



**Middle East Technical University**  
**Informatics Institute**

# Rooftop Identification and Classification from High-Resolution Aerial Imagery for Photovoltaic Potential Analysis using Deep Learning

**Advisor Name: Prof. Dr. ALTAN KOÇYİĞİT**  
**(METU)**

**Student Name: Oğuz Kağan Ünal**  
**(Information Systems)**

**January 2026**

**TECHNICAL REPORT**

**METU/II-TR-2026-**



**Orta Doęu Teknik Üniversitesi**  
**Enformatik Enstitüsü**

# Fotovoltaik Potansiyel Analizi İçin Derin Öğrenme ile Yüksek Çözünürlüklü Hava Görüntülerinden Çatıların Tespiti ve Sınıflandırılması

**Danışman Adı: Prof. Dr. ALTAN KOÇYİĞİT**  
**(ODTÜ)**

**Öğrenci Adı: Oğuz Kağan Ünal**  
**(Bilişim Sistemleri)**

**Ocak 2026**

**TEKNİK RAPOR**

**ODTÜ/II-TR-2026-**

# REPORT DOCUMENTATION PAGE

<b>1. AGENCY USE ONLY (Internal Use)</b>	<b>2. REPORT DATE</b> 16.01.2026
<b>3. TITLE AND SUBTITLE</b>  ROOFTOP IDENTIFICATION AND CLASSIFICATION FROM HIGH-RESOLUTION AERIAL IMAGERY FOR PHOTOVOLTAIC POTENTIAL ANALYSIS USING DEEP LEARNING	
<b>4. AUTHOR (S)</b>  Oğuz Kağan Ünal	<b>5. REPORT NUMBER (Internal Use)</b>  METU/II-TR-2026-
<b>6. SPONSORING/ MONITORING AGENCY NAME(S) AND SIGNATURE(S)</b> Informatics Master's Programme, Department of Information Systems, Informatics Institute, METU Advisor: Altan Koçyiğit Signature:	
<b>7. SUPPLEMENTARY NOTES</b>	
<b>8. ABSTRACT (MAXIMUM 200 WORDS)</b>  Accurate identification of roof areas and roof types is crucial for assessing solar photovoltaic (PV) potential. This study proposes a two-stage deep learning method for roof analysis using high-resolution aerial RGB images. In the first stage, a semantic segmentation is applied and trained on the Roof Information Dataset 2 (RID2) dataset with a U-Net architecture using the ResNet34 encoder pre-trained on ImageNet. With this trained model, roof areas are detected. In the second stage, the roof areas obtained from the segmentation stage are classified into four different categories (flat, gable/hip, complex, and bugs) using an EfficientNetB0-based classifier. Between these two stages, since the roof polygons are pixel-based, morphological cleaning and contour extraction processes are applied before entering the classification stage, and the resulting polygons are simplified using the Ramer-Douglas-Peucker algorithm. In the RID2 test set, the mean IoU was 0.882 and Dice was 0.9367. Classification accuracy was 80.77% when tested with images on the test set. When the results of the pipeline established in the project are analyzed, it is concluded that the "bugs" class in the classification stage acts as a filter for false positives obtained from the segmentation stage.	
<b>9. SUBJECT TERMS</b>  Roof segmentation, Roof classification, Deep learning, Remote sensing, GIS, Rooftop solar PV potential	<b>10. NUMBER OF PAGES</b>  28

# TABLE OF CONTENTS

LIST OF TABLES.....	vi
LIST OF FIGURES .....	vii
LIST OF ABBREVIATIONS.....	viii
CHAPTER 1 .....	1
INTRODUCTION.....	1
1.1. Problem Statement.....	2
1.2. Objectives .....	2
1.3. Scope and Limitations.....	3
CHAPTER 2.....	5
BACKGROUND AND LITERATURE REVIEW.....	5
2.1. Semantic Segmentation Architectures .....	5
2.2. Transfer Learning in Computer Vision .....	6
2.3. Building Extraction and Rooftop Classification.....	7
2.4. Polygon Simplification Algorithms.....	7
CHAPTER 3 .....	9
METHODOLOGY.....	9
3.1. Datasets .....	10
3.2. Segmentation Model Design .....	12
3.3. Classification Model Design.....	12
3.4. Polygon Extraction and Simplification .....	13

<b>3.5. Evaluation Metrics</b> .....	14
<b>CHAPTER 4</b> .....	15
<b>IMPLEMENTATION</b> .....	15
<b>4.1. Development Environment</b> .....	15
<b>4.2. Data Preprocessing and Augmentation</b> .....	15
<b>4.3. Training Procedures</b> .....	16
<b>4.4. Web Application Deployment</b> .....	16
<b>CHAPTER 5</b> .....	18
<b>RESULTS AND DISCUSSION</b> .....	18
<b>5.1. Segmentation Model Performance</b> .....	18
<b>5.2. Classification Model Performance</b> .....	20
<b>5.3. Discussion</b> .....	22
<b>CHAPTER 6</b> .....	24
<b>CONCLUSION AND FUTURE WORK</b> .....	24
<b>REFERENCES</b> .....	25

## LIST OF TABLES

Table 1 - Characteristics of the RID2 dataset.....	10
Table 2 - Class distribution of the combined roof-type classification dataset.....	11
Table 3 - Segmentation performance of the U-Net + ResNet34 model on the RID2 test set. ....	19
Table 4 - Pixel-level evaluation metrics of the roof segmentation model on the RID2 test set. ....	20
Table 5 - Class-based classification performance on the test set.....	20

# LIST OF FIGURES

Figure 1 - Roof Types (Hristov et al., 2023) .....	2
Figure 2 - System architecture overview of the proposed two-stage pipeline, illustrating roof segmentation, post-processing and polygon extraction, roof-type classification, and final GIS-ready polygon outputs. ....	9
Figure 3 - Example roof instances manually labeled from high-resolution imagery in Mustafa Kemal Mahallesi (Ankara, Türkiye) to extend the roof-type classification dataset.....	11
Figure 4 - Representative roof-type classification results on test samples, comparing ground-truth labels with model predictions.....	16
Figure 5 - Prototype web application interface demonstrating the visualization of detected roof polygons and predicted roof-type classes layers on an interactive map. ....	17
Figure 6 - Qualitative segmentation results on RID2 test samples, including input images, ground-truth masks, predicted masks, and pixel-level error maps. ....	18
Figure 7 - Normalized pixel-level confusion matrix for the roof segmentation model, summarizing background vs. roof prediction performance on the RID2 test set. ....	19
Figure 8 - Normalized confusion matrix of roof-type classification results on the test set, showing class-level prediction performance and misclassification patterns. ....	21
Figure 9 - End-to-end pipeline integration example illustrating input satellite imagery, segmentation probability map, and the final binary roof mask used for polygon extraction. ....	23
Figure 10 - Final roof inventory visualization after filtering out polygons classified as “Bugs”, highlighting the role of the classification stage as an implicit noise-removal mechanism.....	23

