UNIVERSITY OF NEW HAVEN

TAGLIATELA COLLEGE OF ENGINEERING
Electrical & Computer Engineering and Computer Science

DEEP – LEARNING

Project 2:

SEMANTIC SEGMENTATION BY MULTI-TASK U-NET

Team:

Likith Kagita Veda Samohitha Chaganti Sai Kumar Reddy Tummeti

1. INTRODUCTION

Semantic segmentation is a critical task in the field of computer vision, where the objective is to assign a class label to each pixel in an image, enabling the identification and separation of various objects within the scene. This technique has wide-ranging applications in areas such as autonomous driving, urban planning, satellite imaging, and environmental monitoring. In this project, we focus on using the U-Net architecture to perform semantic segmentation and detect specific objects such as buildings, trees, and roads in images.

The input data for this project consists of images collected from our university campus and surrounding areas, representing a diverse set of urban and natural environments. These images were annotated using the CVAT (Computer Vision Annotation Tool) to label the objects of interest, and the dataset was carefully prepared to ensure a comprehensive representation of different object classes.

For model development, we utilized U-Net, a deep learning architecture specifically designed for semantic segmentation tasks. U-Net is known for its efficiency in producing accurate pixel-wise predictions, which is crucial for tasks like object detection and scene understanding. We used VGG16 as the encoder backbone, leveraging its pre-trained weights on ImageNet to improve the model's feature extraction capabilities and accelerate the training process.

The purpose of this project is to demonstrate the effectiveness of U-Net for detecting and segmenting urban and natural objects in real-world images. With applications in areas such as city infrastructure mapping, environmental monitoring, and automated urban analysis, this model provides a valuable tool for accurately identifying and segmenting key objects like buildings, roads, and trees. By achieving high-quality segmentation results, we aim to contribute to the development of more robust and scalable computer vision solutions for real-world applications.

2. U-Net with VGG Encoder and Attention-Enhanced Decoder

The U-Net architecture is a widely used model for semantic segmentation, featuring a symmetric encoder-decoder design with skip connections. In this implementation, a **VGG-based encoder** is used for feature extraction, and the **decoder** is enhanced with attention mechanisms to focus on relevant areas and improve segmentation accuracy.

2.1 Encoder: VGG-Based Feature Extraction

The encoder leverages the VGG architecture, pre-trained on large datasets, to extract hierarchical features from input images.

Key Components:

- **Convolutional Layers**:
  - Employ **3x3 filters** to extract spatial features at various resolutions.
  - Capture low-level details such as edges and textures, and high-level patterns like shapes and structures.
- **ReLU Activation**:
  - Introduces non-linearity, enabling the network to learn complex functions.
- **Max-Pooling**:
  - Reduces the spatial dimensions by half at each layer.
  - Allows the network to learn hierarchical features while maintaining computational efficiency.
- **Feature Maps**:
  - Start with **64 filters**, doubling at each level (64 → 128 → 256 → 512).
  - Provide a scalable, multi-level representation of features.

Purpose:

The encoder acts as a powerful feature extractor, capturing both fine-grained and abstract patterns that are crucial for segmentation tasks.

2.2 Decoder: Reconstruction Path with Attention

The decoder reconstructs the spatial resolution of the feature maps and produces the final segmentation output. By incorporating **attention mechanisms**, it enhances focus on relevant regions while suppressing noise.

Key Components:

- **Up sampling Layers**:
  - Restore spatial resolution through operations like bilinear interpolation or transposed convolution.
- **Decoder Blocks**:
  - **Skip Connections**:
    - Merge encoder feature maps with sampled decoder outputs.
    - Retain spatial details lost during down sampling, improving localization accuracy.
  - **Convolutional Layers**:
    - Refine features for precise reconstruction.
    - Paired with **Batch Normalization** to stabilize training.
  - **ReLU Activation**:
    - Maintains non-linear transformations for better learning.
- **Attention Mechanisms**:
  - Highlight critical areas of the feature maps.
  - Suppress irrelevant or noisy features, improving segmentation performance in complex scenes.

Purpose of Attention:

The attention modules selectively emphasize important regions in the feature maps, ensuring the decoder focuses on the most relevant features for segmentation.

2.3 Segmentation Head: Output Prediction

The segmentation head generates the final pixel-wise classification map, assigning a class to every pixel.

Key Components:

- **Final Convolutional Layer**:
  - Reduces the feature maps to match the number of output classes (e.g., 3 for Buildings, Trees, Roads).
- **Activation Function**:
  - **Softmax**: Converts the feature map into probabilities for multi-class segmentation.

Purpose:

Outputs a pixel-wise probability map where each pixel is classified into one of the target classes.

