Rooftop Usability Detection Using Semantic Segmentation and Classification

1. Introduction

With increasing urbanization and a global shift toward renewable energy, rooftops offer a valuable resource for solar energy harvesting. However, manually identifying suitable rooftop regions for solar panel installation or other structural uses is a labor-intensive and error-prone task. This project proposes an automated solution that leverages deep learning to both segment usable rooftop areas from images and classify rooftops based on their usability.

The proposed system uses a combination of semantic segmentation and image classification. DeepLabV3+ is employed to predict pixel-level masks of usable rooftop space, while EfficientNet-B3 classifies rooftop images into usability categories. Together, these models form a unified, intelligent decision-support pipeline capable of analyzing large datasets of aerial or satellite images.

2. Objectives

The primary goals of the project are as follows:

- Develop a segmentation model to accurately identify usable rooftop regions from input images.
- Implement a classification model to assign each rooftop image to a usability category (e.g., usable, partially usable, unusable).
- Create a modular inference pipeline that supports batch processing, visualization, and data export.
- Transform raw rooftop imagery into actionable insights for solar energy planning, real estate, and urban development.

3. Methodology

3.1 Semantic Segmentation

The first stage of the pipeline involves segmenting rooftop images to identify usable space. The DeepLabV3+ model, built with a ResNet-101 backbone, is used for this task. Images are resized and normalized before being passed through the model. The output is a probability mask indicating the confidence that each pixel belongs to a usable rooftop area.

To improve mask quality, a post-processing pipeline is applied:

- Thresholding at 0.4 converts the probability map into a binary mask.

- Morphological operations (closing and opening) smooth boundaries and remove small noise artifacts.
- Connected component analysis filters out small, isolated regions based on a minimum area threshold.

The results are visualized using Matplotlib, displaying the input image, the predicted binary mask, and a combined overlay where usable areas are highlighted in green.

3.2 Image Classification

Once usable rooftops are identified, the next step is to classify the overall usability of each rooftop image. For this, EfficientNet-B3 is used. The model is trained on a structured dataset where each folder represents a different usability class.

Images are resized to 256×256 pixels, center cropped to 224×224, and normalized using ImageNet statistics. The model is trained using CrossEntropyLoss and the Adam optimizer with a learning rate of 1e-4. The classifier is trained for 10 epochs with a batch size of 16, producing a robust model that can generalize across rooftop types.

4. Deep Learning Architectures

4.1 U-Net

U-Net is a popular convolutional neural network architecture designed for biomedical image segmentation. It has an encoder-decoder structure where the encoder captures the context of the image and the decoder enables precise localization using upsampling. Skip connections between corresponding encoder and decoder layers allow the model to combine low-level spatial information with high-level contextual information, making U-Net particularly effective for pixel-wise segmentation tasks.

4.2 DeepLabV3+

DeepLabV3+ builds upon the DeepLab series by introducing an encoder-decoder structure and atrous spatial pyramid pooling (ASPP) to capture multi-scale context. It uses dilated convolutions to preserve spatial resolution and effectively segment objects at different scales. In this project, DeepLabV3+ with a ResNet-101 backbone is used for its superior performance in capturing fine-grained details in rooftop images.

5. Implementation Details

5.1 Tools and Libraries

Library	Purpose
torch, torchvision	Deep learning framework
segmentation_models_pytorch	Prebuilt DeepLabV3+ model
efficientnet_pytorch	EfficientNet architecture for classification
OpenCV, skimage	Image post-processing
matplotlib, PIL	Visualization and image loading

5.2 File Structure

File Name	Description
predict_images.py	Performs segmentation on a test image with visualization
train_classifier.py	Trains EfficientNet-B3 classifier on labeled rooftop data
inference.py	Integrates segmentation and classification pipeline
main.py	Batch processes multiple images for inference
ml_model_rooftop_version2.py	Utilities for model setup and training
saved_models/	Stores trained model weights and label maps
dataset_classification/	ImageFolder dataset structure for training the classifier
test_images/	Contains test images for segmentation

5.3 Parameters & Settings

Parameter	Value
Segmentation Image Size	128 × 128
Classification Image Size	224 × 224
Segmentation Threshold	0.4
Morphology Kernel Size	3 × 3
Minimum Component Area	100 pixels
Training Epochs	10
Classification Batch Size	16

6. Project Workflow and Code Explanation

6.1 Segmentation Using DeepLabV3+ (predict_images.py)

This script takes a rooftop image and predicts the usable rooftop space using DeepLabV3+ with a ResNet-101 encoder. The model is loaded from a PyTorch checkpoint, set to run on GPU if available. The input is preprocessed and passed through the model to generate a probability mask, which is then thresholded to form a binary segmentation mask. Morphological operations and connected component filtering improve the output quality. Results are visualized as input, mask, and overlay.

6.2 Training the Classifier (train_classifier.py)

This script trains an EfficientNet-B3 model to classify rooftop usability. It loads class-wise image folders using ImageFolder, applies transformations, and trains the model using CrossEntropyLoss and Adam. After training, the model and class label mapping are saved for later inference.

6.3 Unified Inference Pipeline (inference.py)

This script integrates both segmentation and classification. It loads pretrained DeepLabV3+ and EfficientNet models, processes the input image, and generates both a binary mask and a usability class label. This modular setup enables streamlined, automated analysis.

6.4 Batch Processing Script (main.py)

This script handles batch processing of images. It iterates over test images, applies segmentation and classification, and saves the results. Ideal for large-scale rooftop image datasets.

6.5 Model Utility Definitions (ml_model_rooftop_version2.py)

This module includes shared utilities like model initialization, loss functions, and training loops. Keeping this code modular improves reusability and maintainability.

6.6 Data and Output Structure

- **test_images/**: Images to test segmentation.
- **dataset_classification/**: Class-labeled images for training.
- **saved_models/**: Stores trained model files and label maps.

7. Results

7.1 Segmentation Output

The segmentation model effectively highlights usable rooftop regions, reducing false positives with post-processing. Overlay images provide interpretable visual output.

7.2 Classification Output

The classifier accurately categorizes images into usability classes. Outputs can be saved in structured formats like CSV or JSON for easy integration.

8. Applications

- **Solar Panel Planning**: Automatically identify optimal rooftops for solar installations.
- **Urban Planning**: Assist in infrastructure development and zoning.
- **Disaster Management**: Evaluate rooftop damage from aerial imagery.
- **Real Estate**: Provide rooftop usability insights in property listings.

9. Future Enhancements

- **Geospatial Integration**: Map results to GPS coordinates.
- **Multi-class Segmentation**: Segment rooftop obstacles.
- **Temporal Monitoring**: Analyze rooftop changes over time.
- **Web Interface**: Enable GUI-based analysis with Flask/FastAPI.
- **Cloud Deployment**: Deploy as scalable API services.

10. Conclusion

This project presents a deep learning-based pipeline for rooftop usability detection. By integrating semantic segmentation and classification models, it offers a scalable, accurate, and interpretable solution for urban and solar energy applications. The modular design allows for further enhancements and real-world deployment.