POLITECNICO
MILANO 1863

AN2DL - Second Homework Report
Bio.log(y) Team

Luca Lepore, Arianna Rigamonti, Michele Sala, Jacopo Libero Tettamanti

hif1beta, ariannarigamonti, michelesala, JacopoSheep

252309, 252321, 252325, 252329

December 14, 2024

1 Introduction

The second homework assignment focuses on a semantic segmentation problem. The goal is to develop a robust model able to accurately segment Mars terrain images by assigning each pixel to its respective class. To address it, a stepwise approach was used, starting with a baseline model and gradually increasing complexity to evaluate performance improvements.

2 Problem Analysis

The dataset includes 2,615 grayscale images (64x128 pixels) of Mars terrain with pixel-wise masks for five classes: Background, Soil, Bedrock, Sand, and Big Rock. It shows significant class imbalance, with Class 4 (Big Rock) being underrepresented (<1%), and a small number of training images. Accurate segmentation depends on distinguishing key features like texture, pixel intensity, and structural patterns, while focusing on foreground classes and excluding the background from evaluation.

3 Methods

3.1 Data Preprocessing

The dataset inspection revealed the presence of many Mars terrain images with aliens in various positions but sharing identical masks. To reduce bias, all duplicates, including a reference image, were removed, decreasing the dataset size from 2,615 to 2,505 images. The dataset was then split into training, validation, and internal test sets, as the official test set masks were unavailable. Finally, the pixel intensity values of the images were normalized to a range of 0 to 1.

3.2 Data Augmentation

To expand the training set and improve class 4 (Big Rock) representation, two augmentation setups were tested. The first, lighter, involved flipping, rotation, shifting, and zooming, quadrupling images with class 4 and doubling the rest. The second setup transferred cropped class 4 portions, with random rotation and scaling, to 80% of images without class 4. Albumentation was then applied, incorporating flipping, rotation, elastic transformations to all images to double their number. A final step quadrupled images containing class 4 using shifting, scaling, zooming, and rotation. In both setups, original images were preserved in the final training sets.

3.3 Models

3.3.1 Baseline CNNs

For this project, two baseline models were implemented. The **first baseline model** (**Single-Layer**

CNN) represents an extremely simple architecture consisting of a single 1x1 convolutional layer with softmax activation. The **second baseline model** (**3-Layer CNN**) features a deeper architecture with three 3x3 convolutional layers with increasing filter sizes (64, 128, 256) and ReLU activation functions, followed by a final 1x1 convolutional layer with softmax activation for mask prediction.

3.3.2 U-Net architectures

The U-Net architecture extends beyond the baseline models by using a symmetric encoder-decoder structure with skip connections to incorporate global context through downsampling while preserving local feature information[1]. Our first implementation (**U-Net Base**) consists of two downsampling blocks in the encoder path using 32 and 64 filters respectively, and a bottleneck layer with 128 filters. This is followed by a symmetric decoder path. Each block contains two 3x3 convolutional layers with batch normalization and ReLU activation. We also developed a deeper variant (**U-Net TC**) with three downsampling blocks in the encoder path (32, 64, and 128 filters), a 256-filter bottleneck layer, and transposed convolutions for upsampling in the decoder path instead of simple upsampling operations.

3.3.3 Residual and Attention-based U-Nets

Two enhanced U-Net architectures were also implemented. The first, **U-Net RA (Residual-Attention)**[2], introduces residual blocks and attention gates to facilitate training of deeper networks and enhance feature selection in skip connections. The second, **U-Net MSRA (Multi-Scale Residual-Attention)**[3], further extends this architecture by introducing multi-scale context through parallel dilated convolutions in the bottleneck layer, enabling feature capture at various scales. Both models maintain three encoding levels (64, 128 and 256 filters) and a 512-filter bottleneck.

3.3.4 Double U-Net Architecture

The **Double U-Net** model combines two parallel U-Net paths with different focuses: the first processes detailed features using residual blocks and attention gates, while the second path captures global context through average pooling and a transformer block in the bottleneck. Both paths incorporate multi-scale context through dilated convolutions. Through concatenation and a final convolution, the outputs are fused to combine local and global features[4].

3.3.5 Pyramid Pooling Network

Another architecture we explored is **DeepLabV3** which introduces Atrous Spatial Pyramid Pooling (ASPP) for multi-scale feature extraction[5]. Our implementation uses three encoding levels (from 32 to 128 filters) followed by parallel atrous convolutions with different dilation rates (6, 12, and 18). The network combines these multi-scale features through ASPP and employs a bilinear upsampling decoder, enabling effective capture of both fine spatial details and global context.

3.3.6 U-Net++

The U-Net++ architecture introduces dense skip connections and deep supervision[6]. Our implementation features five encoding levels with increasing filter sizes (from 64 to 1024 filters) and nested dense connections between encoder and decoder. The network employs deep supervision to improve gradient flow, while the dense skip connection pattern helps bridge the semantic gap between encoder and decoder features.

3.3.7 Transformer Multi-Scale U-Net++

Our final architecture, **TMS U-Net++** combines UNet++ design principles with transformer-based feature processing and adaptive fusion mechanisms. This model features three encoding levels (from 64 to 256 filters) with residual blocks and group normalization, followed by a multi-scale transformer bottleneck of 512 filters and implements nested dense connections with adaptive fusion gates at each decoder level.

