AN2DL - Second Homework Report
TensorTribe

Matteo Figini, Caterina Motti, Andrea Grassi, Marco Gervatini

teofigio, caterinamotti, andreagrassi10, marcogervatini

248094, 252240, 252516, 251749

1 Introduction

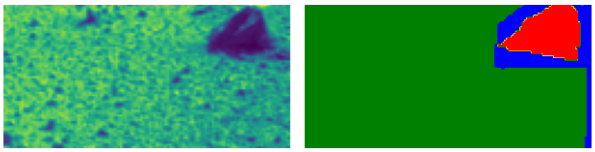
In the second homework of the Artificial Neural Networks and Deep Learning course, we faced an image **semantic segmentation** problem. The goal is, given a dataset with images and segmentation masks represented as multidimensional arrays with the same height and width, to classify each pixel of an image into the correct class. The dataset consist of images of the Mars surface and each pixel should be classified into 5 different classes: Background (class 0), Soil (class 1), Bedrock (class 2), Sand (class 3) and Big Rock (class 4). To evaluate the model's quality, we considered the Mean Intersection over Union metric, excluding the evaluation of class 0. Indeed, inside the images the background class was used to classify all the pixels that did not belong to any of the other classes, thus it is not valuable for the score.

2 Problem Analysis

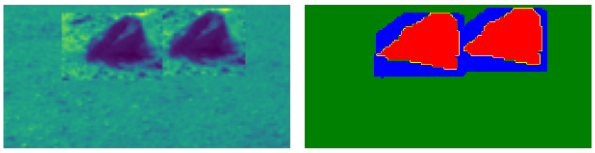
2.1 Dataset Characteristics

The dataset consists of 2615 segmented grayscale images from Mars terrain. Each image is paired with a mask, representing the class of each pixel; both the images and the masks have a size of 64 x 128.

While performing an exploratory data analysis, we first noticed that there ware a bunch of outlier images representing aliens, so we removed them and their respective masks. We also noticed that the last class, representing big rocks, has a much lower frequency than the other classes. In order to re-balance the dataset we used both custom weighted loss functions, detailed in section 3, and a custom oversampling technique following the line of this paper [7]. Regarding the second approach, we selected a specific image from the training set and copy-pasted a big rock (including the contour background) in all the training set, we then concatenate the results. In figure 1a we can see the original image and in figure 1b an oversampled image with the corresponding mask.



(a) Sample taken from the training set



(b) Oversampled image

Figure 1: Result of custom oversampling.

The dataset was splitted between training and validation set with a 80/20 ratio.

We implemented a custom function to handle geometric random augmentations applied to both im-

age and mask; the function was applied to the entire training set and results were concatenated with the original set.

We also experimented with colour augmentations (AutoContrast, RandomSaturation, RandomBrightness, Solarization), but their effect on the overall performance was negligible.

3 Methods

As previously stated in Chapter 1, the main metric we used to evaluate the goodness of our predictions is the **Mean Intersection over Union** (Mean IoU). This metric measures the overlap between the predicted segmentation and the ground truth segmentation; for a single class i, the IoU is calculated as:

$$IoU_i = \frac{Intersection}{Union} = \frac{|A_i \cap B_i|}{|A_i \cup B_i|}$$

where the $Intersection$ represents the true positives, while the $Union$ represents the sum of the true positives, the false positives and the false negatives.

The Mean IoU is defined as the average over all the classes, except the Background class (0) which doesn't count for the final results:

$$mIoU = \frac{1}{|C|} \sum_{c \in C} IoU_i$$

Regarding the loss function, we have implemented a custom weighted function with an equal contribution of the **Generalized Dice Loss** and the **Focal Loss**.

The Generalized Dice Loss is given by:

$$\text{GDL} = 1 - \frac{2 \sum_i (y_{\text{true}}(i) \cdot y_{\text{pred}}(i))}{\sum_i (y_{\text{true}}(i) + y_{\text{pred}}(i))}$$

where $y_{\text{true}}(i)$ and $y_{\text{pred}}(i)$ are the values for the i-th pixel in the true and predicted masks, respectively. It emphasizes the intersection of the predicted and true masks, which is particularly important for segmenting small, irregular objects.

The Focal Loss is defined as:

$$\text{FL} = -\sum_i y_{\text{true}}(i) \cdot (1 - y_{\text{pred}}(i))^{\gamma} \cdot \log (y_{\text{pred}}(i))$$

where γ is the focusing parameter that modulates the loss. It is designed to tackle class imbalance by focusing on hard-to-classify examples. It down-weights easy examples (those with high confidence) and directs the model's attention to more challenging cases where the prediction is uncertain.

The combined loss function is a weighted sum of the two losses:

$$\text{Total Loss} = \alpha \cdot \text{GDL} + \beta \cdot \text{FL}$$

where $\alpha = \beta = 0.5$.

We used AdamW optimizer [5] since it decouples the weight decay from the gradient updates, which prevents overfitting by penalizing large weights. Moreover, to prevent overfitting we employed early stopping (monitoring validation mean IoU) and reduced learning rate on plateau (monitoring validation loss), with the callbacks provided by Keras.

