Scientific Computing, Modeling & Simulation
Savitribai Phule Pune University

Master of Technology (M.Tech.)
Programme in Modeling and Simulation

Internship Project Report

Solar panel optimization using AI-ML

Famit Anil Dongarwar
MT2305

Academic Year 2024-25

Scientific Computing, Modeling & Simulation
Savitribai Phule Pune University

Certificate

This is certify that this report titled

Solar panel optimization using AI-ML

authored by

Famit Anil Dongarwar (MT2305)

describes the project work carried out by the author under our supervision during the period from January 2025 to June 2025. This work represents the project component of the Master of Technology (M.Tech.) Programme in Modeling and Simulation at the Department of Scientific Computing, Modeling & Simulation, Savitribai Phule Pune University.

Abhay

Dr. Abhay Tiwari, Founder and CEO
SPATIALTY.AI, Bangalore, India

Dr. Deepashri Saraf, Program Manager
SPATIALTY.AI, Bangalore, India

Dr. Bhalchandra Pujari, Internal Guide
SCMS-SPPU, Pune, India

Prof. Arun Banpurkar, Head
SCMS-SPPU, Pune, India

Scientific Computing, Modeling & Simulation
Savitribai Phule Pune University

Author's Declaration

Statement on Authorship. This document titled

Solar panel optimization using AI-ML

authored by me is an authentic report of the project work carried out by me as part of the Master of Technology (M.Tech.) Programme in Modeling and Simulation at the Department of Scientific Computing, Modeling & Simulation, Savitribai Phule Pune University.

Statement on Plagiarism. In writing this report, I have taken reasonable and adequate care to ensure that material borrowed from sources such as books, research papers, the internet, etc., is acknowledged as per accepted academic norms and practices in this regard. I have read and understood the University's policy on plagiarism (`http://unipune.ac.in/administration_files/pdf/Plagiarism_Policy_University_14-5-12.pdf` or the latest version thereof).

Statement on the use of LLMs/AI. I further certify that

1. any consultations with LLMs/AI tools have been duly acknowledged;
2. any material borrowed from LLMs/AI tools (including code, text, images, etc.) has been duly acknowledged; and
3. I understand that I am the one who is responsible for the work and the results presented in this report including any measures of accuracy or quality, etc.

Famit Anil Dongarwar
MT2305

Contents

Acknowledgements

I am sincerely thankful to **Abhay Tiwari**, an guide from my company Spatialty.ai, for his helpful advice and for sharing his experiences, which provided me with a better understanding of the challenges involved in this field.

Also, I would like to acknowledge **Deepashri Saraf**, additional guide from Spatialty.ai, who continuously there to help, allowing me to implement complex concepts effectively.

Furthermore, I would like to express my deepest gratitude to **Prof. Bhalchandra Pujari** for his invaluable guidance, encouragement, and support throughout this project on Solar panel optimization. His expertise and insightful feedback were instrumental in shaping the direction of this work.

I would like to thank ChatGPT for being an incredibly helpful tool during my project. From fixing bugs in my code to helping me articulate my ideas more clearly in the report, it served as a reliable guide whenever I was stuck or unsure.

Thank you all for your contributions, without which my this project would not have been possible.

Abstract

Rooftop solar panel installations often face challenges such as complicated roof shapes, shading from nearby objects, and changing weather conditions, all of which can reduce how much energy they generate. This project aims to create an AI and machine learning system that makes solar panel placement more efficient. Using satellite images and computer vision, the system will analyze the shapes and features of the roof. Machine learning will help figure out the best way to arrange panels for maximum energy output, predict sunlight availability by looking at factors like cloud cover and air particles, and identify spots on rooftops that avoid shading and obstacles. The goal is to improve energy efficiency, cut installation costs, and make solar power easier to adopt on a larger scale all through smart, automated, and data-driven solutions.

Chapter 1

Introduction

Solar energy is one of the most promising and rapidly growing renewable energy sources worldwide. It is clean, abundant, and sustainable, making it a key player in the global transition toward low-carbon energy systems. Rooftop photovoltaic (PV) systems, in particular, provide a practical and scalable solution for residential, commercial, and industrial energy needs.

However, the efficiency of a solar PV system heavily depends on how and where the panels are installed. Key factors influencing energy output include:

- Rooftop geometry and available area
- Sunlight exposure throughout the day and year
- Shadows from nearby structures or rooftop features
- Panel spacing and layout strategy

Designing an optimal panel layout manually requires detailed site analysis, which is labor intensive and prone to human error. To overcome these limitations, modern research and industry are turning to Artificial Intelligence (AI) and Machine Learning (ML) technologies.

