

# 1 Mapping Solar Photovoltaic Systems Using U-Net

2 **L. Erez, Prof. T. Svoray, R. Bluestein**

3 <sup>1</sup>Department of Geoinformatics, Environmental Science, and Urban Planning, Ben-Gurion University of  
4 the Negev, Israel

## 5 **Key Points:**

- 6 • Employed a U-Net model to map PV systems in Bedouin settlements shows chal-  
7 lenges due to data complexity, imbalance, and memory limitations.
- 8 • Automated mapping of photovoltaic systems can significantly contribute to renew-  
9 able energy policy and infrastructure in underserved regions

10 **Abstract**

11 The transition to renewable energy sources, such as solar photovoltaic systems, is cru-  
 12 cial for reducing reliance on fossil fuels and mitigating climate change. This study fo-  
 13 cuses on mapping PV systems in off-grid Bedouin settlements in the Negev using the U-  
 14 Net deep learning architecture. The U-Net model was trained and evaluated, achieving  
 15 training and validation losses of 0.442 and 0.457, respectively. However, the model's pre-  
 16 cision (0.337), recall (0.201), and Intersection over Union (IoU) (0.144) on the testing  
 17 set were sub-optimal, indicating challenges in accurately distinguishing PV systems from  
 18 similar objects. These results highlight a significant disparity in performance compared  
 19 to previous studies using U-Net architectures, attributed to the unique challenges of our  
 20 dataset, which includes PV systems placed both on rooftops and on the ground. Addi-  
 21 tionally, Limited computational resources and data imbalance further constrained model  
 22 performance. This research underscores the importance of accurate PV system mapping  
 23 for renewable energy policy and infrastructure development, providing a valuable frame-  
 24 work for automating PV detection in challenging environments and supporting the tran-  
 25 sition to sustainable energy sources in off-grid communities.

26 **1 Introduction**

27 The transition to renewable energy sources is a critical step in reducing dependence  
 28 on fossil fuels and mitigating the adverse effects of climate change. Among the various  
 29 renewable technologies, solar photovoltaic (PV) systems have gained significant traction  
 30 due to their ability to harness solar energy, a clean and inexhaustible resource. Solar PV  
 31 systems offer numerous advantages, including the reduction of greenhouse gas emissions,  
 32 diversification of the energy supply, and potential economic benefits through job creation  
 33 in the solar industry. These systems enable consumers to generate their own electricity,  
 34 reduce their utility bills, and contribute to a more sustainable energy grid. The increas-  
 35 ing popularity of solar PV systems is evident in the substantial rise of distributed so-  
 36 lar PV installations, particularly on rooftops of homes and businesses (Camilo et al., 2018;  
 37 Imamoglu et al., 2017). However, the adoption and integration of these systems vary across  
 38 different countries and communities, resulting in disparities in the benefits and challenges  
 39 associated with this technology.

40 Despite these numerous advantages, the widespread adoption of solar PV systems  
 41 brings about significant challenges in accurately tracking their growth and integration  
 42 into the energy grid. This challenge makes it difficult to plan for future expansion and  
 43 to understand the contribution of solar PV to the energy grid (de Hoog et al., 2020). Fur-  
 44 thermore, existing research mainly focuses on the adoption of PV in Western popula-  
 45 tions rather than in off-grid areas. Understanding PV adoption in these off-grid areas,  
 46 however, is crucial for effective policy development.

47 For example, Diallo and Moussa (2020) showed that in Côte d'Ivoire, the use of so-  
 48 lar home systems increased the average years of schooling by 41.96% and by 1.79 years,  
 49 respectively, and reduced the number of reported illnesses by 2.35%. Additionally, Kabir  
 50 et al. (2017) demonstrated that the installation of solar home systems in rural areas of  
 51 Bangladesh improved access to radio, cellphones, and the internet, integrating the ru-  
 52 ral population into a more global culture.

53 The proposed research aims to analyze the spatial-temporal patterns of PV adop-  
 54 tion in off-grid Bedouin settlements in the Negev. A significant challenge, as previously  
 55 mentioned, is the lack of official records of solar PV installations. To address this issue,  
 56 deep learning algorithms will be employed. The primary goal of this work is to map PV  
 57 systems using the U-Net architecture, with the following operative goals:

- 58 1. Creating the database of the satellite images.

- 59            2. Successfully training the U-Net architecture on the data.  
 60            3. Accuracy evaluation of the trained model.

61            **2 Background**

62            **2.1 Image Segmentation**

63            Image segmentation is a fundamental process in computer vision that involves di-  
 64            viding an image into meaningful regions or segments. This process, is crucial for vari-  
 65            ous applications including medical imaging, autonomous driving, and satellite imagery  
 66            analysis. A key area of interest within image segmentation is semantic segmentation. The  
 67            goal of semantic segmentation is to assign a semantic label to each pixel in the image,  
 68            effectively dividing the image into distinct segments based on their meaning. There have  
 69            been many research efforts to incorporate computer vision techniques which most of them  
 70            involving deep learning methods in the field of semantic segmentation and particularly  
 71            in the field of remote sensing data. For example, Weng et al. (2020) have used deep learn-  
 72            ing to get accurate segmentation of lakes and rivers. Edun et al. (2021) used deep learn-  
 73            ing algorithm to segment solar panels in a satellite imagery.

