AN2DL - Second Homework Report
GradientGang

Luca Bordin, Mattia Menegale, Cosimo Giovanni Negri,
lucabord, mattymene, cosimogiovanninegri,
272482, 273813, 278015,

December 14, 2024

1 Introduction

This project focuses on *semantic image segmentation* of real grayscale images of Martian terrain, using *deep learning* techniques and *Fully Convolutional Networks (FCN)*. Our approach involved:

1. Analysing and cleaning the dataset
2. Developing a **U-Net** from scratch
3. Gradually integrating more advanced techniques to increase the performance of our model

2 Problem Analysis

The provided training set consists of **2615 grayscale** images of **64x128** pixels representing Martian terrain. Each pixel is classified into one of **5 classes** representing distinct terrain types (as shown in Figure 1).

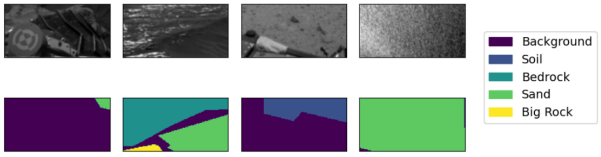

Figure 1: 4 training set images with respective ground truth label and legend with 5 classes

In order to develop a robust semantic segmentation model, we needed to address some challenges related to the dataset:

1. We removed **160 contaminated images**. In order to do that, we identified the first one and then filtered out all the images with the same mask.
2. We observed **class imbalance** in the dataset (as shown in Figure 2). In particular, the *big rock* class was heavily under-represented with a pixel distribution of just 0.13%.

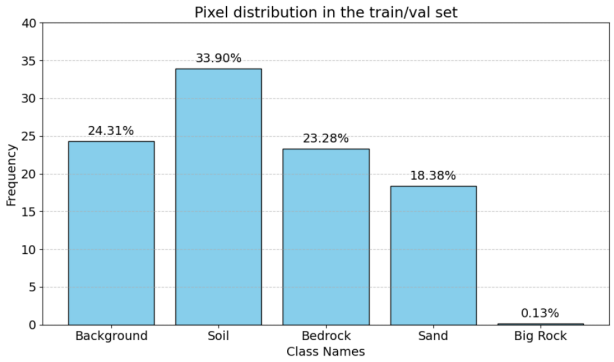

Figure 2: Distribution of training set image pixels across the 5 classes.

Data contamination would lead to a **poor performance** on test data, whereas class imbalancement is primarily related to **overfitting**.

3 Method

3.1 Data preparation

To address the heavy imbalance of the *big rock* class, we **oversampled** the images containing enough pixels of this class, creating 24 augmented versions of each one.
Then, we split the training set into a train set (80%) and a validation set (20%), keeping the augmented *big rock* images in the train set and the original ones in the validation set.
In the end, we implemented an **augmentation pipeline** using the albumentation library with the following transformations: *Affine*, *HorizontalFlip*, and *VerticalFlip*. We expanded the train set by generating 3 augmented versions for each image, in order to train the model on more data.

3.2 Model

First, we implemented a **U-Net** model [2]. The architecture consists of a symmetric encoder-decoder structure connected by skip connections. These skip connections allow the network to combine spatial details from previous layers with high-level semantic features from deeper layers.
The **encoder** (down-sampling path) extracts features using convolutional layers, batch normalization, and ReLU activation. Max-pooling is applied to progressively reduce spatial dimensions, enabling the model to capture context at multiple scales.
The **bottleneck**, positioned at the center of the network, represents the most compressed and abstract feature representation.
The **decoder** (up-sampling path) reconstructs the segmentation mask by progressively increasing spatial resolution. It combines up-sampled features with the corresponding encoder features using concatenation, followed by convolutional layers for refinement.
The final output layer produces a segmentation mask with probabilities for each class using a **softmax** activation function.

3.3 Training

We set up a few callbacks to be used during training:

- An **EarlyStopping** callback to prevent overfitting.
- A **ReduceLROnPlateau** callback to reduce the learning rate when the validation loss has stopped improving.
- A custom **visualization callback** for the sole purpose of seeing model predictions at regular intervals during training.

To evaluate our model, we used metrics such as accuracy and mean IoU (that ignores the background class).

3.4 Ignore Background

Since the mean IoU used to evaluate the models does not take into consideration the background class, we created a new model whom *Categorical Crossentropy* loss function ignored the pixels of the background class. This significantly improved the performance of our model, achieving a mean IoU of **69.5%** on the test set. The trade-offs of this solution will be discussed in Section 6.