AI-ML in Solar Optimization

AI and ML, especially in the form of deep learning and computer vision, have shown remarkable success in automating complex tasks such as image segmentation, object detection, and pattern recognition. In the context of solar panel optimization, these technologies enable:

Rooftop Detection: Identifying and segmenting usable rooftop[2] areas from aerial or satellite imagery using convolutional neural networks (CNNs) or segmentation models like U-Net.

Shadow Detection: Detecting shadow [1] regions caused by nearby structures or obstructions to avoid panel placement in low-irradiance zones.

Layout Optimization: Automatically designing efficient panel layouts that maximize coverage and energy generation while avoiding constraints such as shadows and boundary violations.

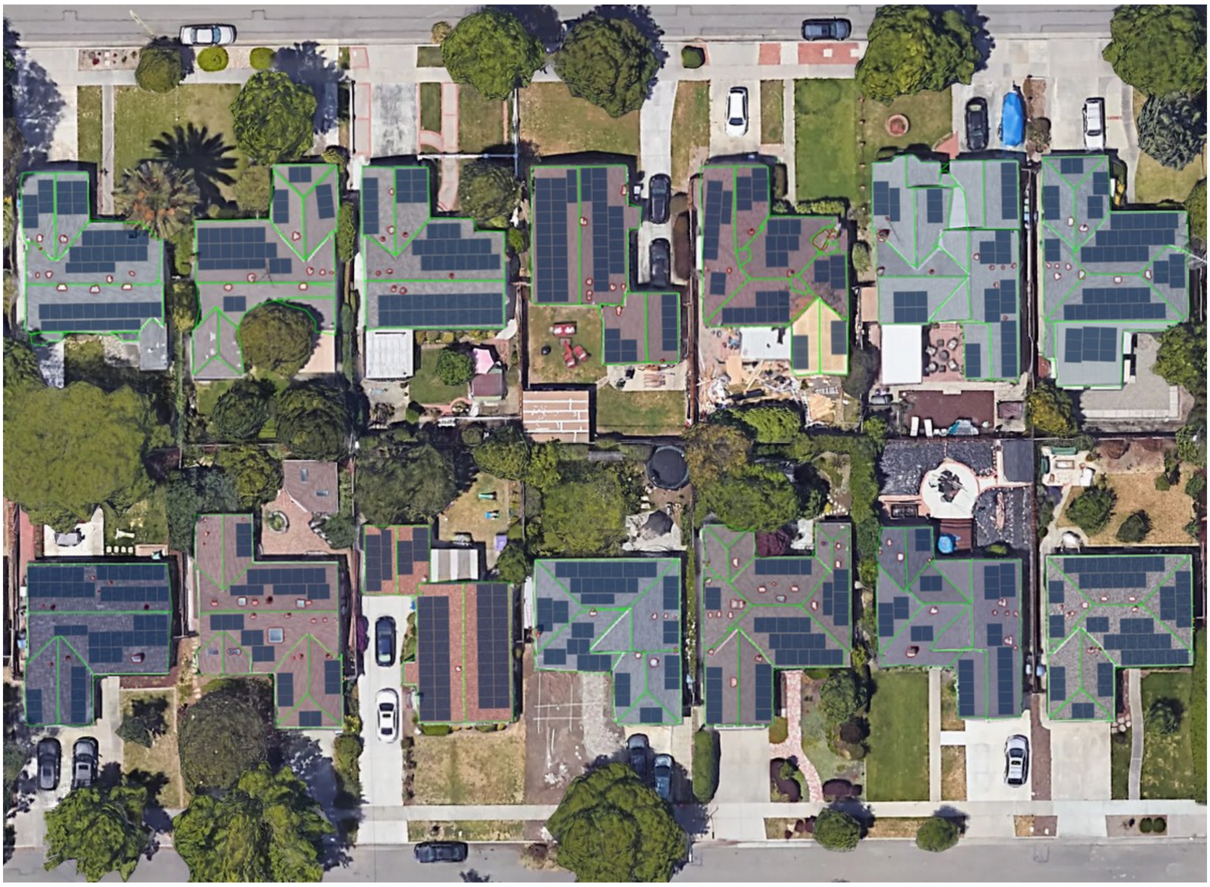

Figure 1.1: Satellite Image with Solar Panel Installation

1.1 Domain background

The global demand for renewable energy is increasing, driven by concerns about climate change, increasing energy needs, and the decreasing supply of fossil fuels. Among the different renewable energy sources, solar energy is as one of the most abundant and sustainable options. Photovoltaic (PV) systems[5], particularly rooftop solar panels, provide a flexible and scalable way to harness solar power in both urban and rural areas.

Despite their potential, getting the most efficiency and return on investment (ROI) from solar panel installations can be quite challenging. Several factors come into play, such as the size and shape of the rooftop, its orientation, shadows cast by trees or buildings, and variations in sunlight throughout the seasons. All of these can significantly impact how much energy the solar panels can generate.

