

HyperionSolarNet: Solar Panel Detection from Aerial Images

Felipe Vergara

felipe.vergara@berkeley.edu

Nathan Nusaputra

nusaputra137@berkeley.edu

Poonam Parhar

poonamparhar@berkeley.edu

Ryan Sawasaki

rsawasaki@berkeley.edu

Abstract

With the effects of global climate change impacting the world, collective efforts are needed to reduce greenhouse gas emissions. Many of these efforts are focused on the energy sector, which is the single largest contributor to climate change. However, ongoing strategies to reduce dependence on carbon-emitting plants have been thwarted due to the inability to accurately forecast renewable energy production, such as solar power. In addition to weather conditions, solar energy generation is dependent on the location and total surface area of solar panels. While it has been a challenge developing a comprehensive database of solar panel data, one solution is the use of artificial intelligence to detect solar panel locations. HyperionSolarNet utilizes deep learning methods for automated detection of solar panel locations and their surface area using aerial imagery. The framework, which consists of a two-branch model using an image classifier in tandem with a semantic segmentation model, provides an efficient and scalable method for detecting solar panels with reliable performance. On the application side, our work includes a user-interface that provides a tool to help visualize the output from the classification and segmentation models using a maps platform. The results from our work can generate additional imagery data that can be applied towards further research and model improvements.

1. Introduction

As awareness of the impacts and risks of climate change continue to increase, efforts are being made to reduce greenhouse gas emissions. Within the energy industry, strategies are being deployed to lower carbon emissions by reducing fossil fuel energy sources and integrating renewable energy. One source of renewable energy is solar power, which is growing in popularity due to lower production costs of photovoltaic (PV) solar panels, more efficient energy output and government-funded incentive programs. Although solar panel production is surging, there has not been a significant reduction in fossil fuel energy due to the inability to accurately forecast solar power generation. Carbon-emitting plants remain operational and run idle to reliably manage energy supply and demands.

Solar energy generation is dependent on weather conditions, including solar irradiance and temperature. Due to the irregular nature of these factors, accurate forecasting of solar

energy is challenging. Studies have approached this challenge by leveraging machine learning methods, such as Long-Term Recurrent Convolutional Network architecture using temporal and spatial weather data over a specified geographic area [1].

In addition to weather conditions, solar energy output is dependent on the quantity and surface area of solar panels. Developing a database of the locations and total surface area of solar panels within a given geographic area would benefit utility providers to help quantify solar power supply and manage overall energy demands. However, gathering accurate data on the location and area of solar panels introduces additional challenges. To address these challenges, this work presents deep learning methods and a framework for detecting the solar panel locations and their surface area from aerial images.

1.1 Related Studies

With the expansion of residential and commercial solar panel installations, numerous efforts to determine the location and quantity of solar panels have been conducted to varying levels of reliability. Traditional estimates are based on government funded studies reliant on existing reports, information from federal renewable energy incentive programs and data from local and state agencies [2]. While these studies offer utility in providing a high-level estimate of the quantity of solar panels, the rapid pace of new solar panel installations, in addition to losing key data sources due to lack of funding and the phasing out of incentive programs, results in these studies quickly becoming outdated and unreliable.

As satellite imagery became more accessible, researchers began to leverage deep learning methods to detect solar panels. One such study that produced notable advances in this challenge is the DeepSolar model, which uses a two branch approach. Using a classification branch based on Google Inception V3, the model classifies satellite images as containing solar panels or not containing solar panels. Images containing solar panels are then fed into the segmentation branch, which aggregates feature maps learned through convolutional layers and generates Class Activation Maps used to estimate the size of solar panels. Due to the extensive resources and expenses required to create ground truth labeling of solar panels, the model uses a label-free, semi-supervised method to train the segmentation branch. DeepSolar is trained on over 350,000 images resulting in a precision and recall around 90% for classification and a mean relative error of 2.1% for size estimation performance [3].

In addition to the work conducted by DeepSolar, there have been similar studies that have used neural networks and satellite imagery to map solar panels. Studies have utilized convolutional neural network architecture, such as SegNet [4] and cross learning driven U-Net [5] for the segmentation of solar panels. Other studies have taken the two tier approaches such as using a Mobilenet classifier with U-Net architecture [6] and using low-quality images to detect PV panels then high quality images to determine the exact location of the panels [7]. A common issue among the studies is working with the computationally heavy segmentation models.

1.2 Contributions of our work

Many studies in this field deploy a two branch model using a less rigorous classification model to identify the solar panel images, which are then passed to a computationally-heavy segmentation model for mask prediction. Our work builds off of those foundational studies and introduces a number of enhancements with the following contributions:

- Implementation of an EfficientNet-B7 classifier in combination with a semantic segmentation model based on U-Net architecture utilizing EfficientNet-B7 encoder-backbone to predict the location and size of PV panels. The use of these models result in notable performance improvements. On the testset data we compiled, our classification model reports an accuracy of 95.63% , and the segmentation model achieves an IoU score of 82.43% .
- While our performance improvements are significant, of equal importance is that the performance is achieved using a fraction of the training data. Previous studies have required training data in the hundreds of thousands of images, which is a quantity of data that may be inaccessible for those who do not have the necessary resources. HyperionSolarNet is able to reach our performance levels using a training dataset of about a thousand images.
- With a smaller training dataset, it is feasible to produce ground-truth labeled data for the classification and segmentation model. While the labeling process still requires sufficient resources to complete, labeling a few hundred images is achievable in comparison to larger datasets used in previous studies. In addition, as HyperionSolarNet is applied in various locations, the masked images output from the segmentation model can be further utilized as training data to build an increasingly robust model.
- In addition to the research conducted on the two-branch model, we created a user-interface that identifies the predicted location and total area of solar panels within a geographical region. This web-application could be beneficial to solar panel companies and utility providers by providing a tool that visualizes the output from the classification and segmentation model using Google Maps.

2. Background and Methods

With the recent advances in deep learning methods, deep learning models are being extensively used in various computer vision tasks such as image classification, object detection, and image segmentation. The fast developments in these tasks have been further accelerated by Transfer Learning. Using Transfer Learning in a classification and segmentation model pipeline, this work presents a supervised deep learning method to estimate the number and surface area of PV panels from high resolution satellite images.

