AN2DL - Second Homework Report
Four Neurons

Daniel Silva Barradas, Rigers Hado, Christian Mariano, Alessia Testa

danielsilvabarradas, rigershado, christo444, aalessiaa

281330, 252621, 260338, 259852

March 1, 2025

1 Introduction

This project focuses on the application of **deep learning** techniques for image segmentation in the context of Mars exploration. To tackle this challenge, the project focuses on these objectives:

- Develop a **robust segmentation** pipeline from scratch that identifies classes accurately.
- Create **preprocessing techniques** to optimize input data for the segmentation task.
- Implement **augmentation** and mapping strategies to improve the **model generalization.**

2 Problem Analysis

The dataset consists of segmented images of the Martian surface, each paired with a mask that categorizes the terrain into 5 distinct classes based on pixel-level annotations.

- **Image Size**: 64x128
- **Color Space**: Grayscale (1 channel)
- **Input Shape**: (64, 128)

A thorough inspection revealed irrelevant and misleading samples, which were removed to ensure dataset quality and relevance. The processed dataset was divided into training, validation, and test subsets, maintaining a balanced distribution of classes across these splits. The main challenges in this task were centered around data quality and class representation:

- **Irregularities in Data**: Extraneous and ambiguous labels needed extensive preprocessing to maintain the integrity of the training data.
- **Class Imbalance**: The under-representation of the fourth class posed a challenge, increasing the likelihood of biased predictions.

3 Method

The process starts with data preprocessing. Data augmentation helps improving model generalization by artificially increasing the size of the training set through transformations such as flipping, rotating, or zooming, each function having a 50% chance of being applied. Depending on the random outcome, different transformations are applied. By learning from different variations of the images, the model generalizes better and avoids overfitting.

Transformations such as batching and shuffling are applied. Batching improves computation efficiency and gradient stability, and shuffling prevents overfitting and biases in training.

Creating the training and testing datasets allows to map the image paths to their corresponding labels while applying the necessary preprocessing and augmentation steps.

It leverages custom blocks like the *U_net_block* [1], which extracts hierarchical features; *parallel_dilated_convs* [2], which capture multi-scale context through dilated convolutions; and a *global_context_module* [3] that provides long-range dependencies for better understanding of the overall image context.

A batch normalization [4] layer is added after the input layer to normalize the data fed to the *U-net* block. Additionally, the *Squeeze_and_Excitation_block* (SE) [5] is included to adaptively recalibrate feature maps, allowing the model to focus on the most important features.

These enhancements improve the model's ability to accurately segment complex images.

To address class imbalance, the occurrences of each class in the segmentation labels for the training data is computed to analyze the distribution of classes within the dataset. This method helps to identify any imbalances that would affect model performance, highlighting the need for interventions like class weighting or dataset balancing. Then the *Focal Loss* [6] function is implemented. By focusing on hard-to-classify pixels, it improves performance on minority classes. A custom metric *MeanIntersectionOverUnion* is defined to evaluate the model by calculating the Intersection over Union (IoU) for multi-class segmentation while excluding certain classes (background class 0).

The model is compiled with the *AdamW optimizer* [7] and weighted focal loss. Class weights based on the distribution of classes in the dataset, are then applied during training to improve performance on this imbalanced dataset. The custom IoU metric is used for evaluation. While *VizCallback* visualizes predictions at specified intervals, early stopping halts training if validation IoU does not improve, ensuring the best weights are preserved.

4 Experiments

The experiments focused on refining the semantic segmentation model through various iterations and adjustments to the architecture, data, and training techniques. The development process followed this outline:

1. **Initial Model with U-Net**: We started by implementing the standard *U-Net* model, as discussed in class, to serve as a baseline for our semantic segmentation task.

2. **Data Augmentation**: We enhanced the model's robustness by adding data augmentation techniques, including flipping, rotation and zooming.

3. **Focal Loss**: To address class imbalance, we incorporated *Focal Loss* as the loss function.

4. **Bottleneck enhancement**: The model architecture was further modified by introducing new blocks to capture more complex spatial relationships:
   - *Parallel Dilated Convolutions*
   - *Global Context Module*
   - *Squeeze-and-Excitation (SE) Block*

5. **Class Weights and Conv2DTranspose** [8] [9]: We introduced *class weights* to handle class imbalance. Additionally, we replaced *UpSampling2D* with *Conv2DTranspose* for better performance in the upsampling layers. *BatchNormalization* was added to the new blocks for improved convergence and stability. This was found to be the best performing model.

6. **Deep Supervision** [10]: We tested *deep supervision* by adding auxiliary output layers at intermediate levels of the network. However, this approach did not show significant improvements compared to the previous iteration.

