Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

DLRV Project Proposal

Shadow Casting Object Segmentation

Kai Glasenapp

Sai Mukkundan Ramamoorthy

Shrikar Nakhye

Supervised by

Prof. Dr. Sebastian Houben

September 2025

1 Introduction

Semantic segmentation in aerial imagery is a critical task for smart city planning, urban monitoring, and emergency response. This project explores the segmentation of shadow-casting objects (e.g., buildings, trees) from aerial RGB images, focusing on the trade-off between segmentation accuracy and computational efficiency.

The work was carried out in the context of the Hackathon about Segmentation of Aerial Images / Satellite Images by City of Bonn, with the aim of producing a lightweight segmentation pipeline that can operate under real-world time and hardware constraints.

2 Objectives

- Evaluate deep learning segmentation models (U-Net, Mask-RCNN).
- Optimize models for speed and efficiency while maintaining accuracy.
- Perform inference on real aerial images of Bonn City.
- Benchmark models with metrics: IoU, F1, inference time, GPU memory usage.

3 Dataset

The dataset consists of high-resolution aerial imagery provided by the City of Bonn (2024). Originally captured with an *IGI UrbanMapper-2* camera at 2.5 cm/pixel resolution, the data was downsampled to 10 cm/pixel to comply with privacy regulations. The downsampled dataset (RGB only) is reduced to approximately 70 GB, allowing efficient training and inference on mid-range GPUs.

Preprocessing steps included:

- Cropping into smaller patches for training.
- Removal of NIR channel.
- Integration of hackathon crowd-sourced annotations.

4 Hardware and Environment

- OS: Linux (Kernel 6.8.0)
- CPU: Intel Core i5-12400 (6 cores, 12 threads)
- GPU: NVIDIA GeForce RTX 4070 (16 GB, CUDA 12.5)
- Frameworks: PyTorch

5 Methodology

6 Model Architectures

- **U-Net:** A classical encoder-decoder segmentation architecture originally designed for biomedical image segmentation [2]. It consists of a contracting path (encoder) that captures context by progressively downsampling the input, and an expansive path (decoder) that enables precise localization through upsampling and concatenation with high-resolution features from the encoder (skip connections). These skip connections allow U-Net to combine semantic information (from deep layers) with fine-grained spatial details (from shallow layers), making it highly effective for pixel-level prediction tasks.

  Why U-Net for this task?

  - Performs well with limited training data due to efficient use of context and localization.
  - Skip connections preserve fine spatial details, crucial for distinguishing small or thin structures.
  - Lightweight compared to more complex models, enabling faster training and inference.
  - Proven strong performance across domains, making it a reliable baseline for remote sensing and 3D vision tasks.

- **Mask R-CNN:** A region-based convolutional neural network for instance segmentation, extending Faster R-CNN by adding a parallel mask prediction branch. Widely used for object-level precision in computer vision tasks such as autonomous driving and aerial imagery analysis.

6.1 Training Strategy: U-Net

- **Loss Function:** Cross-Entropy Loss.
  - Suitable for multi-class segmentation (foreground vs. background).
  - Penalizes pixel-level misclassification and provides stable gradients.
  - Efficient and supported in PyTorch.
- **Training Approach:**
  - Dataset split: 80% training, 20% validation using `train_test_split`.
  - Parameters updated via backpropagation with Adam optimization.
  - Validation monitored with IoU and F1-score.
  - Metrics and system resource usage logged for reproducibility.
  - Training for 20 epochs (balanced cost vs. convergence).
- **Evaluation Metrics During Training:**
  - Intersection over Union (IoU).
  - F1-score (precision-recall balance).
- **Hyperparameters:**
  - Learning Rate: 1×10^{-4}
  - Optimizer: Adam
  - Batch Size: 4
  - Number of Epochs: 20
  - Image Size: 512×512

  - Encoder Backbone: ResNet-34 pre-trained on ImageNet

- **Why This Strategy?**
  - Balances stability (small LR, Adam) with speed (transfer learning).
  - Cross-Entropy provides a strong baseline.
  - IoU and F1 ensure segmentation quality is captured beyond pixel accuracy.
  - Hyperparameters tuned for limited GPU memory.

6.2 Training Strategy: Mask R-CNN (Placeholder)

[Add details about Mask R-CNN loss functions, backbone, training schedule, and evaluation metrics.]

6.3 Evaluation Metrics

To ensure fair and comprehensive comparison across different models, the following evaluation metrics were considered:

- **Intersection over Union (IoU):** Measures the overlap between predicted and ground-truth segmentation masks.
- **F1 Score:** Balances precision and recall, useful when classes are imbalanced.
- **Inference Time:** Average time taken to process one image (ms per image).
- **Training Time:** Total time required for training (measured in hours).
- **GPU Memory Usage:** Peak VRAM consumption during training and inference.

7 Results and Discussion

7.1 Quantitative Results

Table 1 summarizes the benchmark results.

Model	IoU	F1	Inference (ms)	VRAM (GB)
YOLOv8-seg	XX	XX	XX	XX
U-Net	0.742	0.841	52	5.65
Mask-RCNN	XX	XX	XX	XX

Table 1: Comparison of models

7.2 Qualitative Results

Figure 1 shows the training curves for U-Net, illustrating the decrease in training loss and the corresponding improvement in validation IoU and F1-score over 20 epochs.