74            **2.2 Convolutional Neural Networks**

75            Convolutional Neural Networks (CNNs) have been a cornerstone in the advance-  
 76            ment of image recognition and classification tasks. CNNs are designed to automatically  
 77            and adaptively learn spatial hierarchies of features from input images, which makes them  
 78            highly effective for a wide range of visual recognition tasks. CNNs usually consist of con-  
 79            volutional layers, full-connected layers and pooling layers which reduces spatial resolu-  
 80            tion (Minaee et al., 2022).

81            **2.2.1 Rectified Linear Units**

82            Rectified Linear Units (ReLU) is an activation function introduced by (Fukushima,  
 83            1969). The ReLU function is defined by  $f(x) = \max(0, x)$ , which outputs a linear func-  
 84            tion when  $x \geq 0$  as illustrated in Figure 1.

85            **2.2.2 Convolutional Layer**

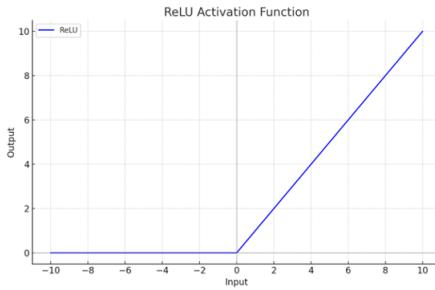
86            A convolutional layer (Conv Layer) uses a kernel to extract features from an im-  
 87            age. The kernel is a small weight matrix that slides over the input image, performing  
 88            element-wise multiplication between the kernel and the overlapping region of the image.  
 89            It then sums the products to form a single value in the output feature map (Wu, 2017).

90            **2.2.3 Pooling Layers**

91            Pooling layers are primarily used for downsampling the feature maps produced by  
 92            convolutional layers, helping to reduce the spatial dimensions of the input. In a pool-  
 93            ing layer, one can define a stride, which is the step size by which the filter moves. For  
 94            example, if the stride size is 2, the filter moves 2 pixels at a time (Wu, 2017).

95            **2.3 Fully Convolutional Networks**

96            Building on the success of CNNs, Long et al. (2015) proposed Fully Convolutional  
 97            Networks (FCNs) to address the limitations of CNNs in pixel-wise classification tasks.  
 98            FCNs include only convolutional layers, allowing them to output a segmentation map  
 99            with the same size as the input image. FCNs have been used to solve a variety of seg-  
 100            mentation problems. For instance, G. Wang et al. (2018) used FCNs for brain tumor seg-



**Figure 1.** ReLU activation function which outputs 0 when  $x$  is negative and outputs a linear function where the slope equals 1 when  $x \geq 0$ .

mentation, and Liu et al. (2016) employed them for accurate iris segmentation in non-cooperative environments.

Despite the advancements offered by FCNs, a notable disadvantage is the reduction in spatial resolution caused by down-sampling operations, such as pooling layers. These down-sampling steps are necessary to capture larger contextual information and reduce computational complexity, but they result in a loss of fine-grained details. When the feature maps are up-sampled to the original resolution, the output can be coarse and less precise (Guo et al., 2018). Fortunately, encoder-decoder architectures have been developed to address this issue.

## 2.4 Encoder-Decoder Architectures

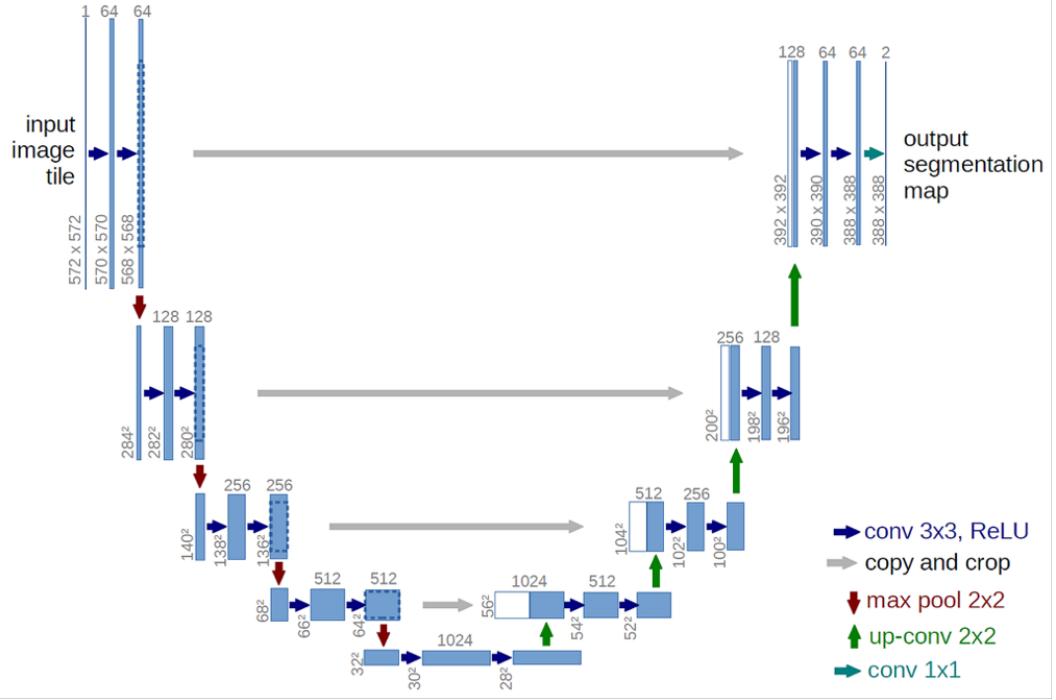
Encoder-Decoder architectures, which are a type of FCNs, use skip connections to combine semantic information extracted from deep layers with higher resolution information from earlier layers, effectively mitigating the resolution loss problem. This approach, is implemented in various deep learning algorithms, including SegNet (Badrinarayanan et al., 2017) and U-Net (Ronneberger et al., 2015), among others.