# LIST OF ABBREVIATIONS

AIRS .....	Aerial Imagery for Roof Segmentation
CNN .....	Convolutional Neural Network
FCN .....	Fully Convolutional Network
FN .....	False Negative
FP .....	False Positive
GIS .....	Geographic Information System
GSD .....	Ground Sampling Distance
GPU .....	Graphics Processing Unit
HVAC .....	Heating, Ventilation, and Air Conditioning
IEA .....	International Energy Agency
IoU .....	Intersection over Union
PV .....	Photovoltaic
ReLU .....	Rectified Linear Unit
ResNet .....	Residual Network
RID2 .....	Roof Information Dataset 2
RGB .....	Red, Green, Blue
TP .....	True Positive
U-Net .....	U-shaped Network
ULMFiT .....	Universal Language Model Fine-tuning
VRAM .....	Video Random Access Memory

# CHAPTER 1

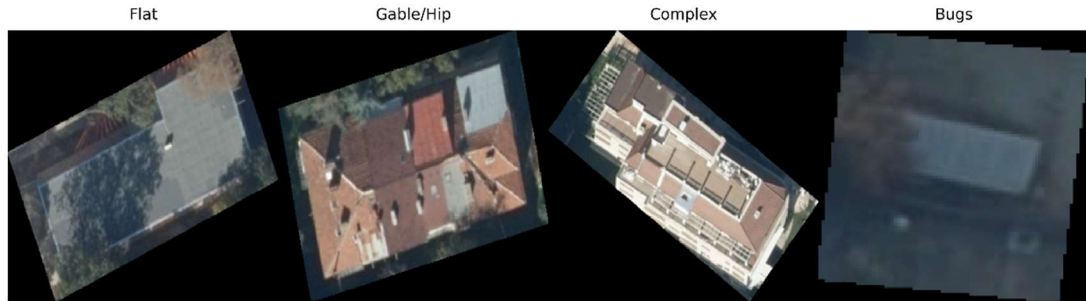
## INTRODUCTION

The global transition to renewable energy sources has accelerated significantly (IEA, 2021), driven by the emergence of the harms of climate change and the rapid increase in the energy production efficiency of renewable energy sources. Although the efficiency of photovoltaic (PV) systems is increasing, considering the land resources required for PV installation, utilizing existing building infrastructure to install these systems in urban areas, rather than on fertile farmlands or forests, offers a more environmentally friendly solution.

However, identifying and evaluating suitable roofs for solar energy installations, and convincing residential/commercial property owners about the payback period, complicates the adoption process for PV installations for individuals. Moreover, conventional methods necessitate labor-intensive and expensive manual evaluations, architectural plan examinations, and expert assessments that are challenging to scale in extensive urban environments. The widespread availability of high-resolution satellite and aerial imagery and advancements in deep learning for computer vision offer an opportunity to automate solar energy detection, evaluation, architectural plan analysis, cost, and amortization analysis processes. For example, Yu et al. (2018) developed a deep learning framework that analyzes high-resolution satellite imagery to identify solar PV installations across the United States.

Semantic segmentation, the process of assigning a class label to each pixel in an image, is a method frequently used to determine building roof boundaries. When combined with classification models capable of identifying roof types (flat, complex, gable/hip and bugs as shown in Figure 1), it becomes possible to create comprehensive roof inventories that can form the basis for solar energy potential assessments. This study

integrates these two deeplearning approaches into a process that supports the creation of an automated and scalable roof inventory and provides decision support inputs for photovoltaic planning and feasibility analysis.



*Figure 1 - Roof Types (Hristov et al., 2023)*

## **1.1. Problem Statement**

Although considerable progress in deep learning for remote sensing, certain challenges remain in roof analysis. Training segmentation models from scratch requires extensive computational resources. Transfer learning, which utilizes pre-trained models in relevant tasks, has been considered a viable solution to this problem. Segmentation models produce some false positives and imperfect boundary predictions. Furthermore, since pixel-based prediction is performed, careful post-processing is required to transform the obtained roof boundaries into more meaningful roof boundary vectors. This study addresses these challenges by proposing a two-stage pipeline where the classification model not only identifies roof types but also acts as a quality filter by identifying and separating segmentation artifacts, such as misclassified roads and trees, obtained during the segmentation phase.

## **1.2. Objectives**

The primary objectives of this project include, firstly, creating a highly robust roof segmentation model using a U-Net (Ronneberger et al., 2015) architecture trained on the Roof Information Dataset 2 (RID2) dataset (Krapf et al., 2025) and including a pre-trained ResNet34 (He et al., 2016) encoder. Secondly, developing an EfficientNetB0-based roof type classification model (Tan & Le, 2019) capable of identifying roof types independently of the first trained model. Thirdly, classifying the roofs obtained from

the segmentation stage using the classification model obtained in the second stage, and investigating the role of boundaries captured by the "Bugs" class in filtering segmentation noise. The fourth goal is to make the pixel-based roof polygons obtained in the segmentation stage usable in real-time Geographic Information Systems (GIS) applications and capable of generating simplified polygons using the Ramer-Douglas-Peucker (Ramer, 1972; Douglas & Peucker, 1973) algorithm, a polygon simplification technique. Finally, demonstrating the functionality of this pipeline through web and mobile prototypes developed for solar energy assessment scenarios.

### **1.3. Scope and Limitations**

The scope of this study is limited to distinguishing roof pixels from the background, focusing on binary roof segmentation of RGB aerial images. It also addresses four different roof-type categories relevant to solar installation suitability by utilizing a separate classification model. This study focuses on determining 'usable area (geometric availability)', which is the first and most critical step of PV potential analysis. Integration with radiation data is left for future work. Since the research was trained with 8 cm/pixel (0.08 m/pixel) data from the RID2 dataset, it is limited to images with a similar Ground Sampling Distance (GSD). The study currently does not account for rooftop obstructions such as chimneys, HVAC (Heating, Ventilation, and Air Conditioning) units, or skylights that affect the usable area for solar panel installation; further shading analysis from these structures and surrounding trees or buildings has not been applied in this study. The classification model was primarily trained on Turkish and European urban morphologies, and generalization to other regions requires further validation studies. In addition, integration with meteorological data, which is not included in the scope of this study, is required for solar radiation calculations.