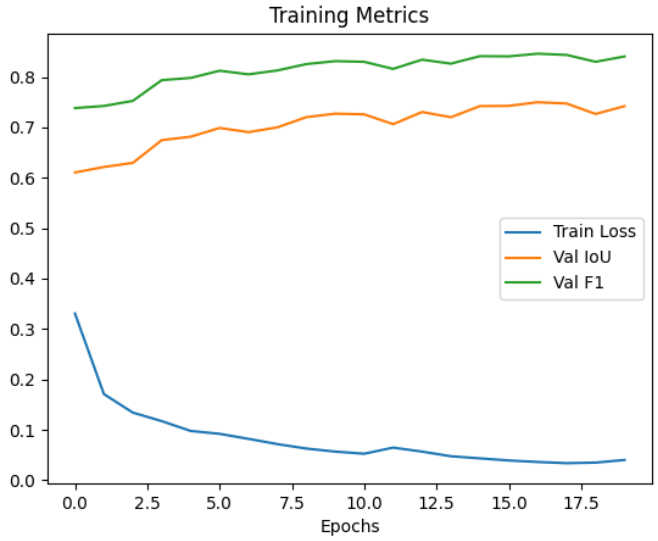

Figure 1: U-Net Training Metrics

Figure 2 shows example segmentation outputs for U-Net compared to other models (placeholders).

7.3 Discussion

- **U-Net:** U-Net achieved a validation IoU of 0.742 and F1-score of 0.841, demonstrating strong pixel-level segmentation performance. The model converged steadily within 20 epochs, with training loss decreasing from 0.33

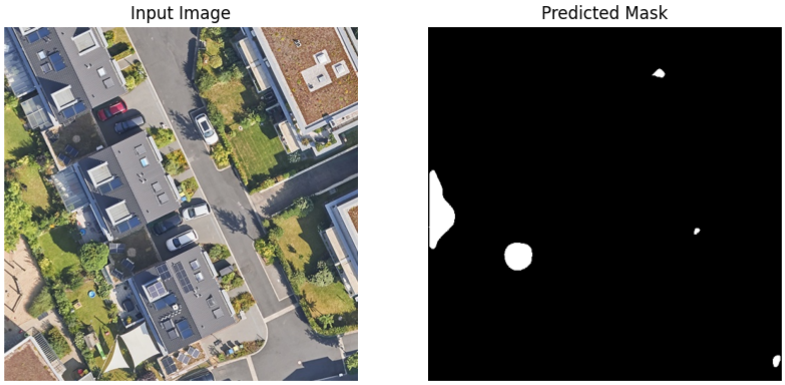

Figure 2: Example segmentation masks U-Net

to 0.04 and validation metrics stabilizing around epoch 15. Inference was efficient at approximately 52 ms per image, with a VRAM footprint of 5.65 GB on the RTX 4050 Laptop GPU. This indicates that U-Net provides a solid balance between performance and computational resource usage.

- **Mask R-CNN:** *[Add results and discussion once trained; typically excels in instance-level precision but may be slower and use more VRAM than U-Net.]*
- **Challenges:** Despite good performance, shadow occlusion and variable sunlight conditions remain challenging for all models. Fine-tuning data augmentation and loss functions may further improve robustness.

8 Documentation and Reproducibility

8.1 Minimum Working Example

Listing 1: Running PrithviVision

```
git clone https://github.com/ItsShriks/
   Shadow_Casting_Object_Segmentation.git
cd PrithviVision
conda env create -f Essentials/dlrv.yml

conda activate dlrv


# Training
python src/train_unet.py -h

# Inference
python inference_unet.py -h
```

8.2 Repository Structure

```
PrithviVision/
├── Essentials/   (environment & setup files)
│   └── dlrv.yml   (Conda environment file)
├── Proposal/   (project proposal documents)
├── dataset/   (training/validation data)
│   ├── unet_dataset/   (for U-Net segmentation)
│   │   ├── images/   (input images)
│   │   └── masks/   (ground truth masks)
│   └── yolo_dataset/   (for YOLO object detection)
│       ├── train/   (training images & labels)
│       └── valid/   (validation images & labels)
├── outputs/ plots/   (results, segmentation outputs)
├── src/   (source code)
│   ├── train_unet.py   (U-Net training script)
│   └── inference_unet.py   (U-Net inference script)
├── .gitignore   (Git ignore configuration)
└── README.md   (project overview & usage guide)
```

[1]

9 Conclusion and Future Work

Future directions for this research include:

- Applying model quantization and optimization for efficient deployment on edge devices such as NVIDIA Jetson.
- Expanding to larger and more diverse datasets to improve robustness and

generalization across different environments.

References

[1] Kai Glasenapp, Sai Mukkundan Ramamoorthy, and Shrikar Nakhye. Prithvivision: Shadow casting object segmentation. `https://github.com/ItsShriks/PrithviVision`, 2025.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351, pages 234–241. Springer International Publishing. ISBN 978-3-319-24573-7 978-3-319-24574-4. doi: 10.1007/978-3-319-24574-4_28. URL `http://link.springer.com/10.1007/978-3-319-24574-4_28`. Series Title: Lecture Notes in Computer Science.