AN2DL - Second Homework Report
SWYZ2024

Yahuan Shi, Xiangjin Zeng, Zhanyi Yang, Heyue Wang

yahuanshi00, zxj258, yzy030521, heyuewang

269497, 241931, 276551, 276720

December 14, 2024

1 Introduction

- Preprocess the dataset by normalizing input images and converting labels to a suitable format.
- Develop a deep learning model for accurate **semantic segmentation** of Mars terrain images.
- Preprocess the dataset by normalizing input images and converting labels to a suitable format.
- Build a CNN-based encoder-decoder architecture with regularization and residual connections.

2 Problem Analysis

Here you can discuss your initial analysis of the problem. Consider including:

Dataset Characteristics: The dataset contains grayscale images of size 64×128, each paired with a pixel-wise label mask for five terrain classes (background, soil, bedrock, sand, and big rock).

Main Challenges:

- Addressing the imbalance between classes, especially the dominance of the background class.
- Achieving generalization for unseen test data while avoiding overfitting.
- Maintaining computational efficiency given the constraints of the segmentation task.

Initial Assumptions:

- The model can generalize well with sufficient data augmentation and class balancing techniques.
- High segmentation accuracy directly contributes to improved image-level classification performance.

3 Method

In this section, we detail our approach for addressing the semantic segmentation task using a UNet-based architecture. The primary components of our methodology include the model design, loss function, and training process.

3.1 Model Design

The UNet architecture is a fully convolutional network that consists of a symmetric encoder-decoder structure. The encoder progressively downsamples the spatial resolution while increasing the feature representation, and the decoder upsamples the spatial resolution to reconstruct the segmentation map.

Formally, the output of the model can be represented as:

$$f(X) = \text{softmax}(W \cdot X + b) \tag{1}$$

where W and b represent the learnable parameters of the final layer, and X is the input to the network.

3.2 Loss Function & Metrics

As classical segmentation tasks, categorical cross-entropy loss is applied. In practice, since the serious imbalance among different categories, a weight for each category is assigned. For evaluation, we used a custom implementation of the Mean Intersection over Union (Mean IoU), excluding the background class (class 0).

3.2.1 Loss Function

To handle class imbalance in the segmentation task, we employed a weighted categorical cross-entropy loss. The loss is defined as:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} w_{y_i} \cdot y_i \log(\hat{y}_i) \tag{2}$$

Here, N is the total number of samples, y_i is the ground truth label, $\hat{y}_i$ is the predicted probability, and w_{y_i} represents the class weight associated with class y_i. This weighting helps the model focus on underrepresented classes during training.

3.2.2 Metrics

The MeanIoU is adapted to evaluate model performance especially in semantic segmentation problem. In many training examples, the background occupies a large portion of mask, which in the further training, will results in a bias towards background. The Custom Mean IoU metric, excluding the background class, is defined as:

$$\text{Custom Mean IoU} = \frac{1}{N-1} \sum_{c=1}^{N-1} \text{IoU}_c$$

To exclude the background class ($c = 0$), we use the condition:

$$\text{Mask} = \{G \neq 0\}$$

3.3 Training Strategy

The model was trained with the AdamW optimizer (learning rate: 10^{-2}) and a scheduler to halve the rate upon validation stagnation, with a minimum rate of 10^{-6}. Early stopping was used to prevent overfitting.

4 Experiments

Before introducing regularization, the architecture adopted a standard U-Net structure. While this configuration allowed the model to learn quickly and achieve high performance on the training set, it exhibited clear signs of overfitting. To address these issues, the following key improvements were made to the model:

Table 1: Key modifications

Version 1	**Version 2**
Standard U-Net structure.	Residual connections, Dropout, and L2 regularization.
Adjusted based on `Mean IoU`.	Adjusted based on validation loss (`val_loss`).
Larger batch size (124), faster training speed.	Smaller batch size (32), more stable optimization.
No explicit Dropout or L2 regularization.	Uses Dropout and L2 regularization.