Traditionally, planning the layout of solar panels has involved manual site surveys and decisions based on rules of thumb. This process can be time-consuming, prone to errors, and often leads to less-than-optimal results. As cities grow denser and rooftops become more complex, there's a greater need for automated and intelligent methods to optimize where solar panels are placed.

Recent advancements in Artificial Intelligence (AI) and Machine Learning (ML), especially in the field of computer vision, provide powerful solutions to these challenges. Using aerial or the satellite images, the deep learning models can:

- Detect rooftop area and outline rooftop areas
- Identify the regions which are shaded
- Analyze geometry and the orientation of rooftop.
- Recommend the best panel placements while avoiding shaded areas and ensuring safety

1.2 Problem statement

Solar panel optimization using AI-ML

The rapid adoption of solar energy as a sustainable power source has highlighted the need for optimizing rooftop solar panel installations. However, determining the ideal number, arrangement, and positioning of panels on rooftops remains a complex challenge. Key factors such as rooftop geometry, shading effects, and environmental conditions significantly impact solar energy generation. This study aims to develop an AI-driven solution that integrates satellite imagery, computer vision (CV), and machine learning (ML) to enhance rooftop solar panel deployment.

Key objectives:

i. To maximize the efficiency and performance of solar panels by identifying optimal panel arrangements.

ii. To estimate sunlight availability by analyzing cloud cover, aerosols and atmospheric conditions derived from satellite data.

iii. To identify the most suitable rooftop locations for panel placement, minimizing shading and structural obstructions.

Chapter 2

Dataset

2.1 Data description

Indian Rooftop Dataset from Kaggle, as my project is all about rooftop detection and panel fitting, so in this dataset i got images which have rooftop and the images are taken from satellite so this dataset is totally relevant to my problem statement.

2.2 Exploratory data analysis

2.2.1 Dataset Overview

- **Image Dimensions:** Checked the width and height of images to understand the resolution distribution. This helps decide the appropriate input size for model training.
- **Number of Images:** Counted the number of images in training and validation sets.
- **Annotation Format:** Verified that the dataset follows the COCO format[3], with annotations for:

  1. Rooftop masks
  2. Shadow masks
  3. Bounding boxes

2.2.2 Class Distribution

- **Number of Instances per Class:** Counted how many rooftop and shadow masks were present across the dataset to check for class imbalance.
- **Mask Area Distribution:** Analyzed the pixel area covered by each class (rooftop vs shadow). This helped identify:

  1. Imbalance in mask sizes
  2. Potential bias in the model if one class dominates

2.2.3 Visual Inspection

- **Sample Image Plotting:** Plotted random images along with their rooftop and shadow masks to:

1. Visually verify annotation correctness
2. Understand the visual complexity (e.g., shadows on non-rooftop regions, rooftops with complex shapes)

- **Overlay Masks:** Displayed masks over the images to verify alignment and quality.

2.2.4 Image Quality and Diversity

- **Brightness and Contrast Checks:**Checked variation in lighting, which can impact model performance.
- **Geographical Diversity:** If available, reviewed how rooftops differ across locations (e.g., flat vs sloped, urban vs rural) to assess generalization needs.

2.2.5 Size and Shape of Rooftops

- **Extracted statistics on rooftop shapes using:**
  1. Bounding box aspect ratios
  2. Contour perimeter and area
- Irregularity or complexity of rooftop boundaries (e.g., L-shaped, polygonal)
- Helps in customizing panel layout fitting algorithms.

2.2.6 Shadow Patterns

- **Shadow Coverage:** Analyzed the percentage of rooftop area covered by shadows.

2.2.7 Annotation Quality Check

- Identified missing or poorly labeled masks.
- Checked for overlapping masks between shadow and rooftop, which must be handled carefully during model training and solar panel fitting.

