AN2DL - Second Homework Report
Team Rookie

Jiaxiang Yi, Yang Hao Mao, Simona Cai, Ying Zhou

rookieyi, leooo, cairookie, yingrookiee

251872, 248391, 252253, 276543

December 14, 2024

1 Introduction

The project's goal was to assign 64x128 grayscale images of Mars terrain, each pixel in these images is categorized into five type of terrain. This job is a semantic segmentation problem that allows us to investigate both the theoretical and practical elements of **artificial neural networks and deep learning**. In the following report we are going to describe how we handled this problem in three stages: **Analysis of the dataset and research of the best data augmentation**; **Build a baseline model**, **Perform to our best model**

2 Approaches

1. Dataset Analysis

   We discovered that there are some "outliers" in the dataset, such as meaningless or irrelevant images, so we did data cleaning, here is distribution of our dataset, which represent number of pixels per class.

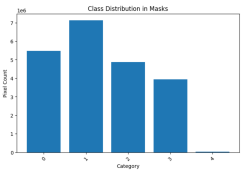

Figure 1: Mask distribution of the filtered dataset

In the Figure 1 We have observed a substantial imbalance in the distribution of samples across the classes in our dataset. Notably, class 4 is significantly underrepresented compared to the other categories, which themselves exhibit a non-uniform distribution. This discrepancy could potentially impact the performance and generalizability of our model. To mitigate this imbalance, it might be necessary to apply various data augmentation techniques to generate new and unseen samples, and another strategy that could help is implementing **loss weighting**, which increasing the penalty for misclassifying underrepresented classes to balance the training focus across all classes. We chose **mean intersection over union metric** as our primary evaluation metric, focusing on its trend in the validation set to keep our analysis straightforward. Regarding dataset splitting, we opted for a 0.90-0.1 split between the training and validation sets due to having only 2,505 images, aiming to maximize training data while ensuring model validation. The test set was already provided separately. To address the increased *risk of over-fitting* with a larger training set, we adopted **early stopping** during the training process. Meanwhile, we preprocessed our dataset dividing each value by 255 thus normalizing the images to stabilize training-validation-test process.

2. UNet from scratch

   Initially, we implemented the basic UNet architecture as introduced during the lecture. This version of the UNet consisted of 3 downsampling and 3 upsampling layers, with no modifications to the original structure. Each UNet block was implemented using simple convolutional layers without additional optimizations or enhancements. This served as our baseline model for subsequent experimentation and comparison with initial MeanIoU of 11.75% on test set.

3. Data Augmentation

   To increase the dataset size and improve generalization, we applied data augmentation, starting with individual transformations like flip, brightness, contrast, and rotation, which worked well, while others like crop, noise, shift, and zoom were less effective. Gradually, we combined the most effective techniques and employed **dynamic augmentation** to prevent *over-augmentation* by adaptively applying transformations. This approach helped maintain data realism and diversity, achieving a final test MeanIoU of 46.48%.

4. Loss Function

   Noticing *over-fitting* due to class imbalance, we implemented a combination of standard and weighted loss functions to penalize misclassification of underrepresented classes more heavily. Specifically, we incorporated:

   Dice Loss, as:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2 \cdot \sum(y_{\text{true,no\_bg}} \cdot y_{\text{pred,no\_bg}})}{\sum y_{\text{true,no\_bg}} + \sum y_{\text{pred,no\_bg}} + \epsilon} \tag{1}$$

   which encourages overlap between predicted (y_pred, no_bg) and true segmentation masks (y_true, no_bg) for non-background regions.

   Focal Loss, as:

$$\mathcal{L}_{\text{Focal}} = -\alpha \cdot (1 - p_t)^{\gamma} \cdot \log(p_t) \tag{2}$$

   applies adaptive weighting to focus on challenging samples, addressing class imbalance.

   Boundary Loss, as:

$$\mathcal{L}_{\text{Boundary}} = \frac{1}{N} \sum \Big[\text{FGBoundary} \cdot \text{FGProb} + \text{BGBoundary} \cdot (1 - \text{FGProb})\Big] \tag{3}$$

   refines segmentation borders by penalizing unclear edges, for requiring precise boundary delineation.