### 2.4.1 U-Net Architecture

As our encoder-decoder architecture, we used the U-Net architecture. U-Net consist of two paths. The first is a contracting path (left side) and the second is an expansive path (right side) as illustrated in Figure 2. The contracting path involves repeated application of two 3x3 convolutional layers, each followed by a ReLU activation function, and a 2x2 max pooling layer with a stride of 2 for downsampling. This path extracts important features from the input image. The expansive path begins with upsampling of the feature map, followed by a 2x2 convolution that halves the number of feature channels. This path reconstructs the spatial dimensions of the feature maps. To mitigate the resolution loss caused by downsampling, the U-Net architecture employs skip connections. These connections link layers from the contracting path to corresponding layers in the expansive path, effectively combining semantic information from deeper layers with high-resolution information from earlier layers.

## 2.5 Binary Classification Task

As mentioned before, semantic segmentation is a pixel-wise classification test. Namely, we want to give a semantic label to each pixel, which depends on whether it contains a



**Figure 2.** U-Net architecture. Each blue box corresponds to a multi-channel feature map. The number of the channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

PV or not. Therefore, this is a binary classification task. In a binary classification task, the goal is to classify input data into one of two classes, which can be defined as follows:

Let  $y = 1$  be the positive class and  $y = 0$  be the negative class. Given a data set  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ , where  $x_i \in \mathbb{R}^d$  is the feature vector and  $y_i \in \{0, 1\}$  is the corresponding binary label for the  $i$ 'th example, a model for binary classification can be represented as a function  $f(x; \theta)$  where  $\theta$  is the model's parameters. The output of the model is typically a probability. Let  $p_i$  be the output of the model for the  $i$ 'th example. Then to decide to which class to assign the  $i$ 'th example, a certain threshold can be used (typically 0.5) (Cristianini & Shawe-Taylor, 2000). For instance,

$$\text{if } p_i > 0.5 \text{ then } y_i = 1 \text{ else } y_i = 0$$

### 2.5.1 Accuracy Measures

To evaluate the model's performance in a binary classification task we used few of the ones presented by (Sokolova & Lapalme, 2009). These measures include precision, recall, F1 score and intersection over union. Below are the definitions and calculation for each metric

#### 2.5.1.1 Confusion Matrix

The confusion matrix offers a detailed breakdown of the classification model's performance by displaying the counts of true positives (TP), true negatives (TN), false pos-

149 itives (FP), and false negatives (FN). It serves as a foundational tool, facilitating a clearer  
 150 understanding of the other accuracy measures used in the model evaluation.

151 **2.5.1.2 Precision**

152 Precision measures the proportion of true positive predictions among all positive  
 153 predictions. It indicates the accuracy of the positive class predictions.

$$\text{Precision} = \frac{TP}{TP + FP}$$

154 **2.5.1.3 Recall**

155 Recall measures the proportion of true positives predictions among all actual pos-  
 156 itive instances.

$$\text{Recall} = \frac{TP}{TP + FN}$$

157 **2.5.1.4 F1 Score**

158 The F1 score is the harmonic mean of precision and recall, providing a balance be-  
 159 tween the two metrics.

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

160 **2.5.1.5 Intersection over Union (IoU)**

161 IoU measures the overlap between the predicted positive regions and the actual pos-  
 162 itive regions.

$$\text{IoU} = \frac{TP}{TP + FP + FN}$$

163 **2.6 PVs Mapping Using Encoder-Decoder Architectures**

164 Encoder-decoder architectures have been widely used in the field of remote sens-  
 165 ing, particularly in the detection and mapping of solar PV systems. These architec-  
 166 tures are designed to handle high-resolution aerial imagery, enabling precise and efficient map-  
 167 ping of PV arrays. Camilo et al. (2018) employed the SegNet architecture to semanti-  
 168 cally segment PV arrays in high-resolution aerial imagery. Their approach demon-  
 169 strated substantial improvements in estimating the shape and size of PV arrays compared to tra-  
 170 ditional CNN-based PV detection methods. Castello et al. (2019) utilized a U-Net ar-  
 171 chitecture for the automatic detection of rooftop solar panels. Their approach involved  
 172 using high-resolution aerial photos and extensive data augmentation to enhance model  
 173 performance. The U-Net architecture, known for its efficiency in biomedical image seg-  
 174 mentation, was adapted to detect solar panels in urban environments. They reported an  
 175 accuracy of about 0.94 and an IoU of up to 0.64, showcasing the model's effectiveness  
 176 in segmenting PV arrays from complex backgrounds. Gonzalez et al. (2021) also explored  
 177 the use of the U-Net architecture for the detection and mapping of PV systems. By lever-  
 178 aging high-resolution satellite images and preprocessing techniques, their model was able  
 179 to accurately identify and map PV arrays. Their U-Net architecture, demonstrated high  
 180 performance in identifying PV arrays even in densely built environments as they achieved  
 181 an IoU of 0.904. This seminar work aims to use the U-Net architecture for the automatic  
 182 detection of PVs in off-grid Bedouin settlements in the Negev. However, a challenge arises

183 as the solar panels are not placed only on rooftops but also on the ground (unlike urban  
184 areas), which increases the possibility of the model misclassifying the panels.