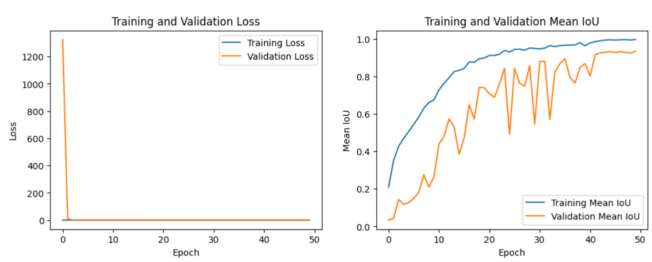

Figure 1: Loss and Mean IoU

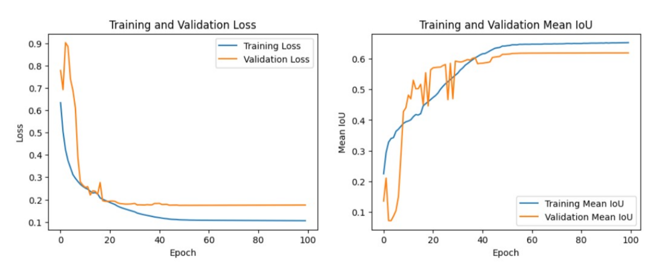

Figure 2: Loss and Mean IoU After

As shown in Figures,
Overfitting:

- The unregularized model exhibits clear overfitting, with training loss approaching zero and a significant gap between training and validation performance.
- Regularization and dropout reduce overfitting, as evidenced by the closer alignment of training and validation metrics.

Generalization:

- The regularized model generalizes better to unseen validation data, as shown by the smoother and more stable validation Mean IoU.

Validation Stability:

- The unregularized model shows significant fluctuations in validation metrics, especially in the Mean IoU, which reflects its sensitivity to data variations.
- The regularized model minimizes these fluctuations, resulting in more consistent validation performance.

Regularization and dropout enhance the model's generalization, reducing overfitting and stabilizing validation performance. Unlike the unregularized model, which overfits to the training set, the regularized model achieves a better balance, making it more robust and reliable.

5 Results

Result: Among all the models we proposed, our best model results **0.61** in accuracy in the hidden test set.
Problems we encountered and our solutions:

- **Outlier Removal**: We removed "alien" outliers from the dataset, but the impact was minimal due to the small number of such samples.
- **Stabilizing Training**: A dynamic learning rate scheduler was implemented to address large oscillations in the validation curve, leading to significantly improved stability.
- **Loss Function Experiments**: Advanced loss functions like Dice Loss and Focal Loss were tested but showed no improvement, so we reverted to simpler methods.
- **Parameter Tuning**: Adjustments in regularization and data augmentation helped achieve the best validation accuracy of 0.63.
- **Oversampling Class 4**: To balance the dataset, we oversampled Class 4, which provided a small but meaningful performance boost.

Unexpectedly, increasing network depth worsened performance due to overfitting caused by the limited dataset. Training accuracy reached 0.92, but submission accuracy dropped to 0.60.

6 Discussion

- The regularization is effective with improvement in smoother loss curve. Several approaches are tested and the simpler but more effective techniques are adapted.
- There exists a significant room for accuracy improvement. Besides, the overfitting issue is revealed with increasing network depth.
- It is assumed that the model generalize with sufficient data augmentation and class balancing, which needs further fine tuning and validation
- The earning rate scheduler improves stability, but does not completely eliminate validation metric fluctuation, which indicates the training process stability issue.

7 Conclusions

Key contributions include data preprocessing, addressing class imbalances, and applying dropout and L2 regularization to reduce overfitting. A custom weighted loss and modified Mean IoU metric effectively handled dataset characteristics.
Future Directions:

- Loss Function Refinement: Testing advanced loss functions tailored for severe class imbalances, like dynamic re-weighting.
- Advanced Architectures: Exploring more sophisticated network designs, such as attention mechanisms or hybrid models.

References