The remainder of the report is organized as follows: Section 2 reviews the relevant background and literature on semantic segmentation, transfer learning, roof analysis, and polygon simplification. Section 3 presents the methodology, including datasets, model architectures, post-processing steps, and evaluation metrics. Section 4 details

the training environment, preprocessing, training strategy, and web deployment. Section 5 reports the experimental results and discusses the findings for both the segmentation and classification stages. Finally, Section 6 summarizes the work and provides directions for future studies.

# CHAPTER 2

## BACKGROUND AND LITERATURE REVIEW

The combination of deep learning and high-resolution aerial imagery has led to the frequent use of remote sensing systems for extracting geographic information from the Earth's surface. Convolutional Neural Networks (CNNs) have exhibited improved accuracy relative to conventional feature-based approaches in tasks such as land cover classification, object recognition, and segmentation (Zhu et al., 2017).

Ma *et al.* (2019) investigated deep learning methodologies in remote sensing image classification. They noted that earlier approaches relied on manually defined rules and features (e.g. heuristics for roof corners or color), whereas modern deep learning models automatically learn multilevel features (edges, corners, textures, color patterns, etc.) from data, eliminating the need for explicit rule-based feature engineering.

### 2.1. Semantic Segmentation Architectures

Semantic segmentation requires pixel-level classification. It therefore requires architectures that can capture semantic context at different scales while maintaining spatial resolution. To meet this need, Long et al. (2015) pioneered in this field by developing Fully Convolutional Networks (FCNs). In this work, fully connected layers could now be replaced by convolutional layers, allowing for the processing of inputs of different sizes and laying the foundation for encoder-decoder architectures.

U-Net, a symmetric encoder-decoder architecture with jump links that can combine feature maps in the encoding path with feature maps in the decoding path, was originally proposed by Ronneberger et al. (2015) and has been shown to help recover

detailed spatial information that is often lost in pooling layers. As a result, U-Net performs exceptionally well in segmentation tasks that demand precise edge definitions. Furthermore, U-Net is a widely used model in medical imaging and remote sensing due to its high performance even with limited training data. He et al. (2016) showed that residual connections enable the training of very deep networks by addressing the vanishing gradient problem, in this context ResNet variants, in particular ResNet34 and ResNet50, are widely used as encoders in segmentation architectures and have been shown to provide powerful feature extraction capabilities when pre-trained on ImageNet (Igloukov & Shvets, 2018). When the U-Net decoder and ResNet encoder are used together, it has been shown to show successful results in segmentation from aerial photographs.

## **2.2. Transfer Learning in Computer Vision**

Transfer learning has become a crucial foundation for computer vision because even with limited data, we can achieve desired results by using previously trained models and manipulating the desired layers of the model. The basic principle of this approach is the ability to usefully transfer visual details, such as edges and textures learned from a previous task, to related tasks (Yosinski et al., 2014).

Wurm et al. (2019) established that employing transfer learning in semantic segmentation models significantly improved the precision in identifying unplanned settlement zones.

The Universal Language Model Fine-tuning (ULMFiT), developed for natural language processing but later yielding successful results in satellite image processing, was introduced by Howard and Ruder (2018). The principle of this technique is the gradual unfreezing of layers during fine-tuning. Layers are gradually unfrozen from the classifier head backward, i.e., toward the first convolutional layers. This prevents previously learned features from experiencing catastrophic forgetting.

Tan and Le (2019) presented the EfficientNet model family, designed through compound scaling of depth, width, and resolution, and neural architecture search. The base model of this series, EfficientNetB0, achieved a notable accuracy rate with a

comparatively modest parameter count of approximately 4.4 million. This also makes the model a suitable option in situations where computational resources are limited.

### **2.3. Building Extraction and Rooftop Classification**

To compare roof segmentation techniques, Chen et al. (2020) labeled and published the AIRS (Aerial Imagery for Roof Segmentation) dataset, which consists of high-resolution orthophotos with building footprint masks. The RID2 dataset provides a total of 4,764 images at 512×512 resolution with multi-class roof segmentation labels that distinguish five different roof orientations and backgrounds (Krapf et al., 2025). In addition to binary building detection, this dataset also contains roof orientation information related to solar energy applications, making it an efficient source for solar angle calculation models.

Hristov et al. (2023)<sup>2</sup> published a dataset of 3,617 GeoTIFF images with a 10 cm/pixel resolution from Sofia, Bulgaria, which is suitable for roof detection and classification into four categories: flat, gable/curved, complex, and bug. The advantageous part of this dataset for our project is the “Bug” class. Images in the "Bug" class include construction sites, unclear images, and non-roof elements. This enables the creation of a system to manage ambiguous detections in the classification process.

Yu et al. (2018) developed a deep learning framework for the detection of existing solar panels in the United States. This work focused on creating an inventory of existing panels rather than assessing roof suitability. It demonstrated the potential of deep learning in comprehensive solar energy assessment.

### **2.4. Polygon Simplification Algorithms**

Converting the pixel-level masks obtained from the segmentation stage into vector polygons requires the use of contour extraction and simplification algorithms. The goal here is to reduce the number of points by eliminating redundant corner points while preserving the essential geometric shape. To achieve this goal, the Ramer-Douglas-Peucker algorithm, developed independently by Ramer (1972) and Douglas and Peucker (1973), is one of the commonly used methods. In this algorithm, when a curve is given, a straight line is drawn between the start and end points, and the point at the maximum perpendicular distance from this line is identified. If this point

exceeds the specified  $\varepsilon$  (epsilon) value, which represents the maximum allowable distance deviation, the algorithm continues to recursively process the two resulting sub-curves. Through this procedure, all points under the specified  $\varepsilon$  are eliminated.