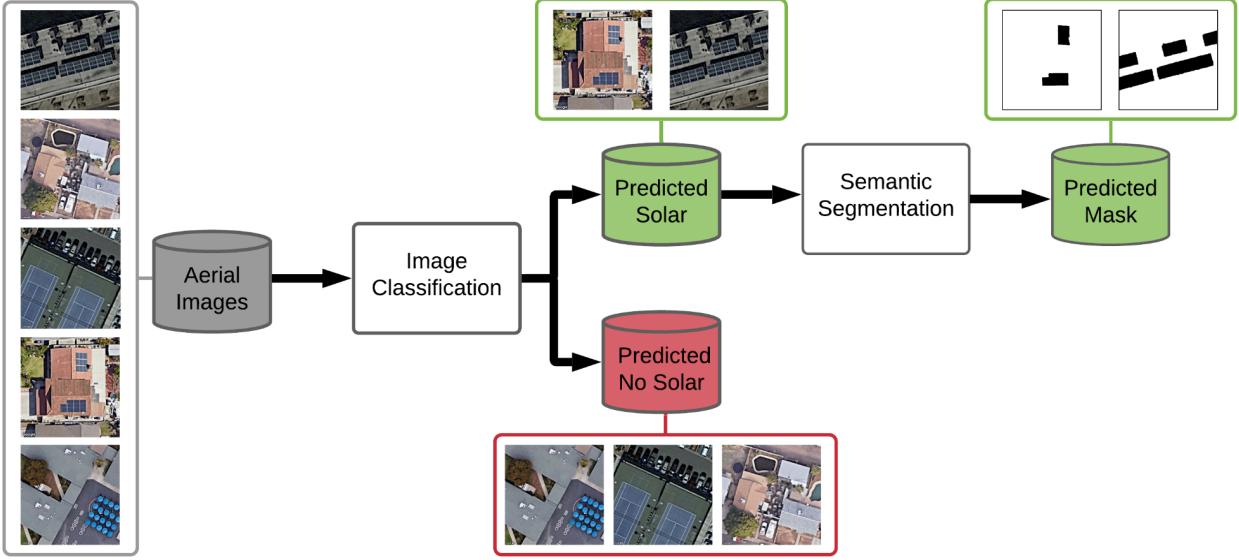


Figure 1: Architecture diagram of the solar panel area prediction pipeline. Satellite images are passed through an image classification model to identify images containing solar panels. The positive predictions from the image classification model are then run through an image segmentation model to identify the pixels belonging to solar panels in an image.

2.1 Image Classification

Image classification is the task of classifying images into the groups they belong to. For our case, we define two categories: images containing solar panels and images without any solar panels in them. There are several pre-trained models available for image classification, including but not limited to MobileNet, VGG-16, Inception-v3, RetNet and EfficientNet [8]. As shown in figure 2, there are trade-offs between accuracy and number of model parameters that are to be considered in model selection. We use transfer learning techniques to fine-tune pre-trained classification models for our specific classification task.

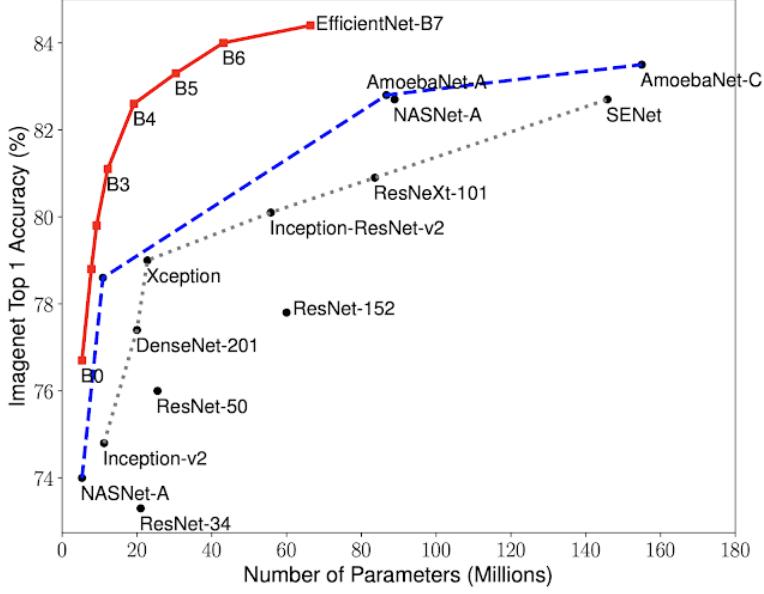


Figure 2: Accuracy scores of various Pre-trained Image Classification models [8]

2.2 Semantic Image Segmentation

Image segmentation is a challenging computer vision task where the goal is to classify each pixel of a given image to the specific class or object it belongs to. Semantic segmentation is a specific type of image segmentation where all the pixels representing a particular class are classified as a single entity.

U-Net [9] is one of the most popular deep learning based semantic segmentation methods. It is a convolutional neural network architecture having Encoder and Decoder layers that down-sample and up-sample the given images to extract features and classify pixels to specific classes. The Encoder or the contraction path is a set of convolutional and max-pooling layers that extract feature maps from the image and reduce its size as it passes through the encoder layers. On the other hand, the Decoder or the expansive path employs transposed convolutional layers along with feature maps from the corresponding encoder layers to restore the resolution and pixel localization information of the image. The layers of the Encoder can be arranged using any of the image classification neural networks, which is called the encoder-backbone of the U-Net architecture. The encoder-backbone also defines how the Decoder layers are built to up-sample the images.

2.3 Performance Metrics

The performance of our two branch model is evaluated on a validation dataset of 168 solar and 324 no-solar images. In addition, we examine model performance on an experiment

conducted on the entire city of Berkeley. For the classification model, we use the performance metrics of accuracy, precision, recall, and F1 score. For the semantic segmentation model, we use the performance metrics of intersection over union (IoU) and F1 score. The IoU metric is the overlap of the predicted mask and the ground-truth labeled mask divided by their union.

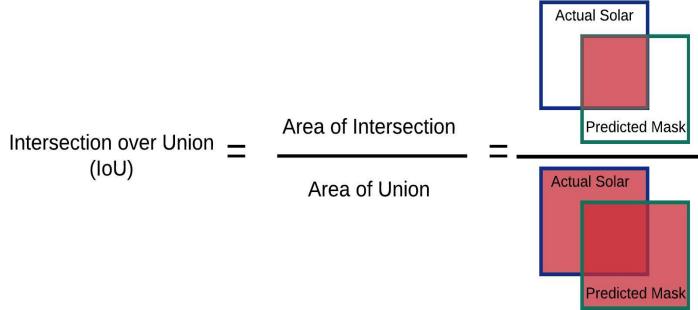


Figure 3: Intersection over Union (IoU) is the area of intersection of the actual object label and the predicted mask divided by their union

3. Dataset

Similar to most projects, quality data is a vital and important resource in supervised model training. An exhaustive search for an existing solar panel dataset was mostly unsuccessful with the exception of a few publicly available datasets that were outdated and imbalanced. Given the lack of available datasets, a significant amount of time was devoted to building our own dataset for the classification and segmentation tasks.