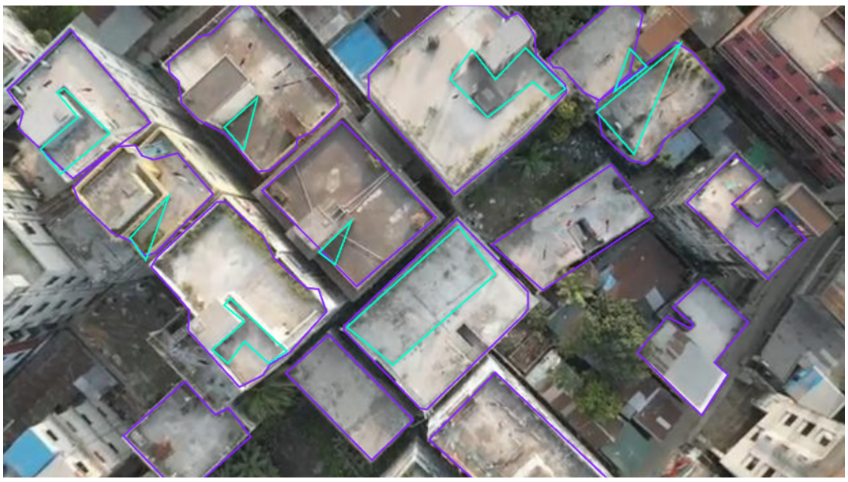

Figure 2.1: Image Annotation with different labels

2.3 Data preprocessing

2.3.1 Loading Images and Annotations

- Image Input: Loaded high-resolution aerial/satellite images, usually in .jpg or .png format.
- COCO Annotations: Used COCO-format .json files to extract:
  1. Segmentation masks for rooftops and shadows
  2. Image and annotation IDs
  3. Class labels (rooftop = 1, shadow = 2, etc.)
- Libraries used: pycocotools, cv2, PIL, numpy, json, etc.

2.3.2 Mask Generation

- From polygon points in COCO annotations, binary masks were created:
  1. One-hot encoded masks or multiclass masks (e.g., rooftop = 1, shadow = 2, background = 0)
- Each mask corresponds to the same resolution as the original image.
- Result: For each image, we now have a corresponding segmentation mask with pixel-wise class labels.

2.3.3 Image and Mask Resizing

- All images and masks were resized to a fixed dimension (e.g., 256×256 or 512×512) for:
  1. Model compatibility
  2. Consistent batch processing
- cv2.resize or torchvision.transforms.Resize was used.

2.3.4 Normalization

- Pixel values of images were scaled from [0, 255] to [0, 1] (or standardized using ImageNet mean/std if using pretrained encoders).

2.3.5 Data Augmentation

To improve generalization and avoid overfitting, several augmentations were applied consistently to both images and masks:

- Geometric Augmentations:
  1. Horizontal/vertical flips
  2. Rotation (e.g., ±30 degrees)
  3. Scaling, cropping, translation
- Photometric Augmentations (only to images):

1. Brightness/contrast adjustment
2. Color jitter
3. Gaussian noise or blur

- Libraries used: Albumentations, torchvision.transforms, imgaug, etc.

2.3.6 Dataset Splitting

- Dataset was split into training, validation, and optionally test sets (e.g., 70-20-10 or 80-20).
- Ensured balanced distribution of rooftop and shadow examples across sets.

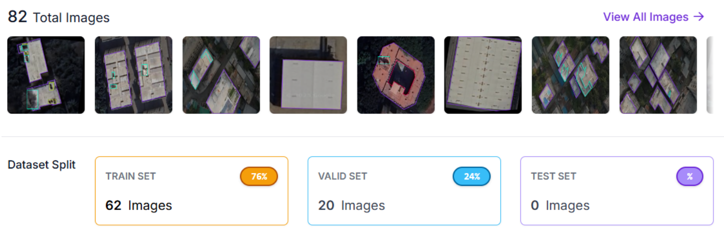

Figure 2.2: Data splitting for train and validation

2.3.7 Data Formatting for Model Input

- Transformed data into tensors:

  1. Images: shape [C, H, W] (e.g., [3, 256, 256])
  2. Masks: shape [H, W] or one-hot [numclasses, H, W]

- Batched using DataLoader with appropriate shuffling, batching, and multiprocessing.

2.3.8 Final Output of Preprocessing

For each sample:

- Preprocessed Image Tensor: float32, normalized, resized
- Corresponding Mask Tensor: integer class values (or one-hot)
- Ready for input into the segmentation model (e.g., U-Net)

Chapter 3

Models and methods

3.1 U-Net Model

U-Net is a popular convolutional neural network architecture specifically developed for image segmentation, particularly in the biomedical domain. It was originally introduced in the paper "U-Net: Convolutional Networks for Biomedical Image Segmentation" [4]. The architecture was designed to deliver accurate segmentation results even when the amount of labeled training data is limited—an issue common in medical imaging.

Structure of the U-Net Architecture

The U-Net architecture is characterized by two main components: a contracting path (encoder) and an expanding path (decoder).

The contracting path functions similarly to a typical CNN, extracting increasingly abstract features through a series of convolutional and pooling layers. This part captures the context of the image by gradually reducing the spatial dimensions while increasing the depth of the feature maps.

The expanding path, on the other hand, is responsible for reconstructing the spatial information. It uses upsampling layers to increase the resolution and combines these with corresponding features from the contracting path through skip connections. These skip connections help in preserving fine-grained spatial details that are important for accurate segmentation.