4 Experiments

We started experimenting using a simple **U-Net** taken from the paper [6]. The architecture is made of three main parts:

- **Downsampling path**: a CNN which extract global features from the input image using convolutional 2D layers, by reducing the spatial dimensions.
- **Bottleneck**: the connection between the downsampling path and the upsampling path.
- **Upsampling path**: maps the global features extracted and tries to localize them into the specific pixels of the image, by increasing the spatial dimensions.

Skip connections between the downsampling and the upsampling path are used to preserve the spatial information, facilitate the gradient flow and combine contextual and detailed information.

The original U-Net architecture was enhanced by adding Dropout layers in the last blocks of the downsampling path and inside the bottleneck (with a dropout rate of 50%). The choice of where to place the Dropout layers was carefully considered based on the network architecture. This provided a form of regularisation while avoiding losing features that may be important for the model. It slightly

increased performance, around 3% more of Mean IoU.

We added **residual blocks**, following the line of the paper [2], as they help build a deeper network without worrying about the problem of vanishing gradients or exploding gradients: this also resulted in a better performance, around 3% increase of Mean IoU.

We incorporated **Squeeze-and-Excite (SE) blocks** in the deepest levels of the network both in downsampling and upsampling paths, improving the model by focusing on the most informative channels.

We experimented with different bottlenecks such as **Pyramid Pooling (PP)** and **Atrous Spatial Pyramid Pooling (ASPP)** but neither of them actually improved the performance: the classic U-Net worked well with a simple bottleneck, with the same basic blocks and a higher dilation rate, to help in capturing broader context.

We also experimented with other networks, such as Double U-Net, both serial and parallel, and U-Net++. In the table 1 are presented the Mean IoU score over the test set, numerically sorted.

Table 1: Networks implemented

Network	MeanIoU
U-Net	0.61892
U-Net++	0.62356
Double U-Net	0.62784
Double parallel U-Net (global + local)	0.63169
U-Net, residual, SE	0.67663
Double U-Net, residual, SE, ASPP, attention gates	0.70139
Double U-Net, residual, SE, ASPP	0.70329

In the end, following the insights from paper [4] we integrated **attention gates** in the Double U-Net, but the model performance remained almost the same.

5 Results and Discussion

In the end, the best model is the Double U-Net, partially taken from [3]. Our exact model is represented in figure 2. It incorporates residual paths, SE blocks at the deepest levels of the network and the ASPP bottleneck taken from the paper [1].

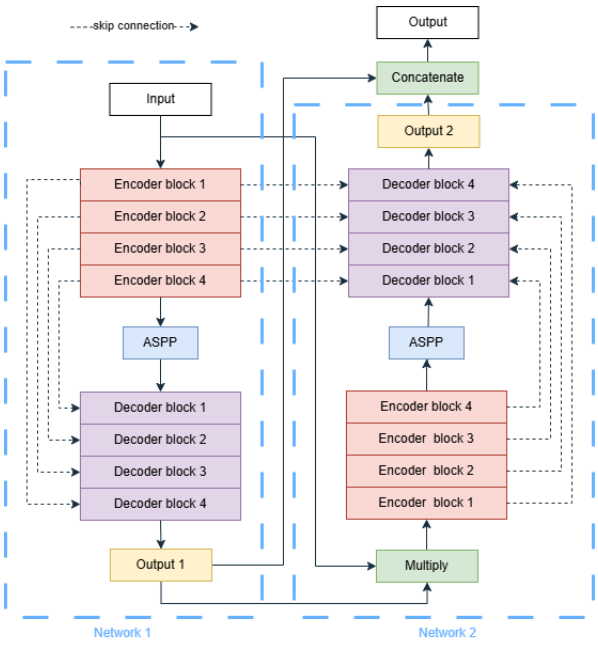


Figure 2: Double U-Net: final model.

6 Conclusions

In this project, we developed a deep-learning model to classify regions of Mars terrain images.
While our model achieved a good accuracy, there is still room for improvement. For example, a post-processing mechanism could be implemented to refine the predictions on the test set based solely on the background regions.

Team contributions:

- **Figini Matteo**: Exploratory analysis of the given dataset, creation of the prototype of U-Net architecture with dropout layers.
- **Gervatini Marco**: Data augmentation, augmentation on big rock class, classical U-net customization, U-net++, attention gates.
- **Grassi Andrea**: Analysis of dataset, rebalancing of big rock class, loss function, Double U-Net, squeeze and excite blocks, attention gates, ASPP, experiments with different nets.
- **Motti Caterina**: dataset analysis, data augmentation, data rebalancing by oversampling big rock class, custom weighted loss function, classical U-Net, Double U-Net (parallel & serial), residual paths, squeeze and excite blocks, ASPP, PP, auxiliary outputs at decoders, attention gates, final best model.

References

[1] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation, 2017.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[3] D. Jha, M. A. Riegler, D. Johansen, P. Halvorsen, and H. D. Johansen. Doubleu-net: A deep convolutional neural network for medical image segmentation, 2020.

[4] A. M. Khan, A. Ashrafee, F. S. Khan, M. B. Hasan, and M. H. Kabir. Attresdu-net: Medical image segmentation using attention-based residual double u-net, 2023.

[5] I. Loshchilov and F. Hutter. Fixing weight decay regularization in adam. 2017.

[6] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. 2015.

[7] L. Wu, J. Zhuang, W. Chen, Y. Tang, C. Hou, C. Li, Z. Zhong, and S. Luo. Data augmentation based on multiple oversampling fusion for medical image segmentation. 2022.