### 185 3 Methodology

#### 186 3.1 Database Creation

187 This research database was meticulously established by gathering satellite images  
188 of Bedouin settlements in the northern Negev, provided by the Municipality of Bedouin  
189 Settlements in the Negev. The database encompasses satellite images from two distinct  
190 years, 2017 and 2020. Each image is comprised of three channels—Red, Blue, and Green  
191 (RGB)—with a cell resolution of 0.14 meters and adheres to the Israel TM Grid coor-  
192 dinate system (EPSG:2039). Given that satellite images inherently lack labels, which are  
193 crucial for semantic segmentation tasks, we undertook the digitization process manually.  
194 Initially, we uploaded a district layer of the settlements along with the raster files and  
195 created a new feature class (Polygon Layer) specifically for digitizing the PV systems.  
196 We digitized each PV system within each district, resulting in a total of 20,721 and 28,599  
197 solar panels for the years 2017 and 2020, respectively. The digitization was performed  
198 using ArcGIS Pro. Following the digitization process, we employed a polygon layer to  
199 divide our research area into eight distinct parts (757.071 squared km). This division was  
200 essential to manage the data effectively and to prepare the images for subsequent pro-  
201 cessing steps. We utilized ArcGIS Pro's "Model Builder" to generate, for each part of  
202 the research area, a shapefile containing the solar panels and the corresponding raster  
203 section of the area. Each raster was named  $Image_i$ , and each shapefile was named  $Labels_i$ ,  
204 where i represents the number of the area. For instance, area number 1 has a raster named  
205  $Image_1$  and a shapefile named  $Labels_1$ . The rationale behind dividing our research area  
206 was to facilitate the slicing process, which will be further discussed in the data prepro-  
207 cessing section. This structured approach ensured a comprehensive and organized dataset,  
208 facilitating accurate analysis and classification of PV systems within the Bedouin set-  
209 tlements. The creation of this database represents a significant step forward in our un-  
210 derstanding and tracking of photovoltaic system deployment in the northern Negev re-  
211 gion.

#### 212 3.2 Data Preprocessing and U-Net Implementation

213 In this sub-section, I detail the data preprocessing methods employed to prepare  
214 the satellite imagery as well as the U-Net implementation for the semantic segmenta-  
215 tion task. The primary objectives were to create raster files containing RGB channels  
216 and a label channel, generate raster images of size 256x256x3 and binary label images  
217 of size 256x256x1 (where pixels hold the value of 255 if they are part of a PV system)  
218 and implement the U-Net architecture. Additionally, we also established a folder struc-  
219 ture as represented in Figure 3 and divided the data into training, validation, and test-  
220 ing sets before loading it.

##### 221 3.2.1 Raster Files Creation

222 Our database consists of satellite imagery of eight areas for the years 2017 and 2020.  
223 To process these images, we utilized the Geospatial Data Abstraction Library (GDAL).  
224 The GDAL commands were executed through Jupyter Notebook using the *subprocess*  
225 module, which allows running shell commands from within the Python environment. The  
226 first step involved converting the input satellite images to RGB format. This conversion  
227 was necessary because ArcGIS Pro, which was used for initial processing, added an ad-  
228 ditional fourth band known as the "alpha" band, representing the opacity of the cells.  
229 As the alpha band was not required for our analysis, we removed it, retaining only the  
230 three essential bands RGB. After removing the unnecessary alpha band, we generated

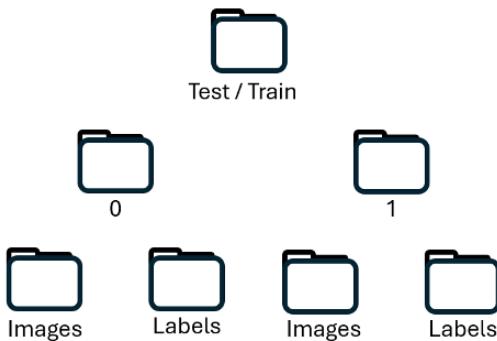
231 the label band by initializing it with zero values. Subsequently, we used GDAL to in-  
 232 tegrate the labels from the shapefile into the raster by "burning" the corresponding la-  
 233 bel values into the image pixels. This process involved rasterizing the shapefile data, en-  
 234 suring that each pixel in the label band correctly represented the presence or absence  
 235 of a PV system. Finally, we consolidated the four bands (Red, Green, Blue, and Label)  
 236 into a single combined raster file. This comprehensive file includes all necessary data for  
 237 the subsequent analysis, ensuring that the label information is seamlessly integrated with  
 238 the image data.

### 239 **3.2.2 Raster Files Preprocessing**

240 On the preprocessing part, we have sliced the images to size of 1024x1024x4 as slic-  
 241 ing it to 256x256x4 was memory consuming our server. Each raster was then saved in  
 242 a folder named tiles which from there we loaded them using *rasterio* (Python library).  
 243 Then, for each file, we counted the number of pixels containing a PV system and cre-  
 244 ated a file which contains information about the rasters (i.e., filename, rows, columns,  
 245 number of pixels containing PV). Finally, we got rid of rasters with less than 1024 rows  
 246 / columns and dropped randomly 50% of the rasters classified as 0 as we don't need that  
 247 much data.