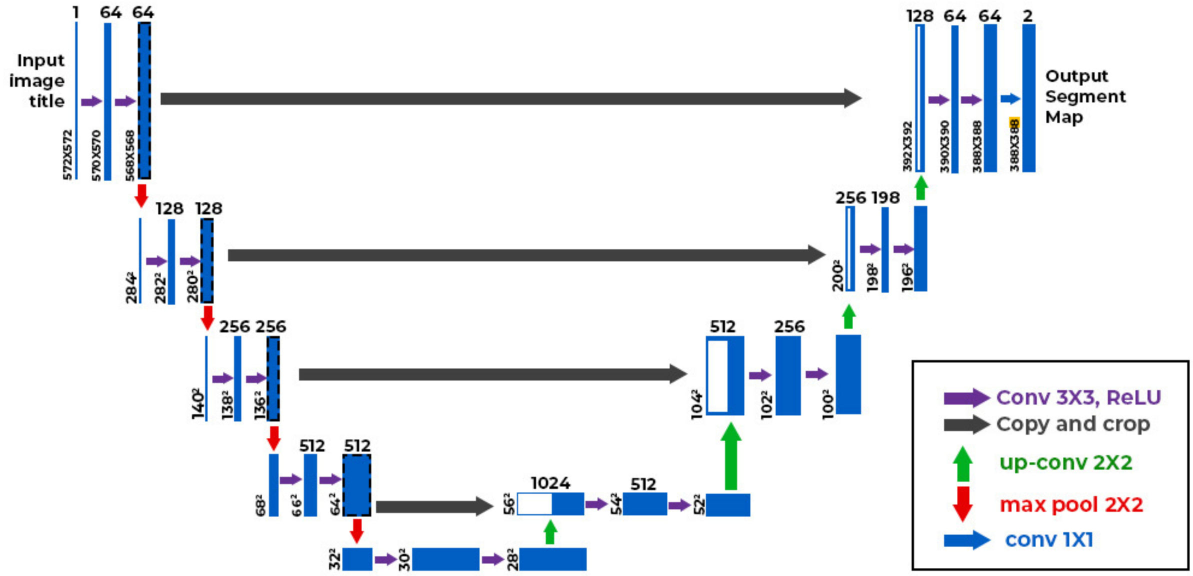

Figure 3.1: UNet Model Architecture

In the original version of U-Net, the input is a grayscale image of size 572×572×1, and the output is a binary segmentation map of size 388×388×2. The reduction in size is due to the absence of padding during convolution. However, using padding can help retain the original spatial dimensions in the output.

As the image goes through the encoder, its width and height decrease while the number of channels (or feature maps) increases. This helps the network learn more complex, high-level representations. At the bottleneck, the architecture reaches its narrowest spatial size, often something like 30×30×1024.

The decoder then progressively upsamples the feature maps, reducing the number of channels and reconstructing the image to its original size. Thanks to skip connections from the encoder, the decoder can make more precise predictions by leveraging both high-level context and low-level details.

Ultimately, the output is a segmentation map where each pixel is classified commonly into background or foreground depending on the task.

1. Skip Connections and Feature Fusion

One of the defining features of U-Net is its use of skip connections. These connections copy feature maps from the encoder and concatenate them with the corresponding decoder layers. This fusion of low-level spatial features with high-level abstract features helps the model recover fine-grained details that are crucial for precise segmentation, especially along object boundaries. Without skip connections, the network may lose critical spatial information during downsampling, which can degrade segmentation accuracy.

2. Convolution and Upsampling Details

Each block in both the encoder and decoder typically consists of two 3×3 convolutional layers followed by a ReLU activation, with batch normalization optionally added for training stability. In the encoder, downsampling is achieved using 2×2 max pooling with a stride of 2. In the decoder, upsampling is performed using transposed convolutions (also called deconvolutions), which learn how to upsample more effectively than fixed methods like nearest-neighbor or bilinear interpolation.

3. Output Layer and Activation

The final layer of the U-Net is usually a 1×1 convolution that maps the output to the desired number of classes:

- For binary segmentation, the output is a single-channel map with sigmoid activation.
- For multiclass segmentation, the output is a multi-channel map (equal to the number of classes) with softmax activation applied along the channel dimension.

4. Padding and Spatial Dimensions

As you mentioned, the original U-Net does not use padding, which results in a reduction of the output size. However, modern implementations often use "same" padding to ensure the input and output sizes match. This is especially useful in applications where the output must align pixel-wise with the input, such as in medical imaging or rooftop segmentation.

5. Model Flexibility and Variants

U-Net is highly flexible and has inspired numerous variants:

- U-Net++: Introduces nested and dense skip pathways for better feature propagation.
- Attention U-Net: Integrates attention gates to focus on relevant features and suppress irrelevant regions.
- Residual U-Net: Incorporates residual connections from ResNet to improve gradient flow in deep networks.
- Pretrained Encoder U-Net: Uses encoders like ResNet, EfficientNet, or VGG pretrained on ImageNet to improve feature extraction on small datasets.