4 Experiments

During our experiments, we tested various strategies and techniques. However, some approaches did not yield the expected results:

- **Loss function**: we implemented a custom loss function which was a linear combination of *Dice Loss* [7] and *Focal Loss* [6], and tried to assign different weights (ranging from 0 to 1) to each function, but we didn't achieve any significant improvements.
- **Augmentations**: we tried multiple combinations of pipelines including the following color [5] and geometric [4] augmentations: *RandomBrightnessContrast*, *CLAHE*, *RandomGamma*, *ToGray*, *ElasticTransform*, *GaussNoise*, *GaussianBlur*, *GridDistortion*. However they negatively impacted the performance.
- **SOTA U-Net**: we attempted to implement and tune the model proposed in [3], but it had a lower performance.
- **Manual labeling**: we noticed that many images in the training set had an imprecise mask, so we tried to fix them. We used CVAT [1] to

refine 1/5 of them manually, but the performance of the model was not improving, so we decided to stop re-labeling them.

- **Pseudo-Labeling**: as we noticed that many ground truth masks, associated to training set images, were imprecise, we implemented a *pseudo-labeling* script, inspired by [8].

5 Results

Table 1 highlights the performance results of two models: the *baseline U-Net* and the *no-background U-Net* that ignores the background class during training. We can observe that:

- **Accuracy**: The *baseline U-Net* achieves a higher validation accuracy compared to the *no-background U-Net*. This difference is expected, as this metric takes into consideration all pixels, including the background class.
- **Mean IoU**: The *no-background U-Net* achieves a higher validation and test mean IoU compared to the *baseline U-Net*. Since this metric ignores the background class, it better reflects the ability of the model to distinguish between meaningful terrain types.

6 Discussion

The results indicate a trade-off between overall pixel accuracy and meaningful segmentation performance. Ignoring the background class during training improves the segmentation of foreground classes, achieving a higher mean IoU. However, this method reduces the precision of the foreground classes and greatly reduces the accuracy of the background class, as the model trained in that way tends to disregard it entirely.
In conclusion, the choice of the best model to be used depends on the application. For tasks where background accuracy cannot be ignored, the **baseline U-Net** might be more appropriate. For scenarios emphasizing detailed foreground segmentation, the **no-background U-Net** proves to be advantageous.

7 Conclusions

This work demonstrates the effectiveness of a U-Net architecture specifically designed for semantic segmentation of Martian terrain. By excluding the background class during training, we achieved a significant improvement in mean IoU, although this reduced the precision for the background class.
To further enhance the model, future improvements could include:

- Integrating **Transformers** for better global context understanding.
- Adding a **Pyramid Pooling Module (PPM)** to capture multi-scale spatial features.

8 Authors' Contributions

- **Luca Bordin:** Worked on data augmentation techniques and implemented the Dice and Focal Loss.
- **Mattia Menegale:** Worked on oversampling techniques and implemented the pseudo-labeling strategy.
- **Cosimo Giovanni Negri:** Conducted manual labeling and contributed to the implementation of the U-Net architecture.

All authors contributed to the overall analysis, development, and optimization of the model.

Table 1: Final models. Best results are highlighted in **bold**.

Model	Validation Accuracy (%)	Validation Mean IoU (%)	Test Mean IoU (%)
baseline U-Net	**80.45**	66.09	50.82
no-background U-Net	68.86	**87.00**	**69.58**

Note: The *Accuracy* is computed on all pixels, while the *Mean IoU* is computed ignoring the background class.

References

[1] CVAT. https://cocodataset.org/#format-data. An open-source tool for annotating digital images and videos for computer vision models.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. https://arxiv.org/abs/1505.04597.

[3] L. L. Junbo Li, Keyan Chen and Z. Shi. Marsseg: Mars surface semantic segmentation with multi-level extractor and connector. https://arxiv.org/html/2404.04155v1.

[4] Keras. Albumentations documentation - geometric transforms. https://albumentations.ai/docs/api_reference/augmentations/geometric/transforms/.

[5] Keras. Albumentations documentation - transforms. https://albumentations.ai/docs/api_reference/augmentations/transforms/.

[6] Keras. Categorical focal crossentropy. https://keras.io/api/losses/probabilistic_losses/#categoricalfocalcrossentropy-class.

[7] Keras. Dice loss. https://keras.io/api/losses/regression_losses/#dice-class.

[8] Y. Zou, Z. Zhang, H. Zhang, C.-L. Li, X. Bian, J.-B. Huang, and T. Pfister. Pseudoseg: Designing pseudo labels for semantic segmentation. https://arxiv.org/abs/2010.09713.