### 248 **3.2.3 Raster Organization**

249 To store the preprocessed raster files we created a hierarchical folder structure where  
 250 the root folder, named either 'train' or 'test' contains two sub-folders named '0' and '1'.  
 251 These sub-folders indicate whether an image contains a solar panel (1) or not (0). Each  
 252 of these sub-folders further contains two folders: 'Images' and 'Labels' where the 'Im-  
 253 ages' folder holds the processed raster images, while the 'Labels' folder contains the cor-  
 254 responding binary raster files. After establishing the folder structure, we utilized a fil-  
 255 tered file that contained the filenames of the rasters along with their classifications to  
 256 facilitate easy access to the file paths. We loaded these files and separated the raster bands  
 257 into two distinct rasters: one containing only the RGB bands and the other containing  
 258 only the label band. Each RGB raster and label raster was named consistently and placed  
 259 in the corresponding folder as described in the previous section. Both raster files were  
 260 saved in the ".tif" format. This organization ensured that each set of RGB and label rasters  
 261 could be easily identified and accessed, streamlining the workflow for subsequent anal-  
 262 ysis and processing tasks.



**Figure 3.** The root folder is either 'train' or 'test'. Each contains two folders named '0' and '1', corresponding to whether the image contains a solar panel. Each of these folders contains 'Images' and 'Labels' sub-folders, which hold the images and labels, respectively.

263 **3.2.4 Raster Slicing**

264 After organizing the rasters, we created text files containing the paths to the im-  
 265 ages and labels. For example, a text file named *image\_1\_training.txt* contains all the paths  
 266 to images that include solar panels and are designated for training. These text files fa-  
 267 cilitated the loading of the rasters in an efficient manner. The loaded rasters and their  
 268 corresponding labels were then sliced into smaller segments of 256x256x3 using NumPy.  
 269 Each segment was saved in a new but similar folder structure to the one defined in sec-  
 270 tion 3.2.3, ensuring consistency and ease of access. By slicing the rasters into smaller,  
 271 more manageable sizes, we optimized them for the subsequent semantic segmentation  
 272 tasks, making the data more suitable for our U-Net architecture.

273 **3.2.5 Data Preparation and Loading**

274 Before implementing the U-Net architecture, it was crucial to load and prepare the  
 275 data for training the model. We began by creating four lists of paths for each dataset  
 276 (training and testing): two lists for the images and two for the labels, corresponding to  
 277 each class. These lists were then combined to form the complete datasets. Next, we cre-  
 278 ated a validation set by randomly sampling 10% of the training data, ensuring a repre-  
 279 sentative subset of image-label pairs. This resulted in three organized datasets: Train-  
 280 ing, Validation, and Testing. To load the data, we used Rasterio and then employed NumPy  
 281 to stack the arrays to prepare them for model training. Given the large size of our raster  
 282 database and the inefficiency of standard Python methods, we leveraged the Dask library  
 283 to perform multiple tasks simultaneously, greatly enhancing the data loading process.  
 284 Finally, we saved the data for each set in npz format. By following these steps, we en-  
 285 sured that the data was efficiently loaded and prepared for training the U-Net model.

286 **3.2.6 U-Net Implementation**

287 We implemented our U-Net model using *TensorFlow 2.10* and *Python 3.10.8* and  
 288 tailored the model to accommodate limited computational resources by adding dropout  
 289 layers. The architecture features an input layer designed for images with predetermined  
 290 dimensions (256,256,3) followed by a normalization Lambda layer. In the contracting path,  
 291 each convolutional block consists of two 3x3 convolutional layers with ReLU activation  
 292 function and *he\_normal* initialization, followed by dropout of 0.1 or 0.2, depending on  
 293 the layer depth. After convolution, each block employs a 2x2 max pooling layer. As the  
 294 network deepens, from 64 filters in the initial layer to 1024 in the deepest layer, we strate-  
 295 gically increased the dropout rate to 0.3 to efficiently manage resource usage and pre-  
 296 vent overfitting. The expansive path mirrors this setup but includes upsampling and con-  
 297 catenation steps as mentioned previously in the background section. Additionally, we  
 298 implemented a FocalDice loss function which is commonly used to address class imbal-  
 299 ance problems in image segmentation tasks although, it is primarily used in medical imag-  
 300 ing as per existing research (P. Wang & Chung, 2018; Yeung et al., 2022). Our FocalDice  
 301 loss function leveraging the strengths of both by computing focal loss and dice loss for  
 302 each observation and average the results while giving the dice and focal similar weight.  
 303 The loss function can be defined as:

$$\mathcal{L}(y, \hat{y}) = 0.5 \frac{1}{N} \sum_{i=1}^n (-\alpha(1 - p_i)^\gamma \log(p_i) + (1 - \frac{2 \sum_{j=1}^m y_{ij} \hat{y}_{ij}}{\sum_{j=1}^m y_{ij}^2 + \sum_{j=1}^m \hat{y}_{ij}^2}))$$

304 The first part in the loss is the Focal loss function and the second part is the Dice loss  
 305 function. Where  $\alpha$  is the weight of the positive class and  $\gamma$  controls the attention that  
 306 the model gives to hard examples. These parameters are chosen by the user. The rea-  
 307 son we chose this loss function over the binary cross-entropy loss is to better address class  
 308 imbalance with the focal mechanism and to improve model performance by enhancing  
 309 the overlap between the predicted labels and the ground truth, as measured by the Dice  
 310 coefficient.

