POLITECNICO
MILANO 1863

Semantic Segmentation of Mars Surface Images

Agnese Negroni, Alberto Pola, Fabio Romagnoli, Niccolò Signorelli

December 18, 2024

1 Introduction

This project aims to develop a neural network capable of performing multi-class image semantic segmentation, specifically identifying which of five classes each pixel in an image belongs to. We adopt the encoder-decoder paradigm applied to Fully Convolutional Networks (FCNs [1]), a common approach in well-known segmentation architectures such as UNet [2].

The model was developed for a two-weeks student competition organized by the professors of the *Artificial Neural Networks and Deep Learning* course at Politecnico di Milano.

2 Data Analysis

The problem involves the classification of 64x128 grayscale images from Mars terrain. The dataset consists of 2615 images with the respective masks, of which 2505 are valid and 110 are invalid. The pixels in the valid images are categorized into five classes, each representing a particular type of terrain. Our goal was to assign the correct class label to each mask pixel.

The cleaned dataset was **highly unbalanced** among the different classes. Table 1 shows the full distribution. Moreover, upon inspection of the given masks, we discovered that most of the dataset was poorly labeled. We chose to ignore this issue, since it soon became apparent that manually editing the masks led to a decrease of accuracy, while keeping the original masking resulted in a test accuracy consistent with our validation accuracy.

Since, according to the rules, no pre-trained model was allowed, we trained our models from scratch using the provided cleaned dataset.

Table 1: Class Distribution Over Total Pixels

Class	0	1	2	3	4
(%)	24.31	33.90	23.28	18.38	0.13

3 Problem Analysis

Our model was evaluated according to the *mean intersection over union* metric, computed as

$$\text{mIoU} = \frac{1}{|C|} \sum_{c \in C} \frac{1(y = c) \wedge 1(\hat{y} = c)}{1(y = c) \vee 1(\hat{y} = c)}$$

given the set of classes C excluding class 0 (background), the ground truth y, and the model predictions $\hat{y}$.

Our goal was to design a model whose architecture could achieve a good balance between global context and fine details, and whose loss function could be aligned with the mean IoU metric and could also effectively address the issue of a highly unbalanced training dataset.

4 Experiments and Results

4.1 Data augmentation

Given the relatively small size of the dataset, we experimented with online augmentation. A series of augmentations were tested, including geometric and noise-based transformations. The best-performing ones were

found to be Gaussian noise and blur, as these can act as simple regularizers, while the geometric transformations, which perturbed the images too much, performed more poorly.

4.2 Loss functions

We understood that addressing the **class imbalance** was the biggest challenge, and in order to address this, the choice of loss function was paramount in determining how well the model would learn also the edge cases. We proceed with the revision of the most common loss functions in semantic image segmentation ([3]), and their applicability and success in our specific problem. We implemented all the loss functions with **class weighting**, to mitigate the class imbalance effects. All loss functions respect the following nomenclature, where N is the number of samples, C is the number of classes, $y_{i,c}$ is the true label for sample i, class c, $\hat{y}_{i,c}$ is the predicted probability for sample i.

Multi-class Binary classification

While Binary cross-entropy, as its name suggests, isn't apt for multiclass problems, if modified by adding a summation over classes it can be used. By doing this, the function has a one-vs-all representation, treating each class independently.

Table 2: Loss Function Formulas

Loss	Formula
BCE	$-\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\left[y_{i,c}\log(\hat{y}_{i,c})+(1-y_{i,c})\log(1-\hat{y}_{i,c})\right]$
CCE	$-\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}y_{i,c}\log(\hat{y}_{i,c})$
Dice	$1-\frac{2\sum_{i=1}^{N}\sum_{c=1}^{C}y_{i,c}\hat{y}_{i,c}}{\sum_{i=1}^{N}\sum_{c=1}^{C}(y_{i,c}+\hat{y}_{i,c})}$
Focal	$-\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\alpha_c(1-\hat{y}_{i,c})^{\gamma}y_{i,c}\log(\hat{y}_{i,c})$

Categorical Cross-Entropy (CCE)

Generalization of binary cross-entropy for multiclass problems with one-hot encoded targets.

Dice Loss

The Dice loss function is based on the Dice coefficient, which is a metric that quantifies the overlap between predicted and ground truth masks.

Focal Loss

Focal loss is an extension of Categorical Cross-Entropy. It down-weights the contribution of easy examples and enables the model to focus more on learning hard ones. It works well for highly imbalanced class scenarios.

Results

All loss functions were scaled by their respective inverse frequency pixel weighting. We report that BCE, due to the weighting was quite efficient in isolating the underapresented class. CCE performed the best in identifying the other classes, and Focal was successful in learning the representation of the background class. Since our goal was to maximize the IoU of non-background classes, **CCE** was chosen as our loss function, as it provided the best trade-off.