These variants can significantly enhance performance, especially when training data is limited or noisy.

6. Application to Solar Panel Optimization

In the context of solar panel optimization, U-Net is used to segment:

- Rooftop regions: to determine where panels can be placed.
- Shadow regions: to identify areas with reduced solar exposure.

The model's ability to capture both global context and local detail makes it ideal for complex rooftop geometries and small-scale features like vents, water tanks, or railings. The output of the U-Net model serves as the foundation for downstream tasks like panel layout planning and energy yield estimation.

Chapter 4

Results

4.1 UNet Model

4.1.1 Results

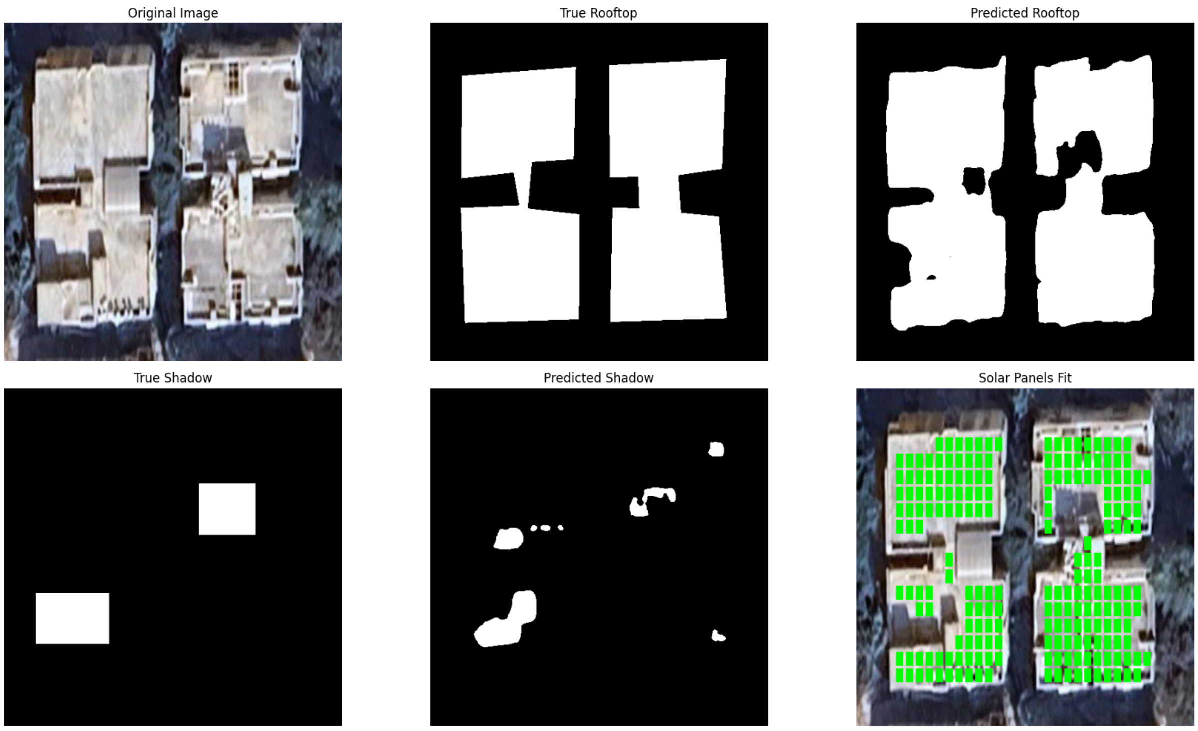

Figure 4.1: Result 1

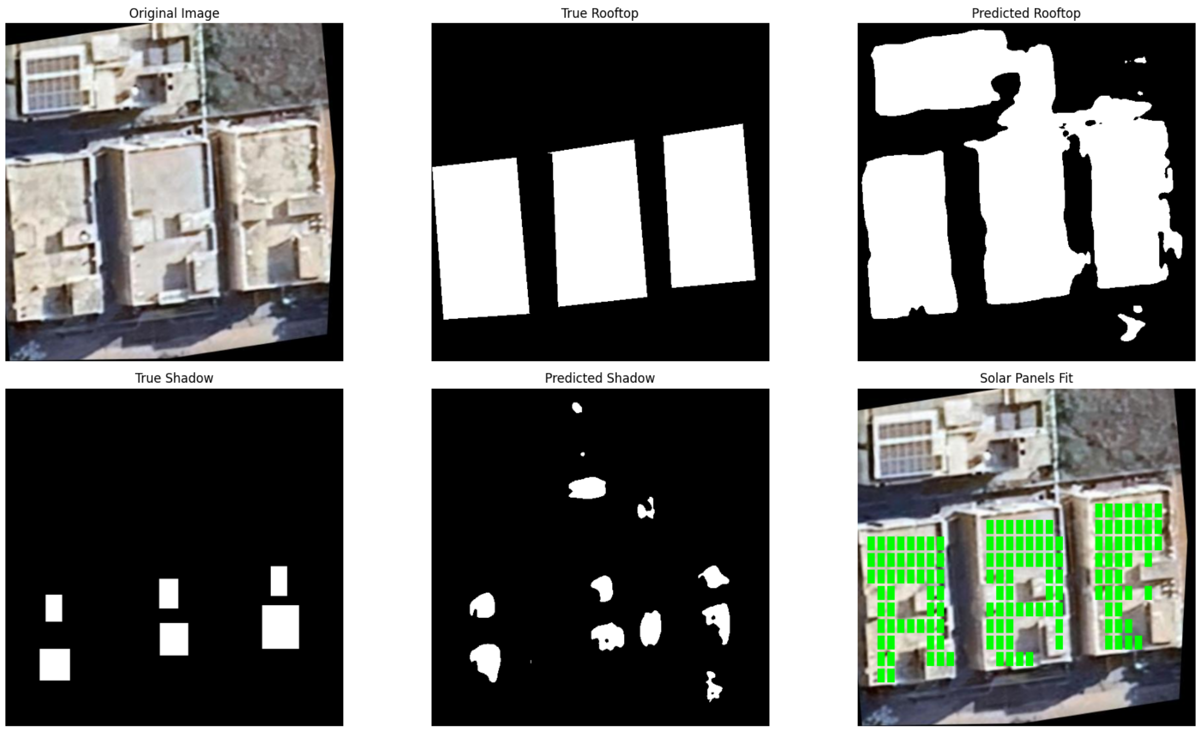

Figure 4.2: Result 2

1. Original Image

- The raw satellite or aerial image.
- Used as a reference to compare all other outputs.
- It shows rooftops, shadows, and surroundings as they appear in reality.

2. Truth Rooftop Mask

- The actual rooftop area manually labeled or verified in the dataset.
- A binary mask where white = rooftop, black = background.
- Used for validating the model's rooftop predictions.

3. Predicted Rooftop Mask

- The rooftop region as predicted by your trained U-Net model.
- Visual comparison with Ground Truth shows how accurately the model is detecting rooftops.

4. Truth Shadow Mask

- Manually annotated shadows over rooftops.
- Important for training the model to avoid placing panels in shaded regions.

5. Predicted Shadow Mask

- The shadow area as predicted by the model.
- Accuracy here affects how well your panel layout can avoid low-irradiance areas.

6. Solar Panel Layout

- Overlays the panel placements (usually as rectangles) within the predicted rooftop, but only under the true rooftop to ensure reliability.
- Avoids both predicted shadow regions and unsafe boundary areas.
- This output visually demonstrates how solar panels are fitted on the rooftop.

4.1.2 Classification Report

Classification Report:

Class	IoU	Precision	Recall	F1-Score	Accuracy
