AN2DL - Second Homework Report
Hyper Parameters Guessers

Paolo Biolghini, Loris Di Girolamo, Sara Cavallini, Riccardo Camellini

Kaggle: fazpaolo, Kaggle: lorisdigirolamo, Kaggle: saracava, Kaggle: riccardo

246816, 249851, 244980, 244779

December 15, 2024

1 Introduction

This project focuses on *image segmentation* using **deep learning** techniques. The dataset provided was a challenging, unclean dataset of **Mars terrain**. The primary objective was to develop a **neural network** capable of accurately classifying each pixel of these images.

2 Problem Analysis

2.1 Dataset Characteristics

The dataset consists of 2615 images initially provided in the shape (64,128,1), which we preprocessed to obtain a shape of (64,128,3) by replicating the single channel across the three color channels. The corresponding annotations have the shape (64,128,1). During preprocessing, we identified and removed several outlier images using ashing methods to improve data quality.

An important challenge with this dataset is the class imbalance. Specifically, class 4 constitutes less than 1% of the total labeled pixels, which required us to implement strategies to mitigate this imbalance during training. To prepare the data for our model, we converted the labels into one-hot encoding format and created batches of images and labels containing 64 samples each.

The dataset was split twice to ensure robust evaluation. First, we created a validation set for early stopping during training. Second, we extracted a small labeled test set to assess generalization and ensure that the model does not overfit to the validation data. This dual splitting approach helps validate our model's performance effectively and reduces the risk of overfitting.

2.2 Main Challenges

Several challenges were encountered during the development of the neural network, with two main issues being:

- **Class Imbalance** and **Loss Function**: the scarcity of pixels belonging to class 4 necessitated the use of oversampling techniques and, more importantly, the selection of an appropriate loss function that accounts for this imbalance.
- **Model Architecture**: designing the network architecture required an in-depth analysis of state-of-the-art (SOTA) models in the literature. Due to project constraints, it was not feasible to leverage pre-trained models. This limitation increased the importance of optimizing the architecture, particularly addressing bottlenecks in the network to ensure efficient feature extraction and representation.

3 Method

Our methods can be summarized with the following key points:

-To address the class imbalance, the dataset was augmented by upsampling instances of class 4.

-A robust data augmentation pipeline was implemented using the Albumentations library to enhance model generalization. Geometric transformations included horizontal and vertical flips, along with Elastic Transformations and Grid Distortions. Pixel-wise transformations, included CLAHE, Random Brightness/Contrast Adjustments, and Random Gamma Transformations. These techniques increased data diversity, reduced overfitting, and enhanced the model's performance.

- In our model, we employ a composite loss function, $\mathcal{L}_{\text{total}}$, to optimize segmentation performance. This loss function combines the **weighted Dice Loss** and the **Categorical Focal Loss**, with the aim of addressing class imbalance and improving segmentation accuracy for minority classes.

The **Dice Loss** [3], $\mathcal{L}_{\text{Dice}}$, is designed to maximize the overlap between the predicted segmentation and the ground truth. It is computed as:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_{i=1}^{N} w_c \cdot y_i \hat{y}_i}{\sum_{i=1}^{N} w_c \cdot (y_i + \hat{y}_i)} \tag{1}$$

The **Categorical Focal Loss** [2], $\mathcal{L}_{\text{Focal}}$, focuses on improving model performance for challenging examples. It is defined as:

$$\mathcal{L}_{\text{Focal}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \cdot (1-\hat{y}_{i,c})^{\gamma} \cdot \log(\hat{y}_{i,c}) \tag{2}$$

The total loss is formulated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Dice}} + \mathcal{L}_{\text{Focal}} \tag{3}$$

where the contribution of each loss is balanced equally ($\lambda = 1$).

This approach leverages the strengths of both **Dice Loss** and **Focal Loss**, ensuring accurate segmentation while addressing the challenges posed by imbalanced class distributions.