311            **3.3 Training Process**

312        In the training phase we merged datasets spanning two distinct years to form a com-  
 313        prehensive training set, aiming to enhance the model’s robustness and generalizability.  
 314        Subsequently, we trained four separate U-Net models, each configured with a different  
 315         $\gamma$  value, to systematically evaluate the impact of this parameter on the validation loss.  
 316        The values of  $\gamma$  were set incrementally: the first model was trained with  $\gamma = 1$ , increas-  
 317        ing sequentially up to  $\gamma = 4$  for the fourth model ( $\alpha$  was 0.25 for the 4 models). Each  
 318        model was optimized using the ADAM optimizer (Kingma & Ba, 2014) with a learning  
 319        rate of  $1e - 5$ . Furthermore, we used a batch size of 32 as Gonzalez et al. (2021) used  
 320        a batch size of 8 to process images of size 512x512 while training their U-Net model in  
 321        order to preserve the amount of pixels our model learns from each batch. Finally, we used  
 322        an early stopping method with patience of 2 which monitored our validation loss to pre-  
 323        vent overfitting.

324            **3.4 Evaluation**

325        To evaluate our model performance, we saved a test set that will not be seen by  
 326        our model during its training session. The model performance was evaluated using the  
 327        following metrics: IoU, F1 Score, Recall, and Precision. IoU measures the overlap be-  
 328        tween predicted and actual segments, F1 Score balances Precision and Recall, Recall in-  
 329        dicates the proportion of actual positives correctly identified, and Precision measures the  
 330        correctness of positive predictions.



**Figure 4.** satellite images from Bedouin settlements and their digitization (green rectangles). The top and bottom images are from 2017 and 2020 respectively.

331            **4 Results**

332            **4.1 Database Overview**

333        The database consists of high-resolution satellite images from Bedouin settlements  
 334        in the northern Negev, captured in 2017 and 2020. The images were manually digitized  
 335        using ArcGIS Pro to identify and label solar PV systems, resulting in thousands of la-  
 336        beled panels across different districts as illustrated in Figure 4. The images present vary-  
 337        ing challenges in visibility; some panels are easily discernible, while others blend with  
 338        the surroundings due to environmental conditions and shadowing as illustrated in Fig-  
 339        ure 5.



**Figure 5.** Hard to classify PVs and easy to classify PVs on the left image and right image respectively.

#### 340 4.2 Segmentation Results

341 Our final database consists of images with dimensions 256x256x3 and corresponding  
 342 labels of size 256x256x1. We trained five models on this dataset, each with a different  
 343 value of  $\gamma$ . Among these, the model with  $\gamma = 2$  achieved the lowest validation loss,  
 344 as illustrated in Figure 6, and thus has been selected as our final model. While the model  
 345 demonstrated the ability to segment some of the PVs accurately, it also exhibited sig-  
 346 nificant limitations. Specifically, the model incorrectly segmented various objects that  
 347 were not solar panels, particularly cars. Few of the segmentation results are illustrated  
 348 in Figure 7.

**Table 1.** U-Net Model Results for Different Datasets

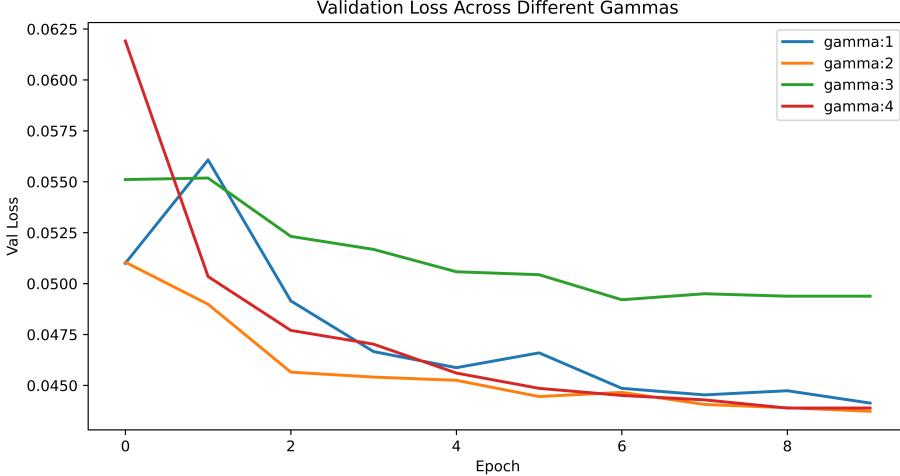
| Dataset    | Precision | Recall | F1 Score | IoU   |
|------------|-----------|--------|----------|-------|
| Train      | 0.511     | 0.269  | 0.352    | 0.217 |
| Validation | 0.284     | 0.296  | 0.290    | 0.148 |
| Test       | 0.337     | 0.201  | 0.252    | 0.144 |

#### 349 4.3 Model Evaluation

350 Our model achieved an average loss of 0.442 on the training set and 0.457 on the  
 351 validation set after 13 epochs. For the training set, the model recorded a precision of 0.511,  
 352 a recall of 0.269, and an IoU of 0.217. On the validation set, the precision was 0.284, re-  
 353 call was 0.296, and IoU was 0.148. Using the precision and recall metrics, we calculated  
 354 the F1 Score, resulting in 0.352 for the training set and 0.290 for the validation set. Af-  
 355 ter completing the training, we evaluated the model on the test set, obtaining the fol-  
 356 lowing results:

- 357 • Precision: 0.337
- 358 • Recall: 0.201
- 359 • IoU: 0.144
- 360 • F1 Score: 0.252

361 These results are summarized in Table 1. We reported precision and recall rather  
 362 than accuracy due to the imbalanced nature of our data, which makes accuracy a less  
 363 meaningful metric.



