Technical Report

Impact of Point Annotation Density and Learning Rate on Forest Segmentation with Partial Cross-Entropy Loss

1. Introduction:

Semantic segmentation of remote sensing imagery is vital for applications like environmental monitoring, urban planning, and resource management. However, obtaining pixel-perfect segmentation masks for training is costly and time-consuming.

In many practical scenarios, such as in the case of satellite imagery annotation, we are limited to incomplete annotations like bounding boxes, polygons, or even sparse point annotations. This motivates research into methods that can learn effectively from limited or incomplete supervision.

In this study, we focus on training a deep learning model with a custom partial cross-entropy loss function using only point annotations instead of dense masks. We investigate the impact of two critical factors:

- The density of point annotations
- The learning rates.

2. Methods:

2.1 Partial Cross-Entropy Loss Implementation:

The core of our method lies in a custom Partial Cross-Entropy (CE) loss function, which includes a focal loss component to address the class imbalance between the forest and non-forest areas. The formula for our loss function is:

```
pfCE = ∑ (Focal loss (pre, GT) * MASKlabeled) / ∑ MASKlabeled
```

Where `Focal loss (pre, GT)` is the focal loss between predicted probabilities and ground truth classes and `MASKlabeled` is the binary mask that marks the annotated pixels.

Specifically, the implemented method performs the following:

- Calculates the one-hot encoding of the ground truth labels.
- Computes the focal loss component using the predicted class probabilities, one-hot encoded ground truth and the focal loss parameters.
- Applies the mask using multiplication and calculates the sum of the loss values in the labeled areas.
- Returns the normalized loss by dividing by the number of marked points.

2.2. Model Architecture:

We used a Transfer Learning, a modified ResNet50 model pre-trained on ImageNet as the basis for our segmentation network. To adapt it for the segmentation task, the final fully connected layer is replaced by a linear layer with 2 output units representing the two classes (forest and non-forest areas). The modified architecture allows us to leverage the knowledge obtained during ImageNet training and speeds up convergence. We have also replaced the original `conv1` layer of the model to be adapted to our input images.

2.3 Data Preparation:

The "Forest Aerial Images for Segmentation" dataset, which includes images of forested areas and corresponding segmentation masks, is used in our experiments. The dataset consists of RGB aerial images and single-channel mask images. The mask images have values of 0 and 1 corresponding to the "non-forest" and "forest" classes, respectively. The dataset is split into training and validation sets using a standard 80/20 split.

Preprocessing steps:

Resizing: Images and masks are resized to a 256x256 pixel resolution using a bilinear interpolation.

Normalization: Images are normalized using the ImageNet mean and standard deviation, to be able to leverage the pre-trained weights from ImageNet.

Point Annotation Generation: Point annotations are simulated from the ground truth masks by randomly selecting a pre-defined number of pixels belonging to the forest class. The selected points are marked with the value 1 in the simulated point mask.

2.4. Training:

The model is trained with the partial cross-entropy loss function using the Adam optimizer. The optimizer's momentum is 0.9 and the weight decay is 1e-

3. Experiments:

All experiments are performed using a batch size of 16 for 10 epochs. The rest of the hyperparameters are explored in the experiments to evaluate their impact on the model performance.

3.1. Point Sampling Study

- Configurations : [10, 50, 100, 200] points per image
- Fixed parameters:
    - Learning rate : 1e-4
    - Batch size : 16
    - Epochs: 10
    - Optimizer: Adam

Results:

```
Test Metrics
    pixel_accuracy: 0.8103422997282442
    mIoU: 0.5480010995944711
    dice_score: 0.627386886997811
    precision: 0.6851465084964247
    recall: 0.6747443510948683
```

Figure 01: Metrics with 10 points

```
Test Metrics with 50 points
    pixel_accuracy: 0.8257717871619298
    mIoU: 0.5770243054644377
    dice_score: 0.6608008305176366
    precision: 0.7083523524447976
    recall: 0.7058650389410689
```

Figure 02: Metrics with 50 points

```
Test Metrics with 100 points
    pixel_accuracy: 0.8109085919105843
    mIoU: 0.5619209093736314
    dice_score: 0.6477134715957904
    precision: 0.6985854232756208
    recall: 0.6971572892139066
```

Figure 03: Metrics with 100 points

```
Test Metrics with 200 points
    pixel_accuracy: 0.8209677769013347
    mIoU: 0.5755965910989771
    dice_score: 0.6600991246965062
    precision: 0.7095490266491139
    recall: 0.7094464238742252
```

Figure 04: Metrics with 200 points

Key Findings :

- Optimal performance at 50 points
- Diminishing returns beyond 50 points
- 25 points provides good balance of accuracy vs annotation effort

3.2. Learning Rate Study

- Configurations : [1e-3, 1e-4, 1e-5]
- Fixed parameters:
    - Points per image : 50 points
    - Other parameters same as above

Results:

```
Test Metrics with 0.001
    pixel_accuracy: 0.7955001263702453
    mIoU: 0.525682483061151
    dice_score: 0.6090863677828292
    precision: 0.669606973589694
    recall: 0.6611109442255308
```

Figure 05: Metrics with 0.001

```
Test Metrics with 0.0001
    pixel_accuracy: 0.8276098143097939
    mIoU: 0.5839135803839414
    dice_score: 0.6687838881786458
    precision: 0.7119093266108844
    recall: 0.7150757306045646
```

Figure 06: Metrics with 0.0001

```
Test Metrics with 1e-05
    pixel_accuracy: 0.8060198968637247
    mIoU: 0.5518008217846282
    dice_score: 0.6440801714912883
    precision: 0.6969781468550081
    recall: 0.69537957665668
```

Figure 07: Metrics with 0.00001

Key Findings:

- Learning rate 1e-4 provides best balance
- Higher learning rates lead to instability

- Lower learning rates require more epochs

3. Recommandations :

- Use 50 points per image for annotation
- Set learning rate to 1e-4

4. Conclusion:

In conclusion, our study underscores that both point annotation density and the choice of learning rate significantly affect segmentation performance in a limited annotation setup. Therefore, choosing the appropriate values is key for obtaining the best possible performance while minimizing the annotation costs and training time. Future work should explore alternative sampling methods or adaptive learning rate techniques to improve model accuracy and robustness.