3.4 Loss function

Various loss functions were evaluated, including Focal Loss, Dice Loss, and Generalized Dice Loss. **Focal Loss** is designed to handle class imbalance by focusing more on hard-to-classify examples. **Dice Loss** measures the overlap between predicted and true masks, prioritizing regions with lower overlap. **Generalized Dice Loss** extends Dice Loss by

weighting each class based on its representation to further address class imbalance[7]. The background class (label 0) was excluded from loss calculations. Each loss was tested individually and in combinations on baseline CNNs and basic U-Net models, with the best loss determined by the highest validation mean IoU.

4 Experiments

After testing the baseline CNN and U-Net models on internal and Kaggle test sets with the original, lightly augmented, and moderately augmented training sets, the original set was chosen, as augmentation showed no improvement in mean IoU. The best loss function was a weighted combination of Focal Loss (0.8) and Dice Loss (0.2), selected based on validation mean IoU. Model performance on internal and Kaggle test sets is summarized in Table 1

Table 1: Performance Comparison of Trained Models (Mean IoU)

Model	Val IoU (%)	Internal Test IoU (%)	Kaggle Test IoU (%)
Single-layer CNN	15.12	13.68	13.84
3-Layer CNN	38.48	38.64	38.65
U-Net Base	51.80	50.16	52.93
U-Net TC	59.45	58.91	59.29
U-Net RA	**65.09**	**65.80**	**65.73**
U-Net MSRA	62.00	62.19	62.35
Double U-Net	59.49	59.51	59.49
DeepLabV3	54.39	56.37	56.18
U-Net++	60.45	62.88	62.56
TMS U-Net++	59.96	61.89	60.07

5 Results

The **U-Net RA** achieved the best results with a mean IoU of 65.80% on the internal test and 65.73% on the Kaggle test, significantly outperforming the baseline CNN's 13.68% and 13.84%, respectively. Class-wise IoU for the best model is detailed below. (Table 2).

Table 2: Class-wise IoU Percentages Achieved by U-Net RA on the Internal Test Set

Class	Internal Test IoU (%)
0	Excluded
1	72.60
2	54.36
3	65.31
4	3.64

6 Discussion

Despite augmentation setups aimed at increasing training images and improving class 4 representation, the non-augmented dataset achieved the best results. This may be due to the high similarity between training and test set images, limiting augmentation effectiveness. Additionally, synthetic samples may have caused the model to overfit to augmented patterns rather than learning meaningful features for the underrepresented class, resulting in unchanged or worse performance. The combination of Focal Loss (0.8) and Dice Loss (0.2) proved the most effective, balancing class imbalance and segmentation accuracy to align with the dataset's characteristics. The best-performing model achieved **65.73%** Mean IoU on the Kaggle test set. Semantic segmentation, unlike image classification requires pixel-level predictions, making the task significantly more complex. Metrics like Mean IoU are harder to optimize, especially with extreme class 4 underrepresentation and a limited dataset size. The lack of pretrained models further constrained the model's ability to generalize. However, improvements are certainly possible, as the IoU for class 4 is only 3.64%. In these terms, the highest IoU for class 4 was achieved by the TMS U-Net++ model, reaching 12.04%.

7 Conclusions

Future work could include hyperparameter tuning of the loss function, focusing on individual parameters and the combination of Dice and Focal Losses to optimize performance. A two-phase training strategy with varying loss weightings could also be explored. Further improvements might involve optimizing the best-performing models and using GANs to generate synthetic images, potentially enhancing generalization. Testing alternative optimizers remains valuable, despite AdamW showing no notable improvements when tested on DeepLabV3. Additionally, creating an ensemble of the U-Net RA and the model with the highest class 4 IoU could balance overall performance while improving predictions for that class. The use of pretrained models as feature extractors could also be a promising direction.

Additional Materials

Each group member's contribution as well as all runned notebooks (and augmented datasets) are available at this link

References

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *arXiv preprint arXiv:1505.04597*, 2015.

[2] Xiaocong Chen, Lina Yao, and Yu Zhang. Residual attention u-net for automated multi-class segmentation of covid-19 chest ct images. *arXiv preprint arXiv:2004.05645*, 2020.

[3] Yun Jiang, Huixia Yao, Chao Wu, and Wenhuan Liu. A multi-scale residual attention network for retinal vessel segmentation. *Symmetry*, 13(1):24, 2021.

[4] Vibha Bhatnagar and Prashant P. Bansod. Double u-net: A deep convolution neural network for tongue body segmentation for diseases diagnosis. In *Proceedings of International Conference on Communication and Computational Technologies*, Algorithms for Intelligent Systems (AIS), pages 293–303. Springer, 2022.

[5] Keisuke Hamamoto, Naoya Hideshima, Huimin Lu, and Seiichi Serikawa. Single image reflection removal using deeplabv3+. In Huimin Lu and Jintong Cai, editors, *Artificial Intelligence and Robotics: 8th International Symposium, ISAIR 2023, Beijing, China, October 21–23, 2023, Revised Selected Papers*, volume 1998 of *Communications in Computer and Information Science*, pages 181–188. Springer, 2024.

[6] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In Danail Stoyanov, Zeke Taylor, Gustavo Carneiro, and Tanveer Syeda-Mahmood, editors, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (DLMIA 2018, ML-CDS 2018)*, volume 11045 of *Lecture Notes in Computer Science*, pages 3–11. Springer, 2018.

[7] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7, 2020.