**Figure 6.** Validation loss error across different values of  $\gamma$  where  $\gamma \in [1, 4]$ .

364

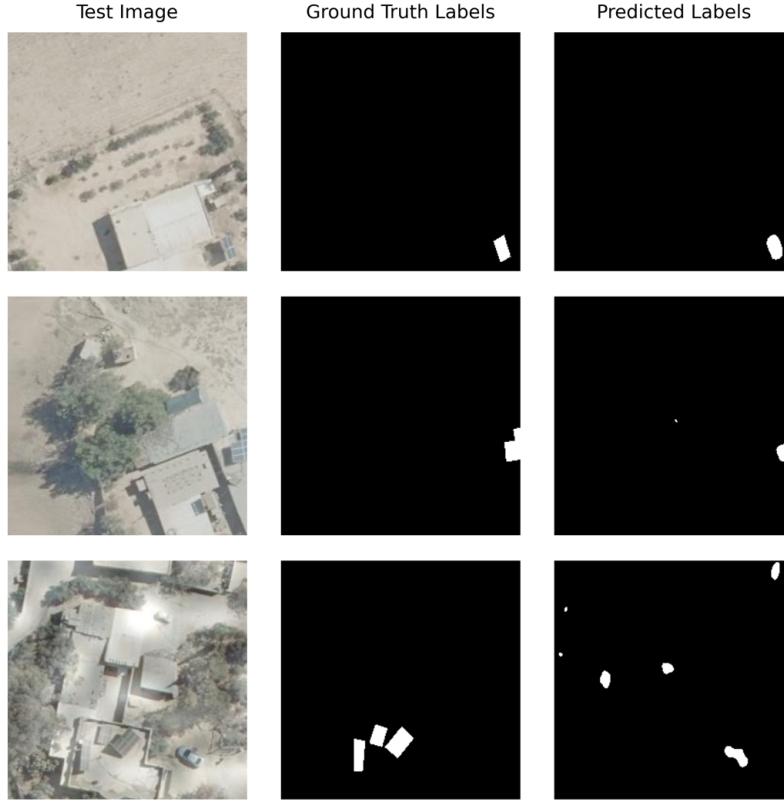
## 5 Discussion

365

### 5.1 Comparative Analysis of Prior and Present Results

366

In previous research, the U-Net architecture was effectively used to detect PVs, achieving high evaluation scores. For instance, Gonzalez et al. (2021) reported an IoU of 0.904 and an accuracy of 0.954 using U-Net. Similarly, Castello et al. (2019) achieved an IoU of 0.64 and an accuracy of 0.94 with U-Net. In contrast, our study, also utilizing U-Net, yielded a precision of 0.337, recall of 0.201, and an IoU of 0.144 on the test set. This significant performance disparity is likely due to the unique challenges of our dataset. Unlike the urban-focused datasets in previous studies, ours includes off-grid Bedouin settlements where PV systems are placed both on rooftops and on the ground, complicating accurate segmentation. Additionally, our computational limitations required us to discard some data, potentially reducing the effectiveness of our model training. Furthermore, we used a classification threshold of 0.5 based on our judgment rather than the precision-recall curve approach employed by (Castello et al., 2019), which might have improved our predictions.



**Figure 7.** Comparison of Test Images, Ground Truth Labels, and Predicted Labels. The third row emphasizes the model’s unintended focus on cars, illustrating a notable misclassification.

## 379           5.2 Impact on Renewable Energy Policy for Off-Grid Communities

380           The research holds substantial significance in the context of renewable energy adoption  
 381           in off-grid communities. Accurate mapping of PV systems in off-grid Bedouin settle-  
 382           ments is crucial for renewable energy policy and infrastructure development. Reliable  
 383           data on PV deployment can inform policy decisions, support the planning and expan-  
 384           sion of renewable energy projects, and enhance the integration of solar energy into the  
 385           energy grid. This can lead to better resource allocation, more efficient energy use, and  
 386           ultimately, improved quality of life in underserved regions. The methodology and find-  
 387           ings from this research can be applied to similar contexts globally, providing a valuable  
 388           framework for automating the detection and mapping of PV systems in challenging en-  
 389           vironments. By improving the accuracy and efficiency of PV system detection, we can  
 390           better support the transition to sustainable energy sources in off-grid communities, con-  
 391           tributing to broader environmental and socio-economic benefits.

## 392           5.3 Limitations

393           The research faced several notable challenges:

- 394           1. **Data Imbalance:** The dataset had a significant imbalance between non-PV and  
 395           PV areas, which likely contributed to the high rates of false positives and false neg-  
 396           atives. The limited number of PV instances made it difficult for the model to learn  
 397           effectively.

- 398      2. **Complexity of Satellite Images:** The satellite images used in this study pre-  
 399      sented varying levels of visibility, with some PV systems blending into their sur-  
 400      roundings due to environmental conditions. This added complexity to the task of  
 401      accurately identifying and segmenting the PV systems.  
 402      3. **Computational Resources:** Training a deep learning model like U-Net while  
 403      also loading high resolution image data is computationally heavy in terms of mem-  
 404      ory. As our resources were limited, we had to give up on some data to get our re-  
 405      sults.

