POLITECNICO
MILANO 1863

AN2DL - Second Homework Report
acque del friuli

Filippo Riva, Christian Rossi, Kirolos Sharoubim, Antonio Sulfaro

fil2001, christianrossi2, carlosharu, antoniosulfaro

260296, 245631, 250813, 259954

December 14, 2024

1 Introduction

The primary goal of this project is to address a *five-class segmentation problem* using 64×128 grayscale images of Mars terrain, leveraging a labeled dataset containing 2,615 samples, and a test dataset of 10,022 unlabeled samples.

To achieve the highest possible **mean Intersection over Union**, we systematically experimented various techniques discussed during the lectures, comparing the performance of each approach.

2 Problem Analysis

The main issues we faced during this homework where:

1. The presence of **outliers**, which were maliciously inserted into the dataset.
2. A considerable **class imbalance**, which might have led to biased models.
3. Achieving a meaningful **generalization**, since many models struggled to avoid overfitting.

3 Method

During development, we followed a similar organization to the one we adopted for the first homework, allowing us to maintain an organized set of properly commented notebooks. In particular we split the work between two set of Jupyter notebooks, each dedicated to a specific task:

- *Dataset Analysis*: used to explore the dataset, identify and remove outliers, and test various augmentation techniques.
- *Model Development and Training*: a structured environment for building neural networks and performing training experiments.

As for the first homework, in order to allow parallel development, each team member created a personalized copy of a notebook to implement their unique approach. This strategy even enabled us to easily track intermediate results, helping identify the techniques that were best suited for the problem.
Furthermore, we adopted an **incremental approach**, starting with a simple U-Net and gradually moving towards more advanced techniques. This step-by-step refinement allowed us to build on our insights and systematically improve performance.

3.1 Problem Resolution

The challenges outlined in the second section were addressed as follows:

1. Outlier removal: since the dataset was composed by few images, we inspected it manually, detecting all the images containing an

alien, and subsequently removed them. We later saw that all the outliers had exactly the same label, so we checked if we left some images in the dataset with that label and removed them.

2. Class balancing: since we had to work with entire segmentation masks, oversampling became less meaningful. Thus, we have decided to use class weighted loss functions to avoid building biased models. This approach was indeed helpful, but its effectiveness was limited due to the huge class imbalance on the *Big Rock* class.

3. Regularization: in order to improve the generalization capabilities of our models, we made extensive use of dropout and regularization techniques, that allowed us to avoid overfitting.

4 Experiments

Below, we present only the models and configurations that provided meaningful results:

1. *Basic U-Net*: architecture with two downsampling and two upsampling blocks (comprising *Convolutional*, *Batch normalization* and *ReLU* and a basic *Bottleneck* layer). We used *Sparse Categorical Crossentropy*. This model was used as a baseline for further improvements.

2. *Deeper U-Net*: to improve the performance of the U-Net, we have decided to use a more appropriate loss function (*Dice*) made for segmentation problems. We also added a third downsampling and upsampling block. Those modifications led to a slight improvement over the baseline. Note that the loss function we used is meant for binary segmentation, so we modified it for multi-class segmentation. However, in this case, we still didn't use a class weighted loss.

3. *Autoencoded U-Net*: to explore alternative approaches and exploit the huge amount of test data we were provided we have decided to train an Autoencoder over the union of train set and test set images. Therefore, we extrapolated the encoder part of the Autoencoder architecture to be used as a pre-trained encoder of our U-Net model for segmentation. The final architecture however achieved the worst result between our experiments even after fine-tuning, so we decided to avoid this route whatsoever assuming that the features learnt by the Autoencoder were not meaningful for a segmentation problem.

4. Improved U-Net:

   - *Augmentations and Weighted Dice*: to try to improve the model's generalization capability we performed some simple augmentations to avoid creating distorted samples. In particular, we applied vertical and horizontal flips, adjusted brightness and contrast, and added some noise to simulate various space scenarios, such as varying lighting conditions and the presence of dust. Furthermore we started training with a Class Weighted Dice to improve the prediction capability of the model on the Big Rock class.
   - *Focal and learning rate scheduling*: to further improve the performance on unbalanced data we tried Categorical Focal Crossentropy to focus on hard to train pixels and improve boundaries precision. Furthermore we added a learning rate scheduler to reduce the learning rate when a plateau was detected on the mean intersection over union on validation.