# CHAPTER 3

## METHODOLOGY

In the first stage of the two-stage pipeline established to perform roof analysis, semantic segmentation is applied using the U-Net model based on the ResNet34 encoder, and a binary roof mask is generated for each pixel as the output of this model, representing the roof and background. In the second stage, a classifier based on EfficientNetB0 is used to classify the detected roofs (candidate regions extracted during the segmentation phase) into four main categories. These intermediate processing steps are detailed in Chapter 3, Section 3.5. Once these two stages are complete, the polygons obtained using the Ramer-Douglas-Peucker algorithm undergo a simplification process. Since both models are developed independently, this allows each model to be optimized for its specific task and provides an opportunity to filter errors obtained from the segmentation stage during the classification stage. (Figure 2)

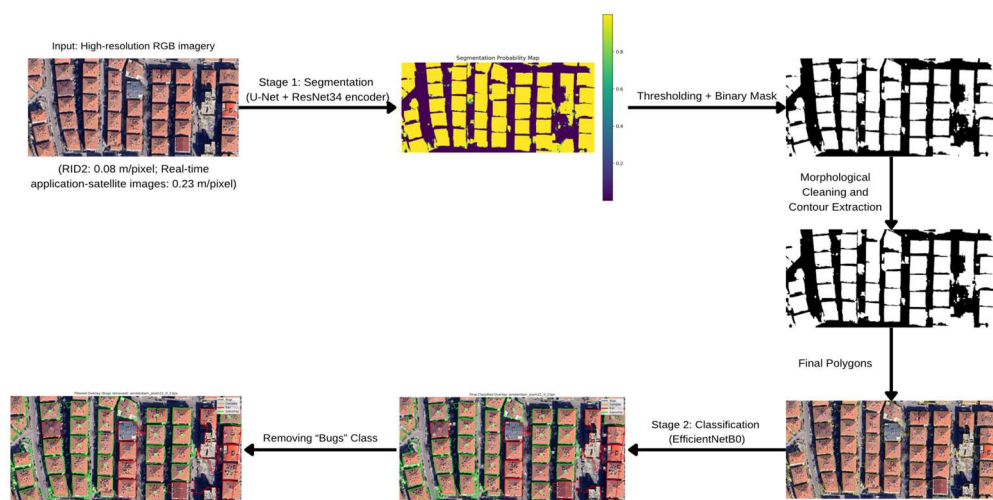


Figure 2 - System architecture overview of the proposed two-stage pipeline, illustrating roof segmentation, post-processing and polygon extraction, roof-type classification, and final GIS-ready polygon outputs.

### 3.1. Datasets

#### 3.1.1. Segmentation Dataset

The dataset used to train the segmentation model was RID2. Within this dataset, five different roof orientation categories were labeled from 0 to 4, and the index for non-roof areas was labeled as 5. In the first stage, since we were only performing segmentation, the data indexed between 0 and 4 was considered as a single class and converted to a binary format. This conversion was done to treat all roof types uniformly as “roof” during segmentation, focusing the model on distinguishing roofs from non-roof areas. The dataset was divided into training (80%), validation (10%), and test (10%) subsets using random sampling, as detailed in Table 1.

Table 1 - Characteristics of the RID2 dataset.

Total Images	4,764
Image Dimensions	512 x 512-pixels
GSD	0.08 m/pixel
Annotation Classes	Roof Segments/Orientations
Training/Validation/Test Split	80% / 10% / 10%

#### 3.1.2. Classification Dataset

In the classification dataset, different sources were combined to capture different roof types and geographical regions with the aim of capturing variability, particularly in Türkiye. The first data source is the dataset obtained by Hristov et al. (2023) from Sofia, Bulgaria, which contains a total of 3,617 GeoTIFF images divided into four classes - flat (956 images), hipped/ridge (1247 images), complex (612 images) and bugs (802 images). In order to include roof types specific to Türkiye in this ready-made dataset, high-resolution satellite images from Mustafa Kemal Mahallesi, Ankara, were also manually labeled (Figure 3) as part of this study and these two datasets were combined and used to train the classification model. The final class distribution of the combined dataset is summarized in Table 2.

Table 2 - Class distribution of the combined roof-type classification dataset.

Class	Sofia (Bulgaria)	Ankara (Türkiye)	Total
Flat	956	41	997
Gable/Hip	1,247	121	1,368
Complex	612	32	644
Bugs	802	15	817
Total	3,617	209	3826



Figure 3 - Example roof instances manually labeled from high-resolution imagery in Mustafa Kemal Mahallesi (Ankara, Türkiye) to extend the roof-type classification dataset.

### 3.1.3. Data Preprocessing

In the segmentation model preprocessing stage, 512\*512-pixel images from the RID2 dataset were used at their native resolution and a normalization process specific to the ResNet34 architecture was applied using the relevant function of the “segmentation\_models” library, which adjusts pixel values according to ImageNet statistics. For the classification model, RGB channels were extracted from the TIFF format images in the dataset of Hristov et al. (2023) and converted to PNG format. For the pipeline, the roof sections obtained from the segmentation phase were resized to a 224\*224-pixel resolution as expected by the EfficientNetB0 model.

### **3.2. Segmentation Model Design**

The segmentation model developed in this study combines the encoder-decoder structure of U-Net with the ResNet34 backbone, which is pre-trained on ImageNet. This combined ResNet34 encoder consists of 34 layers organized in residual blocks with skip-connections that facilitate gradient flow during training. The encoder transfers the feature maps produced at different scales, such as 64, 128, 256, and 512, to the decoder via skip connections.

The decoder uses transposed convolutions (2×2 kernel, 2-step interval) for upsampling. In this process, high-level semantic information from deeper layers is blended with fine spatial details from previous layers, gradually recovering spatial resolution. In the decoder pathway (the expansive path), each block processes the upsampled feature maps through two 3×3 convolution layers. These are followed by ReLU activation functions, which introduce non-linearity, enabling the model to learn complex morphological shapes. In the final output layer, the depth dimension is reduced to a single channel and we get a single image. According to the results, the sigmoid function compresses between 0 and 1. If it is close to 0, it is considered as background and if it is close to 1, it is considered as roof. So the result does not directly tell us whether it is a roof or not, it gives us a probability map.

### **3.3. Classification Model Design**

The classification model employs the EfficientNetB0 architecture as the primary feature extractor, with a custom classification header added at the end of the network. With approximately 4.4 million parameters, which is less than ResNet50's 25.6 million parameters, this model has been shown to achieve comparable performance and high accuracy on ImageNet (Tan & Le, 2019). EfficientNetB0 employs compound scaling to address issues related to network depth, width, and input dimensions.

The classification head designed specifically for this project consists of a Global Average Pooling layer that reduces spatial dimensions, a Batch Normalization layer to increase training stability, a Dense layer with 256 units and ReLU activation, a Dropout layer with a rate of 0.4 to prevent overfitting, and finally a final output layer with 4 units and softmax activation that acts as the output layer.