-The final architecture is based on a modified **DeepLabv3** framework, incorporating elements from ResNet-50 and a custom dual-path analysis strategy. The encoder leverages a series of down-sampling blocks inspired by **U-Net**, with multiple residual paths designed to preserve information that might otherwise be lost during successive convolutions. The bottleneck utilizes an Atrous Spatial Pyramid Pooling (ASPP) block to capture multi-scale contextual features effectively.

A dual-path design integrates global and detailed feature analysis by merging upsampled bottleneck outputs with mid-level features from earlier layers. This fusion is followed by successive convolutional layers with batch normalization, dropout, and ReLU activation to refine feature representations. The final segmentation map is generated through an upsampling layer and a softmax-activated output, ensuring accurate pixel-wise classification across five classes.

-learning rate scheduler: 3e-4 learning rate with reduce on plateau callback or cosine annealing technique.

As suggested by Karpathy [1], we initially focused on finding a network that overfits on one batch and then on the training set. Subsequently, we aimed to improve generalization by incorporating dropout, enhancing data augmentations, reducing batch sizes, and applying learning rate decay.

4 Experiments

Our journey through the Martian image segmentation challenge led us to encounter several roadblocks. Initially, we attempted to address the problem by creating a simple U-Net structure, making minor adjustments to the teacher code. Despite our efforts—such as stacking two U-Nets, modifying loss functions, and altering the network depth—we couldn't achieve significant improvements. After exploring state-of-the-art (SOTA) architectures, we decided to implement the DeepLab v3 model, hoping for better results.We made several modifications to the bottleneck layers, refined residual paths, and experimented with enhanced feature fusion techniques.

In table, a resume of some experiments

5 Results

Despite the significant effort and several hours invested in improving our network, the results achieved were not as promising as we had hoped. Our best performance on the Kaggle test set was

Table 1: Our experiments over local test set and Kaggle test set.

Model	Mean IoU (local)	Mean IoU (Kaggle)
Teacher Baseline	40.53	44.33
Dual UNet	84.58	43.63
DeepLab v3	61.20	48.17
Dual AttentionNet	89.13	51.1

an IoU of 51%, which, while not a substantial improvement, still represents progress. We achieved an 84.58% IoU on the local validation set; however, this network has the tendency to overlook the fourth class, which subsequently led to a low IoU score for this class on the test set.

6 Discussion

The results from our experiments showcased both strengths and limitations. On the positive side, the network achieved a strong Mean IoU of 89% on the local validation set, demonstrating the effectiveness of our architecture modifications, including the ASPP block and dual-path strategy, alongside robust data augmentation.

However, challenges persisted, particularly the network's struggle to learn the minority class (class 4), resulting in poor performance on the Kaggle test set. The absence of pre-trained weights and the disparity between local and test results further highlighted issues of generalization and potential overfitting.

7 Conclusions

While achieving strong local performance (89.13% IoU), the model struggled to generalize to the Kaggle test set (51% IoU), particularly underperforming for minority classes.

Future improvements could focus on advanced techniques such as feature fusion and hyperparameter optimization. Additionally, addressing class imbalance through strategies like targeted sampling, curriculum learning, and ensemble methods could enhance the model's performance.

Our network doesn't reflect at all our effort and level of commitment to this project; we get lost several times during the trip to reach our final solution but we can say that it had thought us a lot.

7.1 Contribution

Every person in this team collaborated to reach the final solution, everyone with different responsibilities: Loris is responsible for data cleaning, data augmentation, and network structure exploration. Paolo focuses on data augmentation, network exploration, and automated hyperparameter tuning. Sara handles hyperparameter tuning, testing, and network structure exploration. Riccardo is dedicated to hyperparameter tuning and model testing

References

[1] A. Karpathy. A recipe for training neural networks. `https://karpathy.github.io/2019/04/25/recipe/`, 2019. Accessed: 2024-12-14.

[2] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017.

[3] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *arXiv preprint arXiv:1707.03237*, 2017.