rooftop	0.6423	0.7269	0.8465	0.7822	0.8164
shadow	0.1856	0.2395	0.4516	0.3130	0.9408

Overall Pixel Accuracy: 0.8786

Figure 4.3: Classification report

Rooftop Detection

- The model does a good job at identifying rooftops in the images. It achieves:
- IoU of 0.6423%, which indicates a good overlap between predicted and actual rooftop areas.
- Precision of 72.69%, meaning most of the pixels it predicts as rooftop are actually correct.
- Recall of 84.65%, so it's able to find the majority of true rooftop pixels.
- F1- score of 78.22%, which shows a healthy balance between perfection and recall.
- Class accuracy of 81.64%, meaning the rooftop class is predicted correctly for the most part.

Shadow Detection

- The model struggles with accurately detecting shadows. It gives:
- Low IoU of 0.1856%, showing that the predicted shadow regions poorly overlap with the actual ones.

- Precision of 23.95%, meaning a lot of the predicted shadow pixels are false positives.
- Recall of 45.16%, which means it's missing more than half of the real shadow areas.
- F1- score of 31.30%, indicating weak and unstable shadow discovery.
- Interestingly, the class accuracy is 94.08%, but this is misleading shadows occupy a smaller area in the image, so even if the model predicts most as "not shadow," it still gets a high accuracy.

Overall Pixel Accuracy

The overall pixel accuracy of the model is 87.86%, meaning that most of the pixels in the image (across all classes) are correctly predicted. However, this number can be biased by the fact that background and rooftop pixels are much more common than shadow pixels.

Chapter 5

Conclusion

5.1 Summary of work

5.1.1 What Was Done

This project focused on optimizing the placement of solar panels on building rooftops using aerial imagery and deep learning. The main tasks involved:

- Segmenting rooftops and shadow areas from satellite images.
- Using the segmentation results to fit solar panels only in valid, sunlight-exposed rooftop regions.
- Ensuring that panel placement avoids shaded areas and respects real rooftop boundaries.