   Weighted Cross Entropy, as:

$$\mathcal{L}_{\text{WCE}} = -\sum_{i=1}^{N} w_{c_i} \cdot y_i \cdot \log(\hat{y}_i) \tag{4}$$

   adjusts the loss by assigning higher weights to underrepresented classes, addressing class imbalance during training.

   To balance these components, we defined a **combined loss function** as:

$$\mathcal{L}_{\text{Comb}} = w_1 \cdot \mathcal{L}_{\text{CE}} + w_2 \cdot \mathcal{L}_{\text{Dice}} + w_3 \cdot \mathcal{L}_{\text{Focal}} + w_4 \cdot \mathcal{L}_{\text{Boundary}} \tag{5}$$

   We tried different weights for w_1, w_2, w_3, and w_4 to used in Deep Supervision section.

   Noticing that our mean IoU *excludes the background class* predictions, we assigned it a weight of 0 in the loss function.

3 Perform to our best model

- ASPP Block

  Our model struggled to learn pixel-level semantics in the context of the entire image. By incorporating an **Atrous Spatial Pyramid Pooling** block at the bottleneck stage, the model effectively captured multi-scale features and improving MeanIoU to 49.44% on test set.

- Transformer

  In addition to ASPP, we tried incorporating a **Transformer block** in the bottleneck for global feature capture, but it only achieved 47.68% with no significant improvement, so we did not proceed with it.

- Normalization

  We continued optimizing the model by applying normalization to improve training efficiency and generalization. Pixel values were scaled during preprocessing, while **BatchNormalization** stabilized training within mini-batches, and **LayerNormalization** normalized features per sample, enhancing global context modeling in Transformer blocks.

- Attention Mechanisms

  To enhance the model's focus on relevant features and suppress irrelevant regions, we introduced the **Attention Gate block** on the skip connection. The Attention Gate effectively integrates multi-scale features and improved our result to 51.6%. Building on this, we further introduced additional attention mechanisms **Channel Attention** and **Spatial Attention** blocks to trying better capture critical characteristics. However, these last modifications did not surpass our best performance.

- Sub-Pixel Convolution

  When encountering a performance bottleneck, we focused on enhancing the decoder and replaced traditional UpSampling with **Sub-Pixel Convolution** to enhance detail and edge quality in semantic segmentation. However, it did not improve MeanIoU significantly, achieving only 47.75%, close to the baseline.

- Deep Supervision

  Finally, we try to incorporated **Deep Supervision**. By applying **weighted loss functions** to outputs from different decoder levels with w_1, w_2, w_3, and w_4 respectively combination of: (1.0,null,null,null),(null, 0.45,0.45,0.1) (null, 0.1, 0.1, 0.8) within 5 , we effectively utilized multi-level and multi-scale features, improving the model's ability to capture both local details and global semantics. This strategy significantly boosted performance, achieving a new benchmark to 61.79% on test set.

4 Results

We finally achieved a 61.79% performance using key components such as Attention Gates, Atrous Spatial Pyramid Pooling, Deep Supervision, and dynamic augmentation. Unexpectedly, a deeper U-Net architecture and a higher number of filters did not improve our model's performance.

5 Discussion

- Strengths and weaknesses

  Our model focuses on enhancing **contextual feature understanding** and improving the **receptive field** to better capture both global and local information. However, it still struggles with fine-grained segmentation tasks, particularly when dealing with complex scenarios involving multiple regions with intricate structures. We did not use a dynamic learning rate because, although it improved training with a smoother curve,it did not achieve the best MeanIoU results.

- Limitations and assumptions

  We considered increasing the depth of the UNet by adding more upsampling and downsampling layers. While a deeper UNet might theoretically handle more complex cases, our experiments showed no significant improvements. Additionally, we experimented with a **Dual-UNet** architecture to further enhance feature extraction and fusion but observed similar results without notable performance gains.

6 Conclusions

- Restate main contributions : Everyone gave their best effort to improve our models.

- Suggest improvements : We did not experiment extensively with different hyperparameters, and our learning rate remained fixed at 0.001.

- Propose future work : Introduce more training strategies, such as warm-up periods, experimented with different hyperparameters and more combination techniques of feature fusion on deep supervision.