weighted one to create a combined loss, in order to help the model focusing on the less represented class and difficult cases. In particular, we have realized a multi-scale supervision model, to be able to apply different loss combinations at different levels of our model, as shown in table 1.

Table 1: Loss weights at various levels

Loss	Focal	Dice	Weight
Low	0.7	0	0.3
Inter	0	0.2	0.8
High	0.7	0.3	0

With this combined loss, we noticed an improvement in the recognition of Big Rock and, additionally, we modified augmentation by making it specific for each class. After all this work, we obtained a MIoU of almost 45% on validation set, and even though the trend was very oscillating, we reached a peak of 56% on the test set.
To help the convergence and remove the oscillations, we used normalization techniques:

- *Instance Normalization*
  It normalizes each individual image (or sample) in a batch by calculating the mean and standard deviation on each channel separately, useful for microscopic batches.

- *Layer Normalization*
  It normalizes all features of each sample, considering all dimensions except the batch one, useful when we need to consider all channels equally important.

- *Group Normalization*
  It divides the channels into groups and normalizes each group separately, calculating the mean and standard deviation for each group. It is a compromise between Batch Normalization and Layer Normalization, suitable for medium and small batch sizes.

Next, however, we encountered a problem: we struggled with increasing the MIoU on validation set. Indeed, by applying augmentation, we managed to raise MIoU on training set, but this trend was not followed by validation MIoU (fixed at a value of circa 50%).

We understood that there was overfitting, so we introduced dropout layers and a learning rate scheduler. This reduced overfitting, but didn't help with the MIoU not raising. We, thus, decided to simplify our model, because we realized that it may have been too complex for the dataset provided.

4 Final work

We kept the WNet structure, but drastically simplified it, see table 2.

Table 2: Structure of the simplified WNet

	Encoders	Bottleneck	Decoders
1° U	3 simple	inception	3 simple
2° U	3 simple	1x1 conv	1 residual, 2 simple

Furthermore, we managed to implement the boundary Loss, in order to improve the recognition of the edges between terrains. We also noticed that the performance of the model was better using low weighted loss. At the end, the following values (table 3) of losses were the ones that gave us the greater results.

Table 3: Loss weights

Loss	Focal	Dice	Boundary
Low	0.4	0	0.6
Inter	0.3	0.5	0.2
High	0.2	0.5	0.3

In particular, we weighted the low loss at 0.1, the inter loss at 0.5 and the high loss at 0.4[2].
With this model, we obtained a MIoU of **62.77%** on test set (see figures 3 and 4).

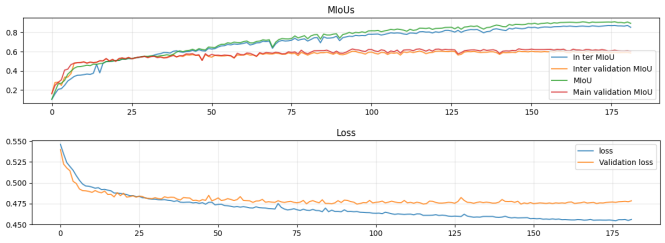

Figure 3: Graph of the MIoU and the loss during training and validation.

Table 4

Terrain	Iou
Soil	0.7138
Bedrock	0.5477
Sand	0.6750
Big Rock	0.0709
Mean	0.5019

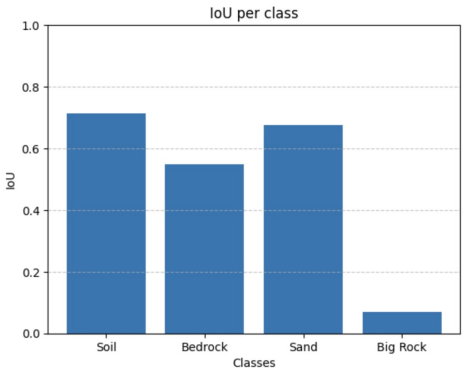

Figure 4: Final IoU per class, calculated on the validation set.

Eventually, we added also Test Time Augmentation, that slightly improved the performance.

5 Conclusions

The contributions of each team member to the final work are:

- **Base model:** Daniel Giuseppe Boi
- **Data Augmentation and Oversampling:** Alessandro Lorenzini
- **Testing of various architectures:** Daniel Giuseppe Boi
- **Loss:** Manuela Benini, Alessandro Lorenzini
- **Model:** we worked all together to find the optimal configuration

We found this project really challenging, but we all cooperate to push our model to the maximum, applying the techniques studied in the course.

[2]Respectively, they are at the first bottleneck, the output of the first U and at the end of the model.

References

[1] G. Boracchi. Lecture notes. PowerPoint presentation, 2024. Available here.

[2] E.Lomurno. Lecture 5 here.

[3] Authors: Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto,Xianhua Han,Yen-Wei Chen,Jian Wu; Title: *Unet 3+: a full-scale connected Unet for image medical image segmentation.*