The classification model was trained in two stages: first, the EfficientNetB0 base was frozen and only the special classification head added to the model was trained for 20 epochs with a learning rate of 0.001 using Adam optimizer, which causes the randomly initialized head to learn appropriate combinations without disturbing the pre-trained weights. Once this phase is successfully completed, the previous layers are left frozen to perform fine-tuning and the last 40 layers of EfficientNetB0 are unfrozen. Value reduced to (1e-5) to alleviate the catastrophic forgetting problem. It was then continued for another 20 epochs. An early stopping mechanism was implemented to monitor validation performance, and the model checkpoint yielding the optimal metric was retained.

### **3.4. Polygon Extraction and Simplification**

#### **3.4.1. Post-Processing Pipeline**

The segmentation mask obtained from the first model is subjected to post-processing steps before polygon extraction, the first of which is to close the small gaps in the detected regions with an area threshold of 500 pixels. In the second step, Gaussian smoothing ( $\sigma = 1.5$ ) is used to reduce the distortions caused by pixelization at the region boundaries. The last step is the extraction of the contours that define the region boundaries, where the Marching Squares algorithm is applied to the probability map to generate vector contours at a threshold of 0.5.

#### **3.4.2. Ramer-Douglas-Peucker Algorithm**

Although the contours extracted by post-processing produce smoother roof boundaries, the polygons are simplified using the Ramer-Douglas-Peucker algorithm to perform vertex reduction. A tolerance value of  $\epsilon = 1.0$  to 2.0 pixels for the roof polygons was found to simplify enough without affecting our precise area calculations. These epsilon values correspond to a ground distance of approximately 8-16 cm in images with a Ground Sampling Distance of 8 cm. At the end of the post-processing, the simplified polygons are converted into GeoJSON format suitable for GIS applications.

### 3.5. Evaluation Metrics

The segmentation phase is based on the following metrics:

$$IoU \text{ (Intersection over Union)} = \frac{TP}{TP + FP + FN}$$

$$Dice = \frac{2 * TP}{2 * TP + FP + FN}$$

True Positive (TP), False Positive (FP), and False Negative (FN) refer to the number of true positive, false positive, and false negative pixels, respectively. IoU directly measures the overlap between predicted and ground truth masks, making it the standard metric for segmentation evaluation.

For classification performance, accuracy, precision, recall, and F-1 score were used. For detailed class-specific performance metrics, a confusion matrix was used.

# CHAPTER 4

## IMPLEMENTATION

This chapter presents the implementation details of the proposed pipeline, including the training setup, preprocessing workflow, and deployment architecture used to deliver real-time roof detection and classification results.

### 4.1. Development Environment

The training of the models was carried out in the Google Colab environment using an NVIDIA A100 GPU (80 GB VRAM). In the development process, TensorFlow was used for model training, the segmentation-models library for the U-Net architecture with pre-trained encoders, OpenCV and scikit-image tools for image processing, and finally the Shapely library for geometric operations on the extracted polygons.

### 4.2. Data Preprocessing and Augmentation

A data generator was developed to process the dataset efficiently. With this generator, images and masks are not stored in bulk, but are instead loaded instantaneously and preprocessed in accordance with the ResNet34 architecture and multi-class masks are converted into binary masks. In some images, in order to alleviate the class imbalance, image fragments with a roof pixel ratio below 1% were extracted from the training set, but these extraction steps were not applied in the validation set. In addition, random horizontal and vertical rotations, 90-degree rotations, brightness ( $\pm 10\%$ ) and contrast adjustments (in the range of 0.9-1.1) were applied as a data augmentation strategy within the scope of segmentation. This data augmentation method was applied to both images and masks. For the data prepared for the classification model, random rotation (horizontal and vertical), random rotation ( $\pm 15\%$ ), random zoom ( $\pm 10\%$ ), and random contrast (0.2 range) techniques were applied to the data during training.

## 4.3. Training Procedures

### 4.3.1. Segmentation Model Training

The training of the segmentation model was configured to take a maximum of 50 epochs with the Adam optimization algorithm, a learning rate of  $1e-4$ , a batch size of 16, and  $512 \times 512$ -pixel chunks, and an early stopping mechanism. If the validation loss value did not show any improvement for 5 epochs during training, the learning rate was reduced by a factor of 0.5 and a minimum limit of  $1e-7$  was given. If there was no improvement in the validation IoU score for 10 epochs, the training process was stopped.

### 4.3.2. Classification Model Training

The classification model was trained using a two-stage training strategy as detailed in Chapter 3, Section 3.4. In the first stage, only the classification head was trained for 20 epochs with a learning rate of 0.001. In the second stage, the last 40 layers of EfficientNetB0 were fine-tuned for another 20 epochs with a learning rate of  $1e-5$ . A batch size of 32 was preferred in both stages. Example classification outputs (ground-truth vs. predicted labels) on test samples are shown in Figure 4.



*Figure 4 - Representative roof-type classification results on test samples, comparing ground-truth labels with model predictions.*

## 4.4. Web Application Deployment

To illustrate the applicability of the constructed pipeline, a web application has been established at [solareka.enerjjeika.com](http://solareka.enerjjeika.com). The system architecture comprises a frontend and a backend that offers a mapping interface, along with a Python-based microservice for roof identification utilizing deep learning models.

The segmentation and classification models are delivered via a specialized microservice that processes satellite image tiles obtained from mapping APIs. Upon a user's selection of a location on the map, the system retrieves the relevant high-

resolution satellite imagery, processes it via the segmentation model to identify roof boundaries, extracts distinct roof polygons, classifies each polygon utilizing the classification model, and presents the outcomes as GeoJSON layers rendered on the interactive map. Users can observe recognized roofs, color-coded by classification category (Flat, Gable/Hip, Complex, or Bugs), and obtain specific information for each identified roof polygon (Figure 5).



*Figure 5 - Prototype web application interface demonstrating the visualization of detected roof polygons and predicted roof-type classes layers on an interactive map.*

# CHAPTER 5

## RESULTS AND DISCUSSION

This chapter presents the quantitative and qualitative evaluation results of the proposed two-stage roof analysis pipeline. The performance of the segmentation model is first assessed using standard pixel-level metrics and visual outputs, followed by the evaluation of the roof type classification model using class-based performance scores and confusion matrix analysis.