2.4 Architecture Flow

1. **Input**: The raw input image is passed through the VGG-based encoder.
2. **Encoder**:
   - Performs convolution and pooling to generate a hierarchical feature representation.
- **Decoder**:

- Up samples the feature maps. o Uses skip connections to integrate spatial information from the encoder.
- Applies attention mechanisms for selective focus.

- **Segmentation Head**:
  - Produces the final segmentation map with pixel-wise class probabilities.

Why This Architecture is Effective

- **VGG Encoder**:
  - Offers a pre-trained backbone for efficient and robust feature extraction.
- **Attention Mechanisms**:
  - Focus on relevant regions while suppressing noise, improving segmentation in challenging scenarios.
- **Skip Connections**:
  - Preserve spatial details, ensuring accurate localization and reconstruction.
- **Symmetric U-Net Design**:
  - Balances feature extraction and spatial detail recovery, making it ideal for tasks requiring high-resolution output.

3. Dataset

3.1 Data Collection

The dataset was collected in the **University of New Haven** and its surroundings. It contains **207 images** capturing diverse environments such as **buildings**, **trees**, and **roads**. The dataset was curated to provide sufficient variety and coverage for semantic segmentation tasks, ensuring a balanced representation of all target classes.

3.2 Data Partitioning

The dataset is divided into three subsets to facilitate effective training, validation, and testing:

- **Training Set (80%)**:
  - Contains **165 images**, which are used for the model to learn the patterns, structures, and features within the dataset.
- **Validation Set (10%)**:
  - Consists of **21 images**, used during training to evaluate the model's performance and adjust hyperparameters to prevent overfitting.
- **Testing Set (10%)**:
  - Includes **21 images**, reserved for final evaluation to measure how well the model generalizes to unseen data.

This partitioning ensures a structured approach to training, with enough data for both learning and evaluation.

3.3 Data Processing

The following steps were applied to prepare the data for the model:

- **Resizing**:
  - All images and masks were resized to **256x256 pixels**, ensuring consistency in input dimensions for the model.
- **Tensor Conversion**:
  - Images and masks were converted to tensor format to ensure compatibility with PyTorch.

These preprocessing steps are crucial for maintaining uniformity in the dataset and optimizing computational efficiency.

3.4 Normalization

Normalization was performed on the images to standardize pixel values and stabilize training. The pixel values of the RGB channels were normalized using the following statistics:

- **Mean**: [0.485, 0.456, 0.406].
- **Standard Deviation**: [0.229, 0.224, 0.225].

Normalization ensures that the input features have similar ranges, improving the stability and convergence of the model during training.

3.5 Sample Images with Annotations

The dataset includes images annotated with polygons representing the following three classes: 1. **Buildings**: University buildings and other constructions.

2. **Trees**: All vegetation, including trees of varying sizes and shapes.
3. **Roads**: Pathways and paved surfaces.

Annotations are saved in **COCO format** as JSON files, containing polygon-based masks for each class.

3.6 Data Structure

Each sample in the dataset used for training includes:

· Image Tensor: A tensor of shape (3, 256, 256) representing the resized and normalized image.

· Mask Tensor: A tensor of shape (256, 256), where each pixel's value indicates its class:

o 1 for Buildings.

o 2 for Trees. o 3

for Roads.

3.7 Loss Function

The model utilizes cross-entropy loss to train the network for pixel-wise classification. This loss function is specifically designed for multi-class segmentation tasks, where the goal is to assign each pixel in the input image to a specific class (e.g., building, road, tree). The loss function compares the predicted

segmentation mask (from the model) with the ground truth mask, which is the true segmentation of the image, and calculates the error.

The cross-entropy loss function is particularly effective in segmentation tasks as it penalizes the difference between the predicted class probability for each pixel and the actual class. The model aims to minimize this loss during training to improve segmentation accuracy.

Conclusion

This project successfully implemented a U-Net based deep learning model for semantic segmentation of images into categories like buildings, trees, and roads. The model showed significant improvement throughout training, with a reduction in loss and increases in IoU and Dice scores, achieving an IoU of 0.6053 and a Dice score of 0.6890 by the final epoch.

To improve performance further, future work could focus on additional data augmentation, hyperparameter tuning, and exploring more advanced network architectures. Expanding the dataset and incorporating more diverse images could also enhance the model's generalization and robustness, leading to better results on unseen data.

DATASET LINK:

https://unhnewhaven-my.sharepoint.com/:f:/g/personal/lkagi1_unh_newhaven_edu/EqtsplNQSz9NtoLGG0fWi8oBBr_JZxzUuKjlRfKZtoAU1w?e=6EHKCb