The dataset includes satellite images downloaded using Google Maps Static API. To help train robust models, images are selected with diverse features and collected from across the U.S. Batches of images are pulled from Arizona, California, Colorado, Florida, Hawaii, Idaho, Louisiana, Massachusetts, Nevada, New Jersey, New York, Oregon, Texas and Washington. The images are at zoom levels 20 and 21, and of sizes 416x416 and 600x600 pixels. In addition, the dataset contains a mix of both residential and commercial buildings. Early trained models revealed a number of false positives with images of objects that resemble solar panels. In subsequent image downloads, we focused on collecting no-solar images containing objects that could potentially be misclassified as solar panels, such as skylights, crosswalks, and sides of tall buildings, among others.

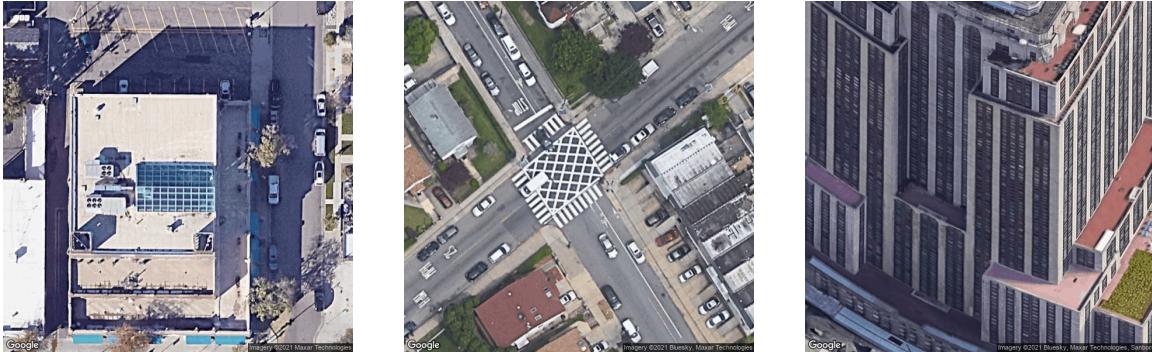


Figure 4: Examples of no-solar images with objects that resemble solar panels: skylight, crosswalk, side of tall building.

For classification, images are simply grouped into *solar* and *no-solar* categories. Building our own dataset allowed us to determine the distribution of data in each class. As shown in table 1, both classes split the data 80% training and 20% validation. Many semantic segmentation projects are limited by the lack of sufficient labeled data, which can be difficult and expensive to obtain. Understanding the challenges and importance of labeled data, we allocated considerable time and resources towards annotating images. For semantic segmentation, we used LabelBox, a platform to manually annotate solar panels in images and generate segmentation masks. A review process was implemented to ensure that all solar panels are correctly identified and labeled. Figure 5 shows the total number of labeled images included in the segmentation dataset, which is equal to the number of images in the solar data in the classification dataset.

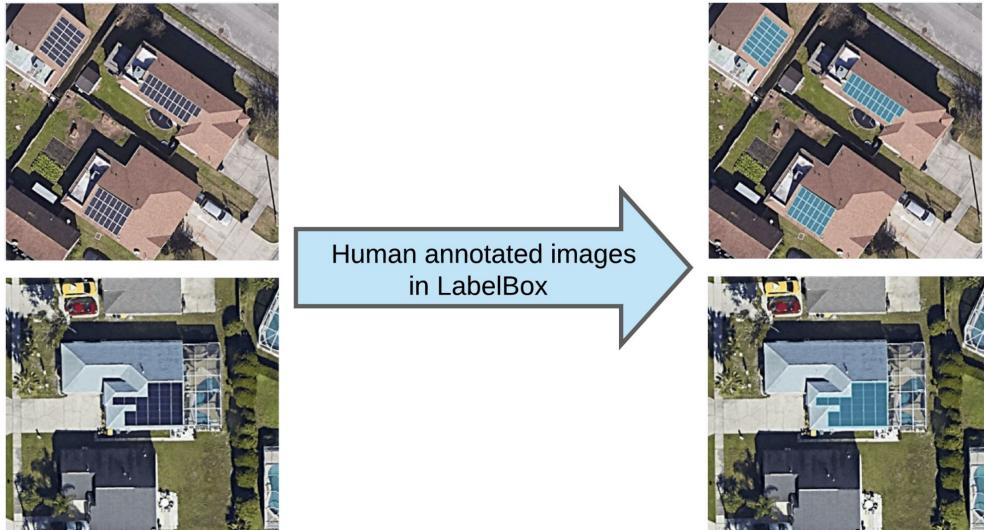


Figure 5: Examples of solar-panel annotations of images in LabelBox

In addition to our training and validation set, we also conducted an experiment on the city of Berkeley, CA. The Berkeley testset consists of random sampling of 10% of the image tiles

from the 8 council districts within the entire city. This sample consists of 2,243 images with a distribution of 1,922 no-solar and 321 solar class images. We use this testset for evaluating our individual models and the complete pipeline.

Classification Dataset			
	Solar	No-Solar	Total
Training	668	1295	1963
Validation	168	324	492
Berkeley Testset	321	1922	2243

Table 1: Classification dataset containing solar and no-solar images

Segmentation Dataset	
Training	668
Validation	168
Berkeley Testset	321

Table 2: Semantic Segmentation dataset containing solar panel images and their segmentation masks

4. Baseline Models

Classification

In order to prepare a baseline for the classification task, we created a simple neural network with a few Convolutional and Max-Pooling layer pairs, a fully connected layer, and a prediction layer. The model summary is provided in Appendix A. We trained this model using the classification training dataset. With the validation dataset, we got an accuracy score of **0.72401**.

Semantic Segmentation

As a baseline for the semantic segmentation task, we trained a U-Net model using the lightweight pre-trained model, **mobilenet-v2** as its encoder-backbone. With that, our segmentation validation dataset reported an IoU score of **0.8107**, and an F1-Score of **0.8841**.

5. HyperionSolarNet Procedures

5.1 Solar Panels Detection using Image Classification

In this work, we employ Transfer Learning and fine-tune a convolutional neural network called **EfficientNet-B7** to classify satellite image tiles into *solar* and *no-solar* classes. EfficientNet-B7 is a pre-trained model that achieves the state-of-the-art 84.4% top-1 and 97.1% top-5 accuracy on ImageNet with 66M parameters and 37B FLOPS.

We fine tune EfficientNet-B7 on a training set consisting of 668 image tiles containing solar panels, and 1295 image tiles without any solar panels. The performance of the fine-tuned model is evaluated against a validation dataset consisting of 168 image tiles containing solar panels, and 324 image tiles without any solar panels. The output of our model for a given image is logits, which we then normalize using the sigmoid function, that in turn produces a probability between 0 and 1 indicating whether the image belongs to no-solar or solar class. We use a threshold of 0.5 to convert the output probabilities to class labels - probabilities less than 0.5 are assigned label 0 (class no-solar), and greater than 0.5 are assigned label 1 (class solar).