406      **5.4 Conclusions**

407      This study represents an important step toward automating the detection and map-  
 408      ping of PV systems in off-grid Bedouin settlements using deep learning techniques. De-  
 409      spite the challenges and limitations, the research provides valuable insights and a foun-  
 410      dation for future improvements. By addressing the identified challenges and exploring  
 411      advanced techniques such as data augmentation, alternative loss functions, and model  
 412      architecture modifications, we can enhance the accuracy and reliability of PV system de-  
 413      tection. Ultimately, this will support the sustainable development of renewable energy  
 414      infrastructure in off-grid areas, contributing to broader environmental and socio-economic  
 415      benefits.

416      **Acknowledgments**

417      I would like to thank Prof. Tal Svory and Roni Bluestein for their guidance and help along  
 418      the way. Special thanks to David Loffe for his assistance in digitization. This research  
 419      would not have been possible without their support and expertise.

420      **References**

- 421      Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolu-  
 422      tional encoder-decoder architecture for image segmentation. *IEEE transactions*  
 423      *on pattern analysis and machine intelligence*, 39(12), 2481–2495.  
 424      Camilo, J., Wang, R., Collins, L., Bradbury, K., & Malof, J. (2018). Application of  
 425      a semantic segmentation convolutional neural network for accurate automatic  
 426      detection and mapping of solar photovoltaic arrays in aerial imagery. *arXiv*  
 427      *preprint arXiv:1801.04018*.  
 428      Castello, R., Roquette, S., Esguerra, M., Guerra, A., & Scartezzini, J. (2019). Deep  
 429      learning in the built environment: Automatic detection of rooftop solar panels  
 430      using convolutional neural networks. In *Journal of physics: Conference series*  
 431      (Vol. 1343, p. 012034).  
 432      Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector ma-*  
 433      *chines and other kernel-based learning methods*. Cambridge university press.  
 434      de Hoog, J., Maetschke, S., Ilfrich, P., & Kolluri, R. (2020). Using satellite and  
 435      aerial imagery for identification of solar pv: State of the art and research op-  
 436      portunities. In *Proceedings of the eleventh acm international conference on*  
 437      *future energy systems* (pp. 308–313).  
 438      Diallo, A., & Moussa, R. (2020). The effects of solar home system on welfare in off-  
 439      grid areas: Evidence from côte d'ivoire. *Energy*, 194, 116835.  
 440      Edun, A., Perry, K., Harley, J., & Deline, C. (2021). Unsupervised azimuth es-  
 441      timation of solar arrays in low-resolution satellite imagery through semantic  
 442      segmentation and hough transform. *Applied Energy*, 298, 117273.  
 443      Fukushima, K. (1969). Visual feature extraction by a multilayered network of ana-  
 444      log threshold elements. *IEEE Transactions on Systems Science and Cybernet-*  
 445      *ics*, 5(4), 322–333.  
 446      Gonzalez, L., Silva, J., Smith, A., & Sanchez, M. (2021). Enhanced detection

- and mapping of photovoltaic systems using deeplabv3 architecture and high-resolution satellite imagery. *Remote Sensing*, 13(4), 560.
- Guo, Y., Liu, Y., Georgiou, T., & Lew, M. (2018). A review of semantic segmentation using deep neural networks. *International journal of multimedia information retrieval*, 7, 87–93.
- Imamoglu, N., Kimura, M., Miyamoto, H., Fujita, A., & Nakamura, R. (2017). Solar power plant detection on multi-spectral satellite imagery using weakly-supervised cnn with feedback features and m-pcnn fusion. *arXiv preprint arXiv:1704.06410*.
- Kabir, E., Kim, K., & Szulejko, J. (2017). Social impacts of solar home systems in rural areas: A case study in bangladesh. *Energies*, 10(10), 1615.
- Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Liu, N., Li, H., Zhang, M., Liu, J., Sun, Z., & Tan, T. (2016). Accurate iris segmentation in non-cooperative environments using fully convolutional networks. In *2016 international conference on biometrics (icb)* (pp. 1–8).
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3431–3440).
- Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2022). Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7), 3523–3542.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention-miccai 2015: 18th international conference, munich, germany, october 5-9, 2015, proceedings, part iii 18* (pp. 234–241).
- Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4), 427–437.
- Wang, G., Li, W., Ourselin, S., & Vercauteren, T. (2018). Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. In *Brainlesion: Glioma, multiple sclerosis, stroke and traumatic brain injuries: Third international workshop, brainles 2017, held in conjunction with miccai 2017, quebec city, qc, canada, september 14, 2017, revised selected papers 3* (pp. 178–190).
- Wang, P., & Chung, A. (2018). Focal dice loss and image dilation for brain tumor segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support: 4th international workshop, dlmia 2018, and 8th international workshop, ml-cds 2018, held in conjunction with miccai 2018, granada, spain, september 20, 2018, proceedings 4* (pp. 119–127).
- Weng, L., Xu, Y., Xia, M., Zhang, Y., Liu, J., & Xu, Y. (2020). Water areas segmentation from remote sensing images using a separable residual segnet network. *ISPRS international journal of geo-information*, 9(4), 256.
- Wu, J. (2017). Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5(23), 495.
- Yeung, M., Sala, E., Schönlieb, C., & Rundo, L. (2022). Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, 95, 102026.

## 496 Appendix A Supplementary Materials

497 The code and data used in this seminar work are available on GitHub: <https://github.com/LidorErez98/Seminar>.