### 5.1. Segmentation Model Performance

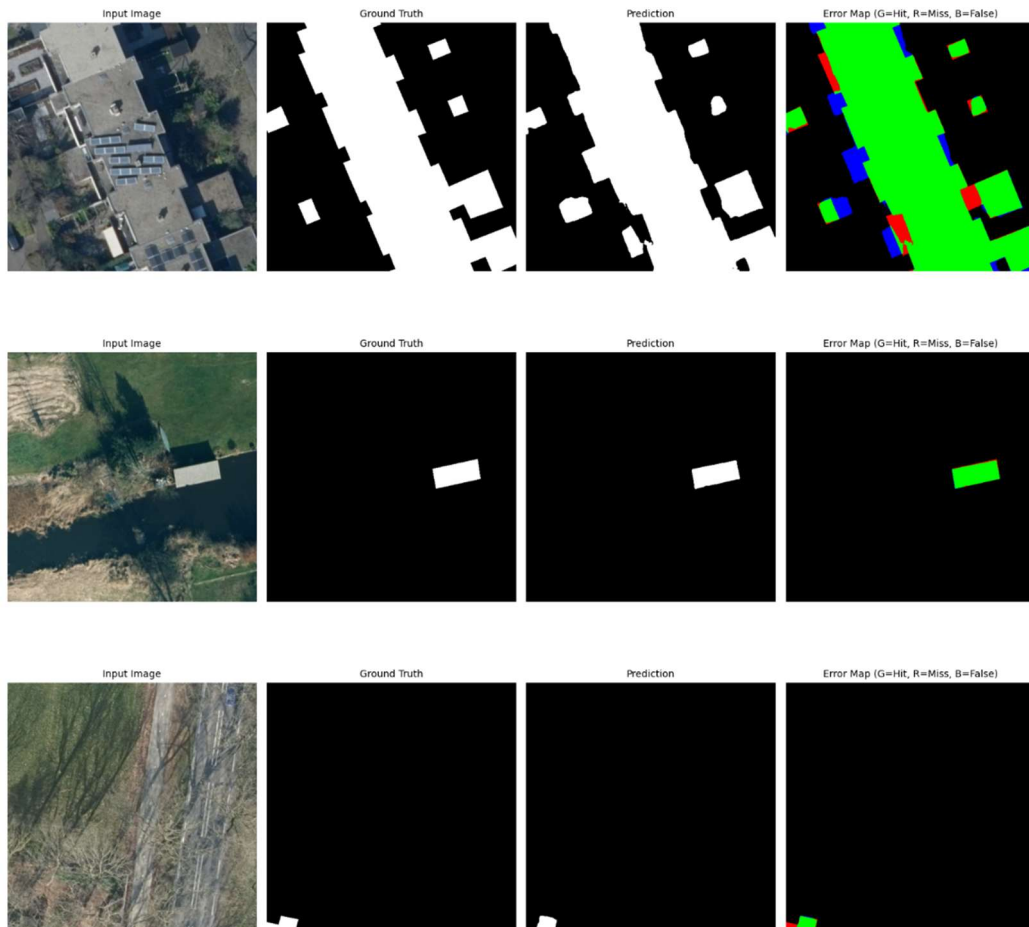


Figure 6 - Qualitative segmentation results on RID2 test samples, including input images, ground-truth masks, predicted masks, and pixel-level error maps. Color coding: Green = True Positive, Blue = False Positive, Red = False Negative, and Black = True Negative.

The U-Net + ResNet34 model was evaluated on 477 test images from the RID2 dataset, and the model yielded high segmentation performance with an average IoU of 0.8819 and a Dice coefficient of 0.9367 (Table 3). Rapid improvement was observed in the first 15 epochs of the training period, followed by gradual improvement. The fact that the difference between the training and validation metrics remained below 5% also indicates that the model's overfitting risk was minimal.

Table 3 - Segmentation performance of the U-Net + ResNet34 model on the RID2 test set.

Metric	Value
Mean IoU	0.8819
Dice Coefficient	0.9367
Test Loss	0.2344

Figure 6 presents qualitative examples of segmentation outputs, including input images, ground truth masks, predictions, and error maps. Additionally, pixel-level evaluation metrics are reported in Table 4.

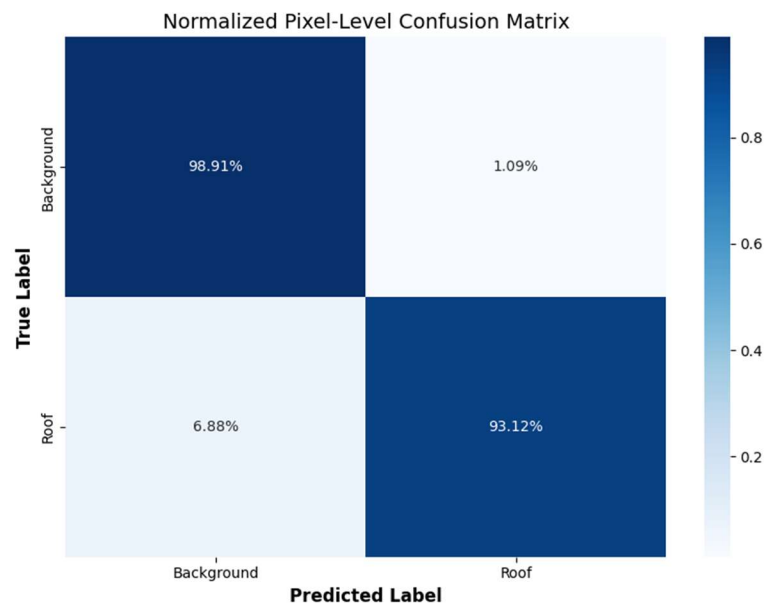


Figure 7 - Normalized pixel-level confusion matrix for the roof segmentation model, summarizing background vs. roof prediction performance on the RID2 test set.

Visual inspections performed on the test set showed that the model performed successfully in houses with clearly defined boundaries, and the model could accurately identify roof boundaries even with varying colors and materials (Figure 6).

It was observed that data resembling flat roof types, such as parking lots and roads, were incorrectly identified as roofs. Errors also occurred in identifying roofs that shared color and texture with the surrounding vegetation. Finally, errors were also seen in identifying the boundaries of roofs with irregular geometries, as clear boundaries could not be drawn (Figure 7).

Table 4 - Pixel-level evaluation metrics of the roof segmentation model on the RID2 test set.