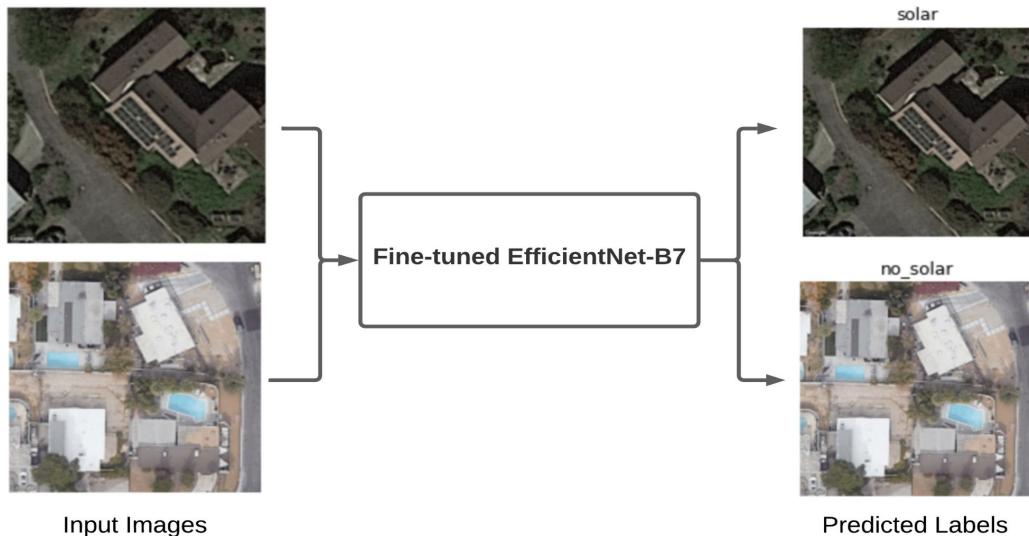


Figure 6: Examples of solar-panel annotations of images in LabelBox

In order to train EfficientNet-B7 for solar panels image classification, we replace its top layer with a Global Average Pooling layer and a fully connected prediction layer having a single output node. Besides training these two layers, we fine-tune the trainable variables in all of the layers of EfficientNet-B7. Refer to Appendix B for our classification model training plots and Appendix C for a summary of the classification model.

To increase variation and diversity of input images for model training, we augment the images by applying random horizontal flip, translation, rotation, contrast and cropping on them. Refer to Appendix D for example augmented images. We measure the performance of solar panels image classification using accuracy, precision and recall metrics. On the validation dataset, our classification model achieves an accuracy of **97.76%**, with a precision and recall of **95%** and **98%** respectively for the solar class.

Model	Accuracy	Precision	Recall	F1 Score
Baseline	0.7240	0.66	0.81	0.73
HyperionSolarNet Classification	0.9764	0.95	0.98	0.97

Table 3: Performance of the HyperionSolarNet classification model compared with the baseline model

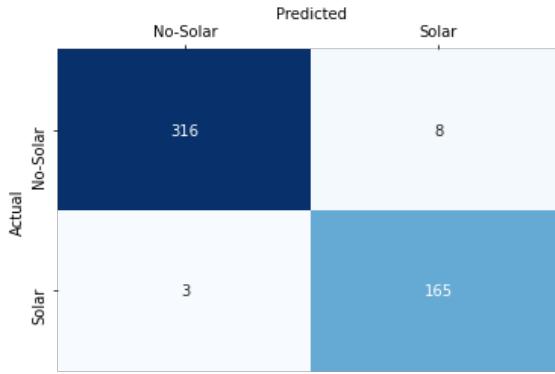


Figure 7: Classification model confusion matrix for the validation dataset

5.2 Solar Panels Size Estimation using Semantic Segmentation

For semantic image segmentation, we use **EfficientNet-B7** as the encoder-backbone to train a **U-Net** model for segmenting solar panels in satellite images.

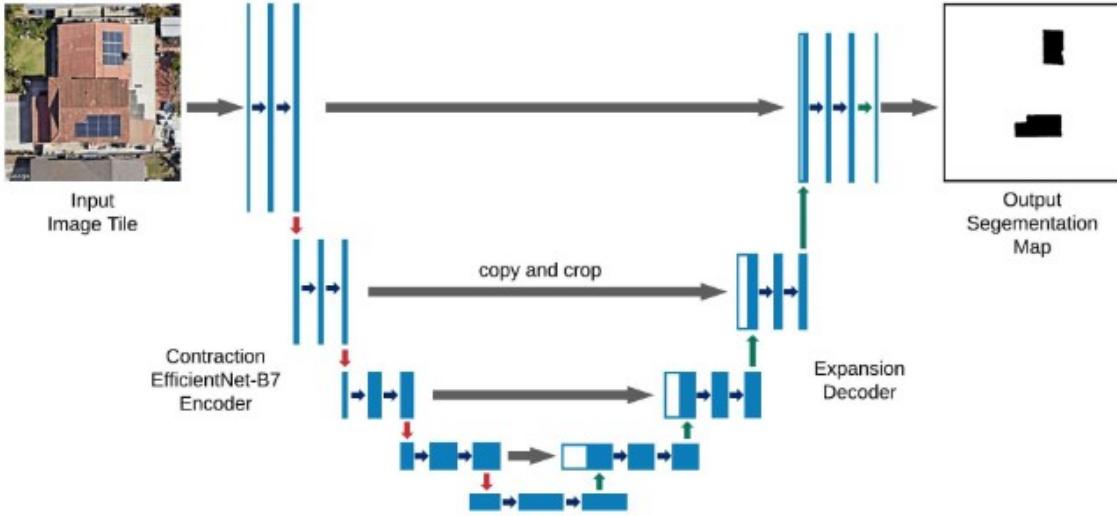


Figure 8: An example of U-Net architecture trained for segmenting images having solar-panels . An image having solar panels is passed through the network, and it produces a segmentation mask for the pixels belonging to solar panels in the image.

We annotated 836 images containing solar panels using the LabelBox platform, and produced their corresponding segmentation masks. Of those, 668 images and their masks are used as input training set for model training, and the remaining 168 images and masks are used as the validation dataset. Since we have images of different sizes in our training dataset, we resize them to a size of 512x512 pixels before feeding them to model training. The trained model produces segmentation masks of size 512x512 pixels. The semantic segmentation model training plots are provided in Appendix E and examples of the predicted mask images are shown in Appendix F. On the validation dataset, our model achieved an IoU score of **85.91%** and an F1 score of **91.97%** .