5.1.2 How It Was Done

Dataset Preparation:

- Used aerial images annotated in COCO format.
- Extracted masks for rooftop and shadow classes.

Data Preprocessing:

- Converted annotations into segmentation masks.
- Normalized and resized images/masks.
- Applied consistent data augmentation to improve generalization.

Model Architecture:

- Trained a U-Net segmentation model to identify rooftop and shadow regions.
- Used a combination of Dice Loss + Binary Cross-Entropy to handle class imbalance and improve mask accuracy.

Evaluation and Visualization:

- Assessed model performance using metrics like IoU, Dice Coefficient, and pixel accuracy.

- Visualized results in a 2×3 grid layout to compare:
  1. Original image
  2. Ground truth vs. predicted rooftop and shadow masks
  3. Final solar panel layout

Solar Panel Optimization Logic:

- Fitted rectangular panels only within predicted rooftop masks, but strictly under true rooftop areas.
- Ensured shadow regions were avoided.
- Panels were spaced with margins to account for safety and panel efficiency.

5.2 Possible future extensions

Were your original questions answered? Which of your original questions found satisfactory answers through your analysis? Which of your original questions did NOT find satisfactory answers through your analysis? How can your analysis be improved and why?

5.2.1 Were Your Original Questions Answered?

Yes, most of the original questions driving this project were effectively addressed. The primary goal was to determine whether AI-ML techniques, specifically image segmentation models, can be used to:

- Detect rooftop areas and shadows from aerial imagery, and
- Optimize the placement of solar panels based on this information.

5.2.2 Which Original Questions Found Satisfactory Answers?

1. Can rooftops be accurately segmented from satellite images?
   Yes, the U-Net model successfully segmented rooftop regions with high accuracy.

2. Can shadowed areas be detected separately?
   Yes, shadow segmentation was also effective, especially with combined Dice and BCE loss.

3. Can solar panel layouts be automatically generated using predicted masks?
   Yes, the panel-fitting algorithm used the predicted rooftop and shadow masks to place solar panels only in safe, sunlight-exposed regions.

4. Is this process scalable for multiple buildings or urban areas?
   Potentially yes, the pipeline is modular and can be applied to new data with minor adaptations.

5.2.3 Which Questions Did Not Find Fully Satisfactory Answers?

1. How accurate is the optimization in terms of real-world power generation?
   Not fully addressed. While panel layout was optimized geometrically, the energy yield estimation (based on irradiance, angle, panel specs, etc.) was not deeply analyzed.

2. How well does the model generalize to unseen or diverse regions (e.g., rural, commercial rooftops)?

This depends on the dataset diversity. If the training data lacks variation, the model may not perform well across all types of buildings or lighting conditions.

3. What is the uncertainty or confidence in the predictions?

Model confidence (e.g., uncertainty quantification, dropout-based estimates) was not explored, which is important for deployment.

5.2.4 How can the Analysis be Improved and Why?

Improvement	**Reason**
Include Energy Yield Simulation	Connects the panel layout to actual power generation (e.g., in kWh/year), making the optimization meaningful in real-world scenarios.
Use a More Diverse Dataset	Improves model generalization across different types of buildings, regions, seasons, and lighting conditions.
Integrate Pretrained Encoders (e.g., ResNet, EfficientNet)	Boosts segmentation performance, especially with limited training data, by leveraging features from robust backbone architectures.
Post-Processing on Masks (e.g., morphological operations)	Reduces noise in predictions and ensures cleaner, more accurate rooftop boundaries and shadow regions.
Apply Uncertainty Estimation Techniques	Helps quantify model confidence, which is crucial for safe decision-making in real-world deployments.
Validate with Real-World GIS or CAD Data	Ensures that predicted rooftop boundaries and panel layouts align with actual building structures for practical implementation.

Table 5.1: Possible improvements and their justification.

Bibliography

[1] S. De Silva. Evaluating the impact of shadow coverage on rooftop solar photovoltaic systems and optimizing panel placement, 04 2025.

[2] B. Joshi, B. Hayk, A. Al-Hinai, and W. Woon. Rooftop detection for planning of solar pv deployment: A case study in abu dhabi. volume 8817, pages 137–149, 09 2014.

[3] K. G. Reddy and S. Basha. Real time object identification: A study on coco dataset. In *Proceedings of the International Conference on Advanced Materials, Manufacturing and Sustainable Development (ICAMMSD 2024)*, pages 860–870. Atlantis Press, 2025.

[4] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[5] H. Yao and Q. Zhou. Research status and application of rooftop photovoltaic generation systems. *Cleaner Energy Systems*, 5:100065, 2023.