Pixel-Level Metric	Value
Pixel Accuracy	0.9785
Precision	0.9502
Recall	0.9312
Specificity	0.9891
F1 Score	0.9406

## 5.2. Classification Model Performance

The classification model was developed via a two-phase transfer learning approach. In Phase 1, just the custom classification head was trained, with the EfficientNetB0 base remained frozen, resulting in a validation accuracy of 80.91%. During Phase 2, the final 40 layers of the basic model were unfrozen and fine-tuned using a decreased learning rate ( $1e-5$ ) to modify the pretrained features for the roof classification domain. The model was assessed using a test set of 723 images distributed across classes, achieving an overall test accuracy of 80.77% (Table 5).

Table 5 - Class-based classification performance on the test set.

Class	Precision	Recall	F1-score	Total Image
Bugs	0.7857	0.5946	0.6769	74
Complex	0.7705	0.6667	0.7148	141
Flat	0.7273	0.7907	0.7577	172
Gable/Hip	0.8659	0.9226	0.8934	336
<b>Weighted Avg</b>	0.8061	0.8077	0.8041	723

As shown in Table 5 and Figure 8, Gable/Hip roofs performed better with 92.26% recall and 86.59% accuracy due to their unique geometric features such as sloping surfaces, ridge lines and sequential tile patterns. Flat roofs, on the other hand, showed a balanced performance with 79.07% recall and 72.73% accuracy. Although the smooth textures of flat roofs were identifiable, they were often confused with other flat surfaces, such as parking lots or terraces. Complex roofs showed 66.67% recall and 77.05% precision. The reduced recall suggested that many complex structures were misclassified as Gable/Hip or flat roofs. The category “Bugs” showed 59.46% recall and 78.57% precision. The high precision indicates that when the model identifies an instance as a “Bug”, it is most likely a correct identification. This is a desirable result for our two-stage pipeline.

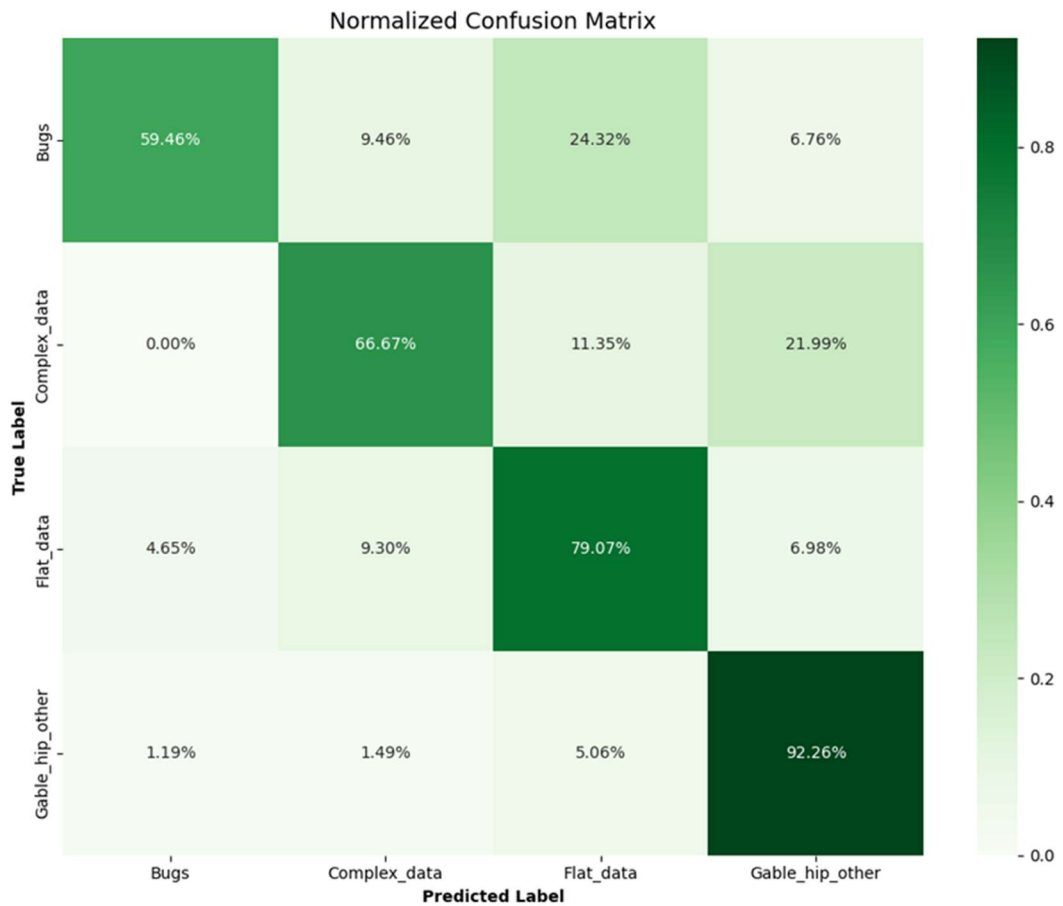


Figure 8 - Normalized confusion matrix of roof-type classification results on the test set, showing class-level prediction performance and misclassification patterns.

### 5.3. Discussion

One of the main findings of this study is that the classification stage acts as a filter against the errors obtained in the segmentation stage. When the segmentation model produces false positive predictions, the classification step often assigns these elements to the “Bugs” category. When we filter out the predictions classified as “Bugs” from the final output, we can observe that the pipeline removes most of the noise from the segmentation. This ensures that only valid roof polygons are displayed on the map. In order to analyze the pipeline with real data, a comprehensive investigation was conducted using high-resolution satellite imagery from Ankara. The outputs were manually inspected and labeled (Figure 9). The test results showed that out of the 1420 polygons identified, 1008 (70.99%) were classified as “Bugs”, while 412 were recognized as real roofs. Analysis of the images classified as “Bugs” revealed that 879 images were roads, roof components, parking lots, and densely vegetated areas. The remaining 129 boundaries were actually roofs, but some were complex roofs and others were roofs adjacent to roads. Figure 10 shows the final state of a pipeline after removing bugs. Some roofs were detected as flat despite being gable/hip. The main reason for these performance degradations was the use of satellite imagery instead of aerial imagery. Since there was no source available to broadcast dynamic aerial imagery in a real-time application, the pixel-per-resolution of the satellite images was optimized, and tests were completed accordingly.

Training the classification model with data from different geographical regions improved performance. Fusing images from Ankara with images from Sofia enriched the dataset used in Hristov et al. (2023) for training roof image and roof material recognition. Visual analyses revealed that the model trained solely with Bulgarian data performed poorly in detecting roof boundaries in test images from Türkiye.