Model	IoU Score	F1 Score
Baseline	0.8107	0.8841
HyperionSolarNet Semantic Segmentation	0.8591	0.9197

Table 4: Performance of the HyperionSolarNet segmentation model compared with the baseline model

Augmentation with Albumentations

In order to bring variation in our training data images and to increase the dataset size, we augment the images and their segmentation masks. The augmentations include random cropping, flipping, changing brightness and blur, scaling, and grid and optical distortions.

tion of images. We use the Albumentations library [<https://github.com/albumentations-team/albumentations>] to perform the augmentations, which enabled us to apply the same augmentation to an image and its corresponding segmentation mask.

Segmentation Models Library

For training our semantic segmentation model, we use *segmentation_models* library. It offers a high level API for training segmentation models, while making 4 different model architectures and 25 backbones available for each architecture. Furthermore, it provides implementation of several segmentation losses (e.g. Jaccard, Dice, Focal) and metrics (e.g. IoU, F1-score), which are very helpful for segmentation model training. The library code is available here: https://github.com/qubvel/segmentation_models.

Procedure for Estimating Total Surface Area and Number of Solar Panels

The remaining step in the process is to generate an estimate of the total surface area of solar panels. This is accomplished by programming a function that accepts the output from the segmentation model and returns the total area of solar panels within a given image. The segmentation model produces a predicted mask for each image denoting the pixels that are classified as being a solar panel. This mask is a 512x512 matrix of ones and zeroes, with ones identifying the solar panel pixels. The predicted mask is then resized to the size of the original image, for example 600x600 pixels. This matrix is fed into the function along with the latitude and zoom of the image, which are used to derive the length of each individual pixel. Without diving into the math behind the equation, the length per pixel is calculated using the Mercator Projection, which accounts for the differences in length per pixel depending on the zoom and distance from the equator. Google Maps Static API provides the equation:

$$\text{meters_per_pixel} = 156543.03392 * \text{math.cos}(\text{latitude} * \text{math.pi} / 180) / \text{math.pow}(2, \text{zoom})$$

After converting to imperial units and determining the area per pixel in square feet, the total area of solar panels is calculated by multiplying the count of ones in the matrix by the area per pixel value. In turn, the number of solar panels is calculated by dividing the total solar panel area by 17.6 ft², the area of a standard PV panel.

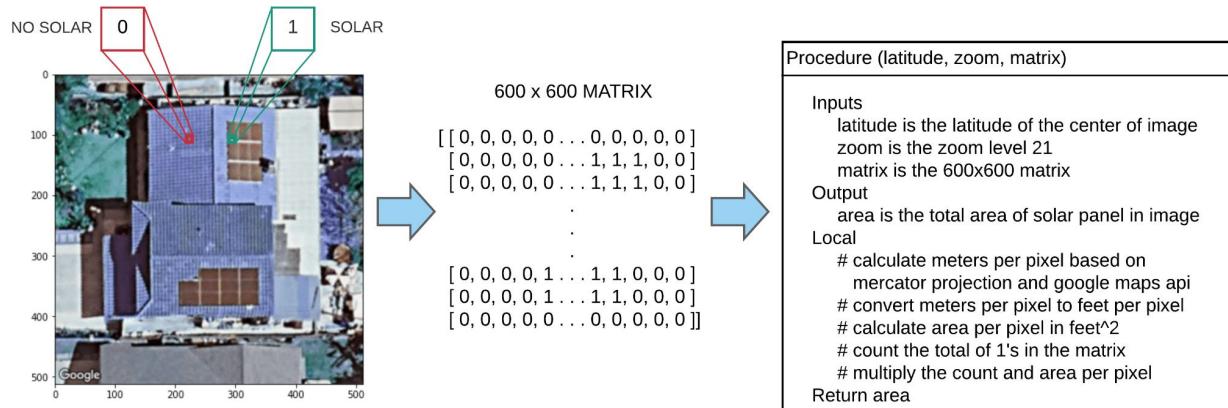


Figure 9: Procedure for calculating the area of solar panel in an image

6. Experimental Setup and Hyperparameters

We trained both of our models on a Google-Colab machine having **Tesla V100-SXM2-16GB GPU**, and on an AWS **g4dn.12xlarge** instance having **4 NVIDIA T4 GPUs**.

For training and fine-tuning our classification model, we use a starting learning rate of 1e-05 and reduce it by a factor of 0.1 as the validation accuracy reaches a plateau and does not improve for 10 epochs. The base model EfficientNet-b7 has 813 neural network layers. Using a batch size of 8 images, we fine-tune the trainable variables in all of the layers of EfficientNet-b7 for 150 epochs. We use RMSprop as the optimizer and Binary Cross-Entropy as the loss function for our training procedure.

For training the semantic segmentation model, we achieved the best performance using a learning rate of 1e-04 and the Adam optimizer. We reduce the learning rate by a factor of 0.1 as the validation IoU score reaches a plateau. We perform the model training for 150 epochs. For our training loss function, we use the Binary Cross-Entropy and the Jaccard loss.

7. Berkeley Testset Results

7.1 Classification Model Evaluation

The HyperionSolarNet classification model reports a mean accuracy of **0.9563** on the Berkeley testset. The following table shows the Precision, Recall and F1 scores for both the solar and no-solar classes.

Class	Precision	Recall	F1	Support
no-solar	0.98	0.97	0.98	1922
solar	0.82	0.91	0.86	321

Table 5: Performance of the HyperionSolarNet classification model against the testset.



Figure 10: Confusion Matrix of the Berkeley testset with the HyperionSolarNet classification model.

Figure 10 presents the confusion matrix we obtain by running the segmentation model against the Berkeley testset. We note that 63 no-solar class images out of 1922 images, and 30 solar images out of 321 were misclassified. To gain a better understanding of where the model is underperforming, we reviewed all the misclassified images. We present a few examples in Appendix G where the title of the images indicate their actual label, but our model classified as the opposite. As shown in the images, there are cases where the model still classifies false positives due to objects such as skylights. Many of the false negatives are due to the model having difficulty detecting PV panels located along the edge of images. As HyperionSolarNet is utilized in various locations, the misclassified images can be correctly labeled and utilized as training data to develop a robust model.

7.2 Segmentation Model Evaluation

In the Berkeley testset, we had 321 solar-panel images. We manually annotated those images using the LabelBox platform, and created mask labels for them. We evaluated the HyperionSolarNet segmentation model against these testset images, and the model reported an IoU score of **0.8243**. Table 6 presents the performance results.

	IoU	F1
Berkeley Testset Evaluation	0.8243	0.8922

Table 6: Berkeley testset results with HyperionSolarNet segmentation model

In order to understand the model performance better, we examined the predictions that had the IoU score less than 0.4. A few examples out of those 8 predictions are shown in Appendix H. We note that in these images, the solar-panels are not very clear and distinguishable, and it is difficult even for human eyes to identify solar panels in there.