Table 3: Class-wise and Overall Metrics by Loss Function

	BCE	**CCE**	**Dice**	**Focal**
Acc (%)	74.3 ± 0.6	**76.8 ± 0.5**	68.4 ± 1.1	74.6 ± 1.0
IoU (%)	**51.2 ± 1.1**	51.1 ± 1.1	42.4 ± 0.4	49.3 ± 0.1
		Class-wise IoUs (%)		
Class 0	39.25	39.98	28.97	**40.94**
Class 1	71.07	**75.35**	64.17	74.36
Class 2	59.22	**62.34**	49.23	60.15
Class 3	65.24	**70.29**	59.40	67.68
Class 4	**17.39**	12.26	3.26	9.08

Class weights

Initially, the weights were computed as a standard inverse frequency of their pixel presence. Subsequently, the weights were further refined according to our findings and problem analysis. The first line in table 4 shows the inverse frequency weights. After taking into account the metric used in evaluating our success, the background class weight was set to 0, and the other ones were scaled accordingly, as shown in the second line. The final weights that were used in our training, shown in the third line, were reached heuristically, but still followed the distribution achieved earlier, and also took into account some oversampling of the big rock class. In our last phases of training, images containing the mentioned class were quadrupled.

Table 4: Loss Weights

Class	0	1	2	3	4
Inv. Freq.	0.82	0.59	0.86	1.09	152.22
0 excl.	0.0	0.0198	0.0275	0.0359	0.9168
Tweaked	0.0	0.025	0.035	0.04	0.88

4.3 Experiments

At the beginning of the challenge, we looked for research papers regarding image semantic segmentation models to take inspiration from. Two stood out: MarsSeg ([4]), designed specifically for Martian surface segmentation, and DeepLabV3+ ([5]), a state-of-the-art model developed by Google. Both of these models rely on an encoder-decoder architecture and typically leverage pre-trained encoders. We tried to develop them with a custom encoder, but, unfortunately, the results were disappointing. Therefore, we decided to focus on developing UNet-based models, concluding that pre-trained encoders are critical for the success of both MarsSeg and DeepLabV3+.

4.4 Architecture

To familiarize ourselves with common CNNs for semantic segmentation, we began with the baseline UNet model, renowned for its encoder-decoder structure and skip connections that effectively preserve spatial information. To enhance its performance, we experimented with different depths of the UNet, up to 6 layers, and we tested techniques to improve encoder-decoder communication, such as weighted addition of feature maps, where feature maps from the encoder and decoder are combined using learnable weights to emphasize relevant features and attention mechanisms. Despite these modifications, these enriched models did not outperform the original UNet. Additionally, we introduced residual connections between layers, which directly add the input of a block to its output facilitating gradient flow and improving training stability. We also inspected transpose convolutions, a learnable upsampling method, as a replacement for simple max-upsampling to better reconstruct spatial details. To further explore more sophisticated architectures, we implemented UNet++ ([6]), which leverages nested and dense skip connections with additional convolutions, and UNet3+ ([7]), which employs full-scale skip connections that aggregate feature maps from all encoder and decoder levels at each stage, enabling more effective multiscale feature fusion. Among all the models tested, the architecture that achieved the best performance for our semantic segmentation task was a **UNet3+** enhanced with residual connections and transpose convolutions.

4.5 Optimizer and learning rate

For the optimizer, we used Adam with weight decay, and employed Cosine Annealing as a learning rate scheduler. It was considerably more numerically stable then alternatives and would always reach the desired convergence for even the most complex models.

4.6 Post-processing refinements

To mitigate omission of the background class, an attempt was made to train a UNet on binary masks: background vs foreground. Once the model learnt to classify these, the probability mask generated was meshed with the prediction of our multiclass model, in an attempt to recover background patches. While this was giving promising results, a lot of parameter tuning of how the masks are thresholded is required to achieve better results.

5 Conclusions

Through the development of this project, we understood that it was crucial to carefully select a loss function aligned with the IoU metric to address class imbalance. Moreover, the restriction against using pre-trained models limited our ability to leverage transfer learning and benefit from the richer feature representations learned by larger models.

The final model achieved an **overall accuracy of 64.87%** on the evaluation test set, placing it in the top 30% of the class. The highest accuracy achieved in the class was 69.39%.

It is worth noting that this challenge prohibited the use of pre-trained models, requiring all participants to train their networks from scratch. Additionally, the training masks provided were of very low quality, which posed significant challenges during the development phase. Despite these limitations, the model demonstrated decent performance and provides a solid foundation for further improvements.

References

[1] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *CoRR* abs/1411.4038 (2014). arXiv: 1411.4038. URL: http://arxiv.org/abs/1411.4038.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.

[3] Shruti Jadon. "A survey of loss functions for semantic segmentation". In: *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, Oct. 2020, pp. 1–7. DOI: 10.1109/cibcb48159.2020.9277638. URL: http://dx.doi.org/10.1109/CIBCB48159.2020.9277638.

[4] Junbo Li et al. *MarsSeg: Mars Surface Semantic Segmentation with Multi-level Extractor and Connector*. 2024. arXiv: 2404.04155 [cs.CV]. URL: https://arxiv.org/abs/2404.04155.

[5] Liang-Chieh Chen et al. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: *CoRR* abs/1802.02611 (2018). arXiv: 1802.02611. URL: http://arxiv.org/abs/1802.02611.

[6] Zongwei Zhou et al. *UNet++: A Nested U-Net Architecture for Medical Image Segmentation*. 2018. arXiv: 1807.10165 [cs.CV]. URL: https://arxiv.org/abs/1807.10165.

[7] Huimin Huang et al. *UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation*. 2020. arXiv: 2004.08790 [eess.IV]. URL: https://arxiv.org/abs/2004.08790.