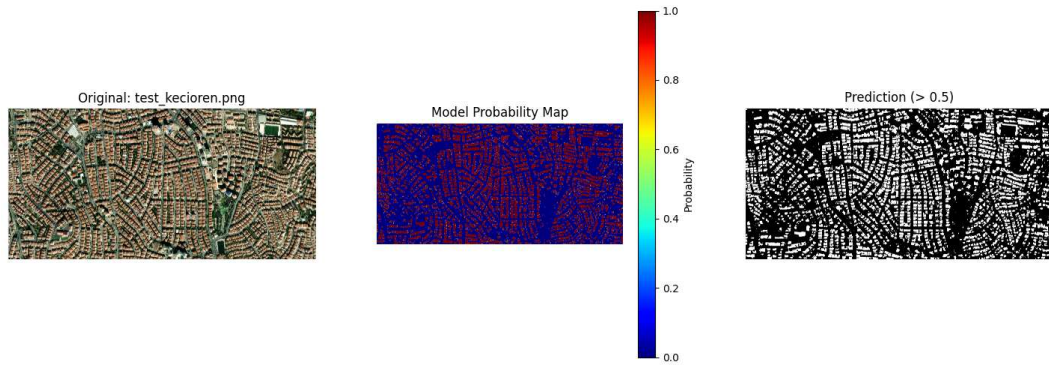


Figure 9 - End-to-end pipeline integration example illustrating input satellite imagery, segmentation probability map, and the final binary roof mask used for polygon extraction.

Since the segmentation model is trained solely on the RID2 dataset, it may yield erroneous results when tested with real-world satellite imagery. While the performance of the "Bugs" class in the classification model is acceptable, its heterogeneous structure means that further subdividing this class into subcategories such as construction, roads, and trees could facilitate the interpretation of the results.



Figure 10 - Final roof inventory visualization after filtering out polygons classified as "Bugs", highlighting the role of the classification stage as an implicit noise-removal mechanism.

# CHAPTER 6

## CONCLUSION AND FUTURE WORK

Within the scope of this study, a two-stage roof analysis pipeline has been established that integrates semantic segmentation and roof type classification for photovoltaic potential analysis. First, roof boundaries are detected from high-resolution images, and then they are categorized according to roof type classes to support scalable and automated roof inventory creation.

This study observed that segmentation errors are captured by the “Bugs” class due to the classification stage acting as an implicit noise filter. Polygon simplification processes provide a vector output suitable for GIS applications by reducing the size of outputs obtained in GeoJSON format by 70% while maintaining area calculation accuracy.

Future studies may include identifying roof structures that affect the usable area for solar panel installation. Identifying elements such as chimneys, HVAC units, ventilation systems, skylights, or antennas that significantly reduce the usable area of roofs and create shadows will enable more accurate panel placement calculations. In their recent study, Li et al. (2024) showed that such obstacles can be identified from high-resolution images and panel placement can be calculated based on roof orientations. Furthermore, the layout of the roof lines is another factor affecting panel placement. Identifying these breaks will also help to determine the geometric boundaries where the panels should be placed. The prototype web application deployed at [solareka.enerjeka.com](http://solareka.enerjeka.com) demonstrates the real-time functionality of the system and serves as a foundation for these future enhancements.

## REFERENCES

Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., ... Zhou, Y. (2021). Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306.

Chen, Q., Wang, L., Wu, Y., Wu, G., Guo, Z., & Waslander, S. (2018). Aerial imagery for roof segmentation: A large-scale dataset towards automatic mapping of buildings. <https://doi.org/10.48550/arXiv.1807.09532>

Douglas, D.H. and Peucker, T.K. (1973) Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature. *The Canadian Cartographer*, 10, 112-122. <http://dx.doi.org/10.3138/FM57-6770-U75U-7727>

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770–778). <https://doi.org/10.1109/CVPR.2016.90>

Hristov, E., Petrova-Antonova, D., Petrov, A., Borukova, M., & Shirinyan, E. (2023). Imagery dataset for rooftop detection and classification [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.7633595> (Last accessed: 15 Jan 2026).

IEA. (2021). Net Zero by 2050. IEA, Paris. <https://www.iea.org/reports/net-zero-by-2050> (Last accessed: 15 Jan 2026). Licence: CC BY 4.0.

Jochem, A., Höfle, B., Rutzinger, M., & Pfeifer, N. (2009). Automatic roof plane detection and analysis in airborne lidar point clouds for solar potential assessment. *Sensors*, 9(7), 5241–5262. <https://doi.org/10.3390/s90705241>

Krapf, S., Ganß, M., Zinniel, G., & Lienkamp, M. (2025). RID2 – Roof Information Dataset for Identifying Roof Segments and Roof Superstructures in Aerial Images (1.0) [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.14062580> (Last accessed: 15 Jan 2026).

Li, Q., Krapf, S., Mou, L., Shi, Y., & Zhu, X. (2024). Deep learning-based framework for city-scale rooftop solar potential estimation by considering roof superstructures. *Applied Energy*, 374, 123839. <https://doi.org/10.1016/j.apenergy.2024.123839>

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 3431–3440). <https://doi.org/10.1109/CVPR.2015.7298965>

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., & Johnson, B. (2019). Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152, 166–177. <https://doi.org/10.1016/j.isprsjprs.2019.04.015>

Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1, 244–256.

Ren, H., Sun, Y., & Zhang, Y. (2023). A novel 3D-geographic information system and deep learning integrated approach for high-accuracy building rooftop solar energy potential characterization of high-density cities.

<https://doi.org/10.26868/25222708.2023.1735>

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In LNCS (Vol. 9351, pp. 234–241).

[https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv, abs/1905.11946.

Wurm, M., Stark, T., Zhu, X., Weigand, M., & Taubenböck, H. (2019). Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks. ISPRS Journal of Photogrammetry and Remote Sensing, 150, 59–69. <https://doi.org/10.1016/j.isprsjprs.2019.02.006>

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? arXiv, abs/1411.1792.

Yu, J., Wang, Z., Majumdar, A., & Rajagopal, R. (2018). DeepSolar: A machine learning framework to efficiently construct a solar deployment database in the United States. Joule.

Zhong, T., Zhang, Z., Chen, M., Zhang, K., Zhou, Z., Zhu, R., Wang, Z., Lü, G., & Yan, J. (2021). A city-scale estimation of rooftop solar photovoltaic potential based on deep learning. *Applied Energy*, 298, 117132. <https://doi.org/10.1016/j.apenergy.2021.117132>

Zhu, X. X., et al. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4), 8–36. <https://doi.org/10.1109/MGRS.2017.2762307>