7.3 Percentage Error in the Prediction of Size and Number of Solar Panels

Using the label masks of the annotated images in the testset, we calculate the total area and number of solar panels, which we treat as the actual size and number of panels in those images for the purpose of calculating the percentage error in the predictions. Additionally, we estimate/predict the area and number of panels by passing the testset images through our complete classification and segmentation pipeline. Finally, we compare the actual area and the number of panels against the predictions of our pipeline to determine the percent error in prediction for the testset using the following formula. Our architecture pipeline makes an error of 0.7% in predicting the number and size of solar panels in the Berkeley testset.

$$\text{Percent Error} = \frac{|\text{Actual} - \text{Predicted}|}{\text{Actual}} \times 100$$

	Area (sq. ft.)	Number of Solar Panels
Actual	101,765.48	5,787
Predicted	102,609.72	5,828

Table 7: Actual and predicted area and number of solar panels

7.4 Estimated Solar Panels in Berkeley

We estimated the size and number of solar panels using all the image tiles of the Berkeley city. In the following figure, we share the results.

	Area (sq. ft.)	Number of Solar Panels
Estimated Berkeley Solar Panels	1,082,431.98	61,480

Table 8: Estimated number of solar panels and total area

8. Web Application

We built a web-application that provides users an interactive tool to visualize the output from the classification and segmentation models. The interface allows users to outline a desired geographical area on Google Maps. With the click of a button, HyperionSolarNet splits the region into smaller 600x600 pixel tiles, identifies the latitude and longitude coordinates of each tile containing solar panels, and estimates the total surface area of solar panels. In cases where the desired location is a city boundary or larger, we use a GeoJSON file with the boundary coordinates to preprocess all of the solar panel tiles, locations and surface area within the region.

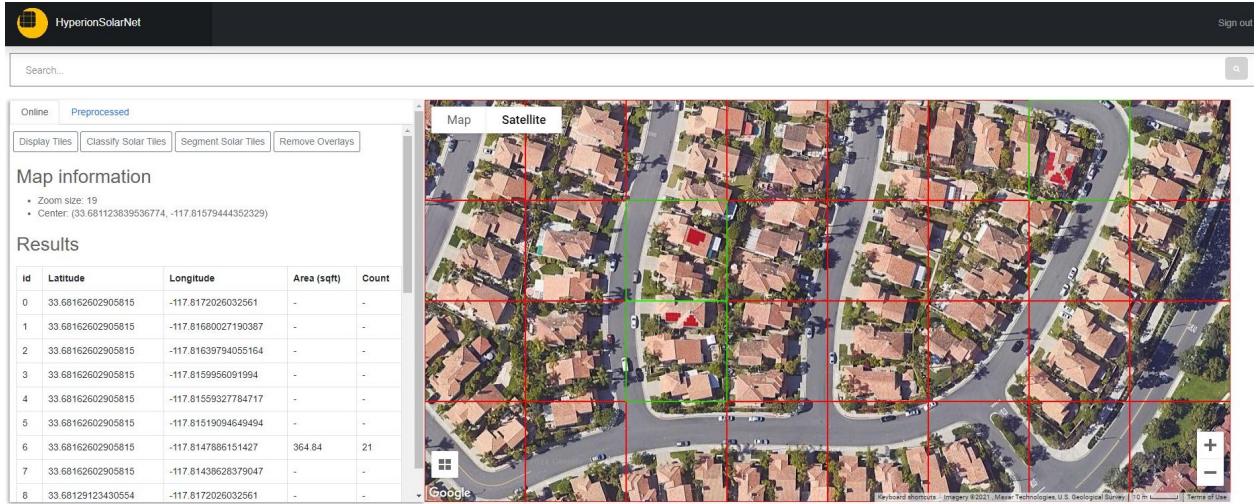


Figure 11: Web-application at <http://www.hyperionsolarnet.com/>

Our application architecture is built on AWS, using EC2 to host our website and S3 to save all static content including models, masked images and GeoJSON files. The website is hosted on a g4dn.xlarge EC2 with T4 GPU and 16 Gb of memory, which is cost effective for inference. The web application is built using Flask, Nginx as a reverse proxy, and Gunicorn as the application server configured to use 3 workers. To achieve inference completely independent from the Flask application, we configure TensorFlow Serving, which provides a server to expose model inference through an API. For our web application on inference, Flask is used as a proxy that connects by calling an API via post to send data to the TensorFlow Serving API which delivers prediction results. As input for inference, we require images for classification and segmentation. The client sends latitude and longitude, height, width and zoom of the map section. Using this information, we use the Google Map API to generate equal size image tiles to use as input for inference, which are converted to numpy arrays. The application returns the classification information for each tile and the segmentation masks overlaid on the image tiles.

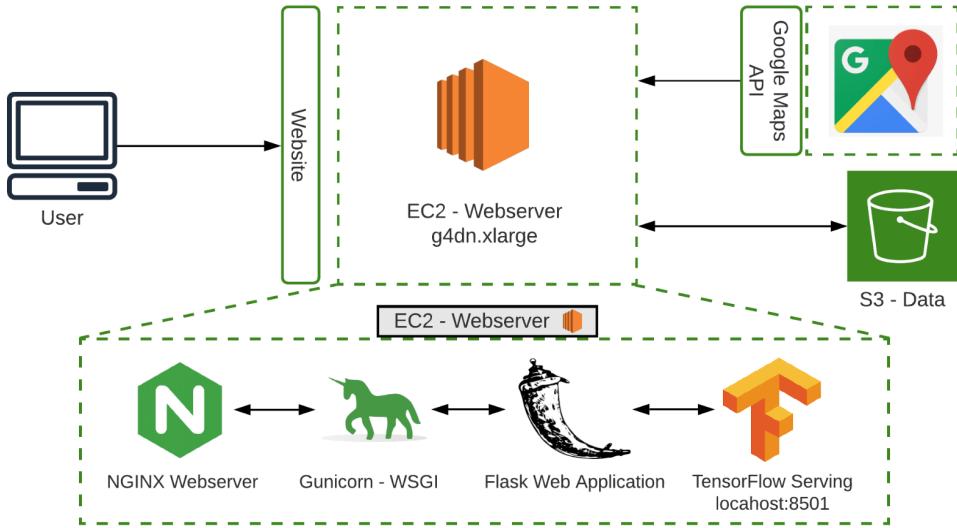


Figure 12: Diagram of HyperionSolarNet system architecture

The application has a few limitations mainly due to memory and costs for GPU instances. Due to memory restrictions our minimum zoom capacity for online processing is 19. To reduce the memory usage, we have tested model optimization using TensorRT with the same restrictions. Future enhancements include improving inference performance by testing new optimizations with TensorRT and TensorFlow and simplifying the model without sacrificing performance.

