POLITECNICO
MILANO 1863

AN2DL - Second Homework Report
Madapenguins

Irene Caridi, Francesca Girolami

irenecaridi, francescagirol

243023, 226700

December 17, 2024

1 Introduction

This project focuses on *semantic segmentation* using **deep learning** techniques. In particular, the aim of the project is to develop a neural network capable of segment mars terrains related to a background (class 0) and four different types of soils: soil (class 1), bedrock (class 2), sand (class 3), big rock (class 4).

2 Dataset analysis

The training set includes 2,615 grayscale images and their masks, each sized at 64×128 pixels. Instead, the test set is composed by 10,022 images.

3 Preprocessing

By a visual inspection of the dataset and the corresponding masks, we saw that some masks seemed to be out of contest with respect to their related images. But, as the dataset consists of real data, we decided to keep all images.

Next, the overall label distribution across all images, showed that the total pixel amount for the first three classes and the background is at least of 4×10^6 . In contrast, the fourth class has only 26,963 pixels, significantly lower than the others, highlighting a important class imbalance.

3.1 Dataset cleaning

Images also included instruments of war and alien faces. The former were correctly classified as background, while to the latter were assigned a specific mask. Consequently, we decided to exclude only those images containing alien faces.

To remove them, we proceed with two ideas:

- A CNN composed by three convolutional blocks is trained on the most representative images (9 alien faces and 40 corrects). Then with a threshold all the images containing aliens were deleted.
- Since the alien images have the same mask, we searched for all images with that mask and delete them.

Both methods were accurate but we choose the easiest one so the second one.

3.2 Other attempts

Data augmentation is a technique to enhance the dataset's variability and robustness for models. But also it can be used to rebalance classes. We applied geometrical transformations like random flipping, then brightness adjustment and contrast variation. But since several trials and different values of augmentation parameters, the model didn't improve.

4 Models

The dataset is splitted: 80% training set and 20% validation set. Each subset is then processed to create datasets with normalized images between [0,1] and then batched. In addition, the training set is shuffled to introduce randomness during the training process.

Model parameters are: batch size of 64, learning rate of 10×10^4, patience of 30 and number of epochs 1000. Early stopping and reduce plateau callbacks are used, monitoring the mean intersection over union (meanIoU). Finally, model is compiled with Adam optimizer.

4.1 Single U-Net

The first model is based on a U-Net architecture [1]: encoder, bottleneck, decoder and skip connections linking the encoder and decoder. The fundamental block used in this model is composed by two convolutional layers (kernel = 3 and "same" padding), a batch normalization layer and a ReLU activation layer (stack = 2). Encoder - 4 blocks, increasing filters: 64, 128, 256, and 512. Each block is then followed by a max-pooling layer. Bottleneck - single block with 1024 filters. Decoder - 4 blocks, mirroring the encoder in reversed order. Each block is followed by an upsampling layer and a concatenation layer to establish skip connections. Finally, the output is passed through a convolutional layer with a kernel size of 1, "same" padding, and a softmax activation function to produce the final segmentation map. This model doesn't classify class 4. Moreover, we have errors in the classification of the other classes and most of them are False Positives. To improve the performance we decided to work both on the structure of the network and on the choice of the loss function.

4.2 Double U-Net

A model with two U-Net linked thought output layer is then built. The first U-Net is composed of 3 blocks (as the one of the previous model) for the encoder and 3 for the decoder. An additional block is included in the bottleneck and the L2 regularizer is added. The second U-Net also has 3 blocks for encoding and 3 for decoding, along with 1 block in the bottleneck. However, it uses a larger kernel compared to the first U-Net and includes only one convolutional layer instead of two for each block. Moreover, a weighted loss function has been implemented from scratch that takes into account the class imbalance. Several attempts have been made by modifying: the number of blocks in sequence, and their parameters. In any case, despite there has been an improvement in the correct predictions for the other classes, class 4 is not classified.

4.3 Best U-Net

The final model has the structure of the single U-Net described in section 4.1.

The encoder block is improved with a dropout layer with 0.2 as drop value after the max pooling layer and a residual block is added after U-Net block, with the number of filter equal to the one in the U-Net block to address the vanishing gradient problem. The bottleneck is built with a squeeze and excitation block, followed by a pyramid pooling block and a U-Net block with 512 filters. The squeeze and excitation block is composed by two parts: the squeeze step applies Global Average Pooling to the input tensor. Then two 1×1 convolutional layers are used. The first convolution reduces the number of channels by a factor determined by a ratio (16), followed by a ReLU activation. The second convolution restores the original number of channels with a sigmoid activation function. Finally, the output is multiplied element-wise with the original input.

