University of New Haven

Deep Learning (DSCI-6011-02)

Project Update 1

Team Members:
Parin Patel
Yash Nayi
Yash Raythatha

Date:
11/13/2024

Project 2 update_1

1. Introduction

This project implements a semantic segmentation model based on the DeepLabV3 architecture, using PyTorch to segment and classify pixel-level features in campus environment images. The goal is to utilize a DeepLabV3 model with a ResNet-50 backbone to effectively segment and classify three classes: stairs, doors, and background. The custom dataset created for this project consists of images captured from different parts of the campus, each paired with a mask indicating the segmented regions of interest.

2. Dataset Description

Dataset Composition: The dataset includes 200 images in .JPG format, each annotated with regions representing either doors or stairs. This setup enables the model to focus on distinguishing these two classes from the background and each other.

Classes: There are two main classes:

- Doors
- Stairs

Dataset Link: Dataset on SharePoint

Annotation Tool: Annotations were created using CVAT (Computer Vision Annotation Tool), providing pixel-level precision, which is essential for training a high-quality segmentation model. Each pixel in the annotation mask is assigned a specific color to represent its class:

- **Red** pixels mark door regions.
- **Green** pixels mark stair regions.
- **Black** pixels indicate the background.

Annotation Process

The images were labeled manually in CVAT, where each door and stair were assigned a distinct color code in the mask. This color-coding translates to different class labels in the dataset, allowing the model to learn how to identify each object separately.

Mask Format: The annotated masks are prepared and were exported in Segmentation Mask1.1(.PNG) format.

4. Data Partitioning

Purpose of Partitioning: To train the model effectively, the dataset is divided into three sets:

I. **Training Set (80%):** Used to train the model by updating its parameters to minimize prediction error.
II. **Validation Set (10%):** Used during training to monitor the model's performance on unseen data, which helps detect overfitting.
III. **Test Set (10%):** Used after training to evaluate the model's final performance.

Benefits: This split helps in assessing how well the model generalizes to new data and avoids overfitting, where a model performs well on training data but poorly on unseen data.

5. Normalization and Data Augmentation

Normalization: Images are normalized using mean and standard deviation values from ImageNet (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]). This normalization scales pixel values to a consistent range, improving model convergence and stability.

Data Augmentation: For the training set, random transformations are applied to increase data variability and improve model robustness:

- **Random Resized Crop**: Scales images between 0.5x and 2.0x before cropping to 513×513 pixels.

- **Random Horizontal Flip**: Adds variability by flipping images horizontally.

For the **validation and test sets**, images are only resized to 513×513 and normalized to maintain consistency without additional transformations. Masks are resized to align with the input image dimensions, facilitating accurate pixel-level classification.

6. Methodology

Model Architecture and Configuration

The model used in this project is DeepLabV3 with a ResNet-50 backbone, a widely used architecture for semantic segmentation.

Configuration and Modifications:

- **Transfer Learning**: The model is initialized with pre-trained weights (DeepLabV3_ResNet50_Weights), enabling faster convergence and improved performance on the custom dataset.
- **Classifier Modification**: The final classifier and auxiliary classifier layers are modified to output three classes (doors, stairs, and background), aligning with the dataset's requirements.
- **Layer Freezing**: All layers except the classifier are frozen to preserve learned features from the pre-trained model, while the classifier layers are unfrozen, allowing them to learn specific distinctions for the custom classes.

This setup leverages the strengths of pre-trained weights while adapting the model for accurate segmentation of doors and stairs in the dataset.

7. Evaluation Metrics

Training: The model was trained using **Cross-Entropy Loss**, suitable for multi-class segmentation tasks. This loss function measures the difference between predicted class probabilities and ground truth labels, enabling the model to learn pixel-wise classifications for doors, stairs, and background. The final training loss observed was **1.0559**.

Evaluation Metrics:

- **Average Loss**: The average loss was calculated for both training and validation to monitor convergence and detect overfitting. The validation loss was **1.0489**.
- **Pixel-wise Accuracy**: However, pixel accuracy will be assessed as a simple metric to measure the percentage of correctly classified pixels, in the next update. This metric will provide actual insight into the model's ability to accurately segment each class.

8. Visualization of Results

To assess the quality of predictions, we have visualized the model's output for the first sample in each minibatch. We are doing so to check if the model accurately segments doors and stairs.

Visualization for an image in validation dataset,