9. Conclusion

As the effects of climate change continue to negatively impact the world, there is a heightened urgency to reduce greenhouse gas emissions by fully integrating solar power into energy grids. One of the keys to implementing solar energy is to develop a comprehensive database of the locations and total surface area of solar panels within a given geographical region. Our work provides solutions to this challenge by leveraging deep learning methods to detect solar panels from aerial images. HyperionSolarNet implements a two-branch model using an EfficientNet-B7 classifier in tandem with a U-Net and EfficientNet-B7 encoder-backbone semantic segmentation model to predict solar panel location and surface area. The use of this two-branch model results in improved performance while using a relatively smaller training dataset. On the application side, our work includes a user-interface that provides a tool that helps visualize the output from the classification and segmentation model using Google Maps API. While the models presented in this work perform reasonably well, the application of our work can produce additional images that can be used for training. Our future research work includes applying HyperionSolarNet to diverse locations across the world to produce additional classified and masked images that can be utilized for further improvement of the model.

Acknowledgements

We thank Alberto Todeschini, Puya Vahabi, Colorado Reed, and our UC Berkeley colleagues for their valuable advice and guidance on this work.

References

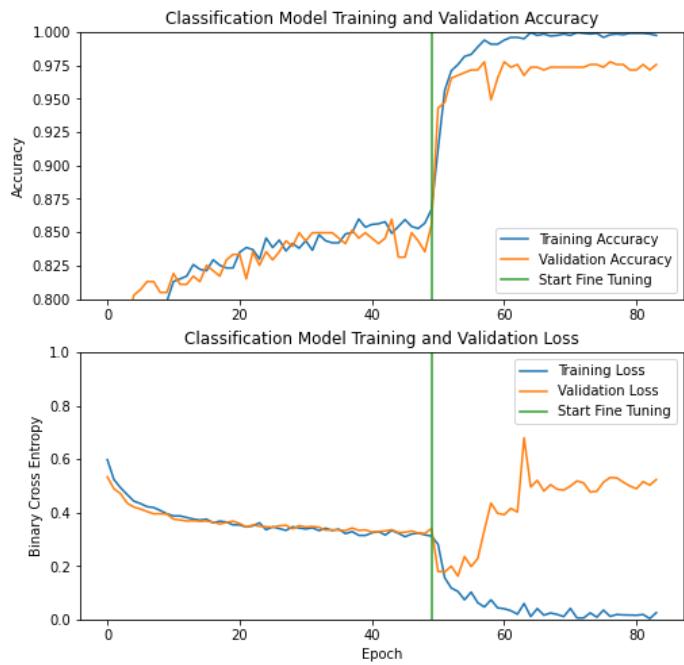
- [1] Johan Mathe, Nina Miolane, Nicolas Sebastien, and Jeremie Lequeux. 2020. PVNet: A LRCN Architecture for Spatio-Temporal Photovoltaic Power Forecasting from Numerical Weather Prediction. [arXiv:1902.01453](https://arxiv.org/abs/1902.01453).
- [2] Galen Barbose and Naim Darghouth. 2019. Tracking the Sun: Pricing and Design Trends for Distributed Photovoltaic Systems in the United States. Berkeley Lab lbl.gov.
- [3] Jiafan Yu, Zhecheng Wang, Arun Majumdar, and Ram Rajagopal. 2018. DeepSolar: A Machine Learning Framework to Efficiently Construct a Solar Deployment Database in the United States. <https://doi.org/10.1016/j.joule.2018.11.021>.
- [4] Joseph Camilo, Rui Wang, Leslie M. Collins, Kyle Bradbury, and Jordan M. Malof. 2018. Application of a semantic segmentation convolutional neural network for accurate automatic detection and mapping of solar photovoltaic arrays in aerial imagery. [arXiv:1801.04018](https://arxiv.org/abs/1801.04018)
- [5] Li Zhuangm, Zijun Zhang, and Long Wang. 2020. The automatic segmentation of residential solar panels based on satellite images: A cross learning driven U-Net method. <https://doi.org/10.1016/j.asoc.2020.106283>.
- [6] M. Arif Wani and Tahir Mujtaba. 2021. Segmentation of Satellite Images of Solar Panels Using Fast Deep Learning Model. <https://www.ijrer.org/ijrer/index.php/ijrer/article/view/11607>.
- [7] Vladimir Golovko, Alexander Kroshchanka, Egor Mikhno, Myroslav Komar, and Anatoliy Sachenko. 2021. Deep Convolutional Neural Network for Detection of Solar Panels . https://doi.org/10.1007/978-3-030-43070-2_17.
- [8] Mingxing Tan and Quoc V. Le. 2019. Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks. [arXiv:1905.11946](https://arxiv.org/abs/1905.11946).
- [9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional Networks for Biomedical Image Segmentation. [arXiv:1505.04597](https://arxiv.org/abs/1505.04597).

Appendix

A. Summary of the baseline classification model

Layer (type)	Output Shape	Param #
<hr/>		
rescaling_3 (Rescaling)	(None, 600, 600, 3)	0
conv2d_4 (Conv2D)	(None, 600, 600, 16)	448
max_pooling2d_4 (MaxPooling2D)	(None, 300, 300, 16)	0
conv2d_5 (Conv2D)	(None, 300, 300, 32)	4640
max_pooling2d_5 (MaxPooling2D)	(None, 150, 150, 32)	0
conv2d_6 (Conv2D)	(None, 150, 150, 64)	18496
max_pooling2d_6 (MaxPooling2D)	(None, 75, 75, 64)	0
conv2d_7 (Conv2D)	(None, 75, 75, 64)	36928
max_pooling2d_7 (MaxPooling2D)	(None, 37, 37, 64)	0
dropout_2 (Dropout)	(None, 37, 37, 64)	0
flatten_1 (Flatten)	(None, 87616)	0
dense_4 (Dense)	(None, 512)	44859904
dense_5 (Dense)	(None, 1)	513
<hr/>		