7. **DeepLabV3+**[11] [12] [13]: Finally, we experimented with *DeepLabV3+*, a more advanced model for semantic segmentation. However, it suffered from severe

Table 1: Table with Model results. Best results are highlighted in **bold**.

Model	MeanIOU	SparseCat	Weighted Focalloss	ValIOU	Kaggle
Base Model	0.3533	-	-	0.06	0.09
Augmented Data Model	0.3265	0.8517	-	0.4003	0.37
Focal-Loss Model	0.4419	0.7166	0.0490	0.4890	0.47
Bottle-Neck Enhancement	**0.5719**	**1.6903**	**1.6048**	**0.5886**	**0.58**
DeepLabV3+	0.5528	2.8639	2.0085	0.3681	0.44

overfitting, despite adjustments to the architecture and training process.

5 Results

The experiments highlighted a series of interesting details. First of all data augmentation proved to be the single most important factor in gaining a more precise and accurate model, followed by implementing focal loss as the loss function, the enhancement of the bottleneck region and class weight regularization. **A notable detail was that after using add() instead of concatenate in the dilation portion followed a significant gain**. **Adding more U-net blocks or simply expanding our decoder encoder layers didn't have any significant effect other then obtaining a faster convergence and worsening the overfitting problem.** The biggest offender in this regard was the *DeepLabV3* architecture implemented using a *ResNet50* [14] as an encoder. The most successful approach involved the use of class weights, *Conv2DTranspose* layers, and *BatchNormalization.*

6 Discussion

The model obtained at the end is a model that was the most balanced in the terms of feature complexity and the ability to not overfit the data (the result in local testing was very close to the Kaggle one). It's also very compact and fast to train. The limitation with such a small model is given by the fact that it wasn't able to perform better and lacked the ability to deal with such a big class imbalance. In contrast, *DeepLabV3+*, a more complex and deeper architecture, exhibited severe overfitting on the available dataset. This can be attributed to its larger number of parameters and greater depth, which typically require a more extensive training set to generalize effectively. With the smaller dataset provided, *DeepLabV3+* struggled to adapt without memorizing patterns in the training data, leading to diminished performance on unseen data. Such behavior suggests that *DeepLabV3+* requires a significantly larger dataset to completely fulfill its potential.

7 Conclusions

At the end the models were not able to deal with the task given to us. One suffered from over-fitting, whilst the other appeared to be too simple. Data augmentation and enhancement of the bottleneck region seemed to be a very good start, but to further improve the performance of semantic segmentation models and address the limitations of both *U-Net* and *DeepLabV3+*, a new hybrid architecture could be proposed. This model would aim to strike a balance between the efficiency of *U-Net* and the depth and feature extraction capabilities of *DeepLabV3+*, providing a more adaptable and scalable solution. Multiple neural networks each designed to classify a specific class, or just image background, could also be exploited to improve the model's performance. In the future, a better understanding of the data and an improved architecture of this models, while also implementing the aforementioned improvements, could lead to a better result.

References

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Im-

age Segmentation." *arXiv preprint arXiv:1505.04597*, 2015.

[2] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. "Dilated Residual Networks." *arXiv preprint arXiv:1706.05587*, 2017.

[3] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. "GC-Net: Non-local Networks Meet Squeeze-Excitation Networks and Beyond." *arXiv preprint arXiv:1904.11492v1*, 2019.

[4] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." *arXiv preprint arXiv:1502.03167*, 2015.

[5] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-Excitation Networks." *arXiv preprint arXiv:1709.01507*, 2017.

[6] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal Loss for Dense Object Detection." *arXiv preprint arXiv:1708.02002*, 2017.

[7] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization." *arXiv preprint arXiv:1711.05101*, 2017.

[8] Vincent Dumoulin and Francesco Visin. "A Guide to Convolution Arithmetic for Deep Learning." *arXiv preprint arXiv:1603.07285v1*, 2016.

[9] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. "Object Detection with Discriminatively Trained Part-Based Models." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[10] Zhicheng Yan, Xiaohui Shen, Zhe L. Lin, Kazuhito Koishida, and Thomas S. Huang. "Deep Supervision with Intermediate Concepts." *arXiv preprint arXiv:1801.03399*, 2018.

[11] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. "Rethinking Atrous Convolution for Semantic Image Segmentation." *arXiv preprint arXiv:1706.05587*, 2017.

[12] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs." *arXiv preprint arXiv:1606.00915*, 2016.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." *arXiv preprint arXiv:1512.03385*, 2015.

[14] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation." *arXiv preprint arXiv:1802.02611*, 2018.