   Both approaches slightly affected performance but did not lead to significant changes. Nevertheless, we chose to retain these augmentations in the subsequent experiments.

5. *Multi-headed architectures*: To improve intermediate features of the network , being inspired by the GoogLeNet architecture, we added multiple classification heads after certain decoder blocks, increasing it's convergence but making it harder to train. After that, we even tried using *Attention Gating Mechanisms* on the U-Net Skip Connections to obtain a network with trainable feature fusion. The latter ended up performing unexpectedly worse than the former, which was (unfortunately) the best architecture we trained during the challenge.

6. *Conditional Random Fields U-Net*: The last experiment with plain U-Net architectures was applying *Conditional Random Fields* on predicted labels of our best architecture. Since this approach performed even worse, we decided to avoid this post-processing solution.

7. *U-Net3+*: to implement learnable skip connections we tried a more complex architecture inspired by a SOTA paper[2]. Additionally we implemented a modified version of the U-Net3+ using the an Attention Gating Mechanism. In both cases we almost no improvements with respect to the baseline model.

With regard to the training techniques used we exploited *Adam* optimizer, early stopping and learning rate scheduling.

The following table summarizes the performance of the models on Kaggle:

Model	mean IoU
Basic U-Net	43%
Deeper U-Net	44%
Autoencoded U-Net	29%
Weighted - Focal U-Net	45% - 47%
Multi-head U-Net	**48%**
Multi-head (attention) U-Net	46%
CRF U-Net	39%
U-Net 3+	42%
U-Net 3+ (attention)	44%

5 Results

The Multi-head U-Net without the Gating Attention Mechanism ended up being the best architecture overall thanks to its simplicity and its ability to find meaningful intermediate feature maps thanks to it's multi-headed architecture. The network was probably one of the simplest we tried out being a standard depth-4 U-Net, with simple skip-connections, non trainable down-sampling and up-sampling and encoder/decoder blocks composed of 2 stacks of convolution layers, batch normalization, and ReLU activation functions. It was trained using Categorical Focal Crossentropy as loss, Early Stopping and Learning Rate Scheduling, over a modified dataset using horizontal/vertical flips and contrast augmentations.

6 Final Considerations

Contrary to our beliefs we were not able to improve beyond the simple proposed architecture. This is due to many reasons :

Scaling to more complex architectures like U-Net3+, Link-Net, ResUNet or U-Net with attention blocks ended up making it harder to train, resulting in overfitting scenarios that even through extensive usage of regularization techniques such as dropout and l2-regularization were unavoidable, resulting in $> 0.60\%$ mean IoU on training an and $\approx 0.45\%$ in validation. Even alternative loss functions like Dice, Lovasz, Weighted Dice and various weighted ensemble of these ones, didn't really differ in overall performance, even if they correctly introduced slights modifications on how the predictions behaved. Furthermore, since we observed a lot of variance on the predictions, we tried to implement an ensemble of 2 differently trained U-Net, with different loss functions, trying to overcome the high-variance problem, with unfortunately no relevant results observed.

Even if our results on these more advanced strategies where unsatisfactory, we are more than certain that with further time and resources we could refine these complex models like U-Net3+ or even try simpler alternatives like U-Net++, and obtain better performances.

7 Conclusions

Each team member contributed equally to modifications and ideas that cumulatively led to the development of our results.
Future improvements could involve:

- Trying simpler but still powerful architectures like U-Net++.
- Building ensembles of 3 or 4 complex models, since we are in a low bias, high variance scenario.

References

1. M. Matteucci, G. Boracchi. AN2DL slides. 2024.
2. Huimin Huang, et al. UNet 3+: a Full-Scale Connected Unet for Medical Image Segmentation. 2020.
3. Robin Vinod. A detailed explanation of the Attention U-Net. Medium 2020.