The pyramid pooling block captures multi-scale contextual information from the input feature map by applying pooling operations at different scales. It works by performing average pooling with various pool sizes $(1 \times 1, 2 \times 2, 3 \times 3, 6 \times 6)$ on the input tensor. For each pool size it is applied an average pooling operation and then it is resized the pooled output back to the original spatial dimensions of the input tensor. The pooled feature maps from all different pool sizes are then concatenated along the channel axis to build a multi-scale feature representation.

Inside the decoder structure the attention mechanism is implemented [2] and the dropout layer is used with 0.2 as drop value. So each block is composed by upsampling layer with bilinear interpolation, a dropout layer, and attention block linking the dropout layer output and the corresponding downsampling output. The attention layer output

is concatenate with the dropout layer output and finally pass through a U-Net block. The attention block applies a 1×1 convolution to both the upsampled feature map and the input feature map, followed by Batch Normalization. After that, the outputs are summed together and passed through a ReLU activation. A subsequent 1×1 convolution is applied, followed by a sigmoid activation. In the end, the original input feature map is multiplied by the attention map.

Finally, the model's output is equal to the one of the single U-Net (section 4.1).

4.3.1 Strategies

To improve the model performances, loss functions and their combination are investigated.

- **Tversky Loss**: suitable for datasets with significant class imbalance and useful to control penalties for false positives and false negatives with α and β parameters.

$$\text{Index} = \frac{\text{TP} + \text{smooth}}{\text{TP} + \alpha \cdot \text{FP} + \beta \cdot \text{FN} + \text{smooth}}$$

The Tversky Loss is defined as:

$$\text{Tversky Loss} = 1 - \text{Index}$$

- **Focal Loss**: it takes into account the hard-to-classify examples introducing the γ parameter, which reduces the loss contribution from easy examples. The α parameter instead is a factor to deal with class imbalance.

$$\text{Focal Loss} = -\alpha \cdot (1 - y_{\text{pred}})^{\gamma} \cdot \log(y_{\text{pred}})$$

Finally a combined Loss is defined: it is the weighted sum of the two lesses described above. After some trial, used weights are $w_{\text{focal}} = 0.7$, $w_{\text{tversky}} = 0.3$.

5 Results

The model achieves the same performance both in validation and test set. In table 1 there are the scores of the final model.

Set	meanIoU (%)
Train	61.77
Validation	53.72
Test	52.46

Table 1: Mean IoU of the final model

The used receptive field is 53 and the confusion matrix on validation set is reported below (fig. 1).

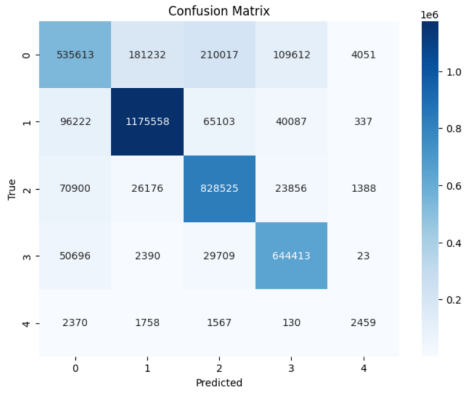

Figure 1: Confusion matrix

6 Discussion

The chosen receptive field is a good compromise to capture the largest object and the smaller one starting from an image 64×128. The meanIoU has a gap between the training and validation meanIoU, suggesting that the model may be overfit.

About class 4, we achieved 2459 right classifications as improvement.

7 Conclusions

The model appears robust and demonstrates good generalization capabilities. We started with mean IoU on the test set of about 0.42040 and unclassified class 4 and we ended up with a score of 0.52461 and 2459 right labels for class 4.

To further improve its performance, regularizers could be introduced to mitigate the gap between training and validation meanIoU. Given the ease of mismatch for some classes, it would also be beneficial to increase the complexity of the decoder, and more in general of the model itself.

The model appears robust and demonstrates good generalization capabilities.

References

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

[2] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018. Accepted to be published in MIDL'18 (Revised Version).

Contributions

Irene Caridi: dataset analysis, preprocessing (dataset cleaning with mask, dataset augmentation), models (single U-Net, best U-Net, strategies), report writing.

Francesca Girolami: dataset analysis, preprocessing (dataset cleaning with CNN, dataset balance), models (single U-Net, double U-Net, strategies), report writing.