B. HyperionSolarNet classification model training plots



C. HyperionSolarNet image classification model summary

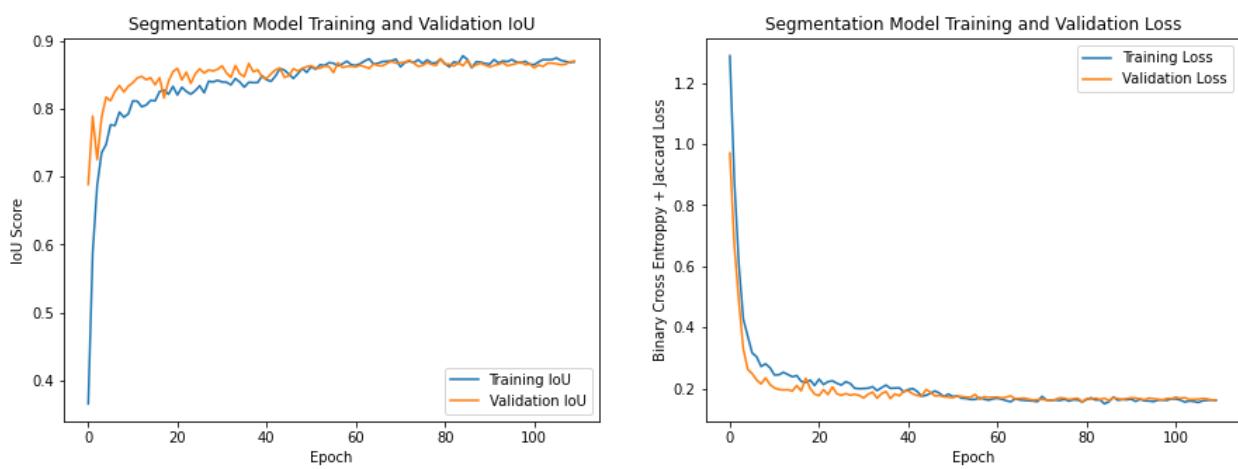
Layer (type)	Output Shape	Param #
input_8 (InputLayer)	[(None, 600, 600, 3)]	0
sequential_1 (Sequential)	(None, 600, 600, 3)	0
efficientnetb7 (Functional)	(None, 19, 19, 2560)	64097687
global_average_pooling2d_3 (GlobalAveragePooling2D)	(None, 2560)	0
dropout_3 (Dropout)	(None, 2560)	0
dense_3 (Dense)	(None, 1)	2561

Total params: 64,100,248
 Trainable params: 54,339,801
 Non-trainable params: 9,760,447

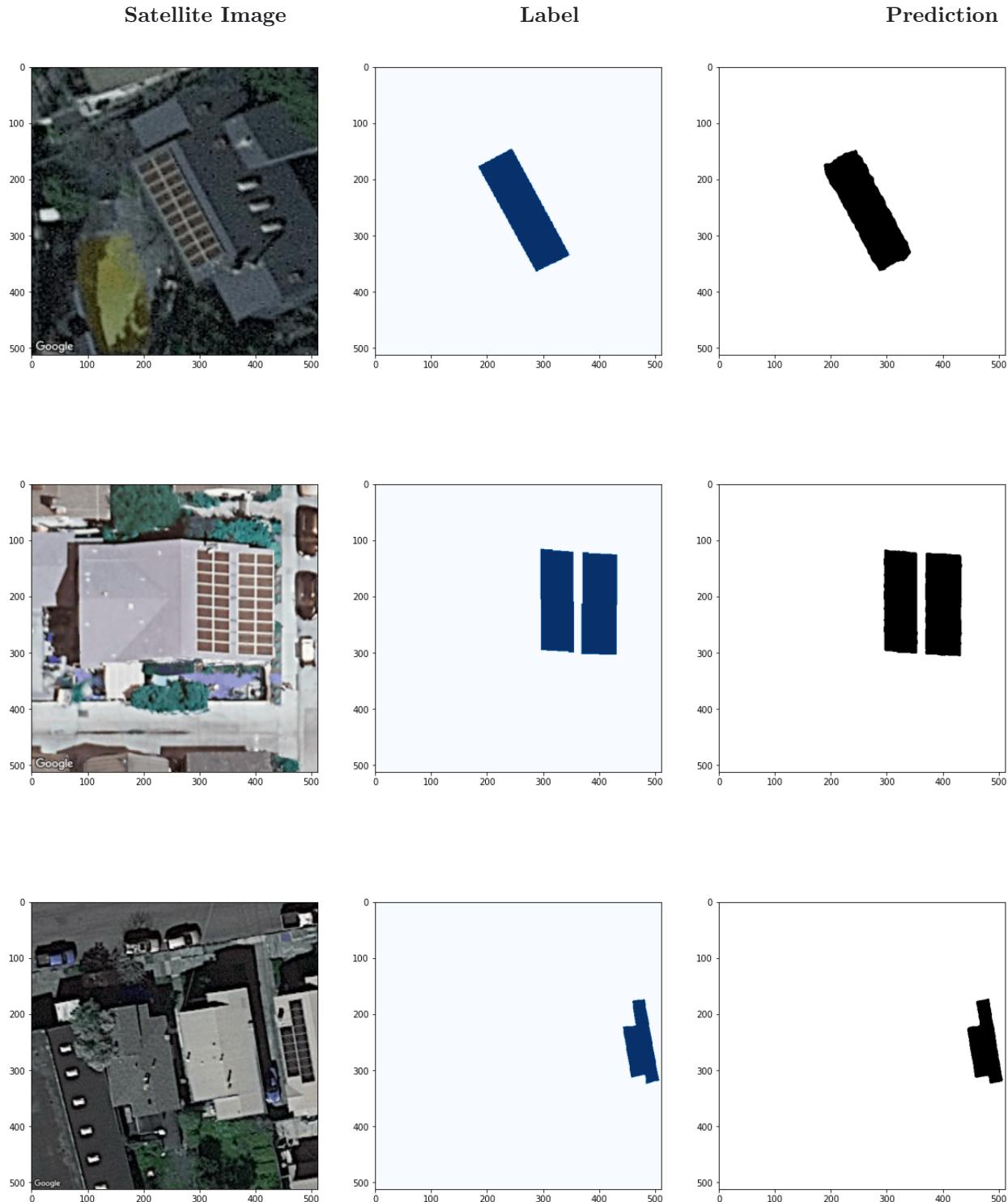
D. Examples of augmentation on input training images

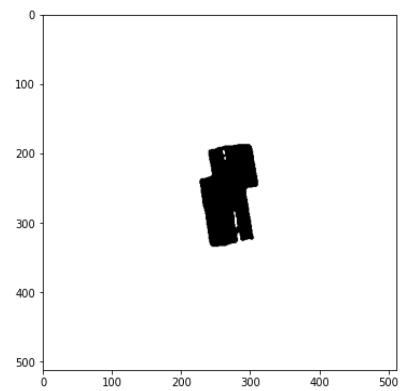
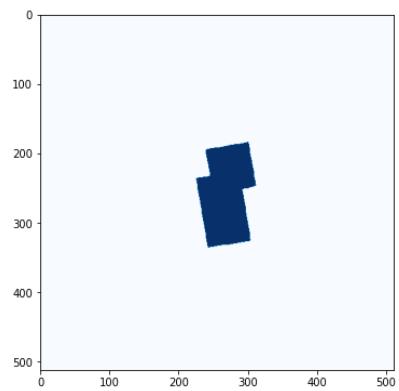
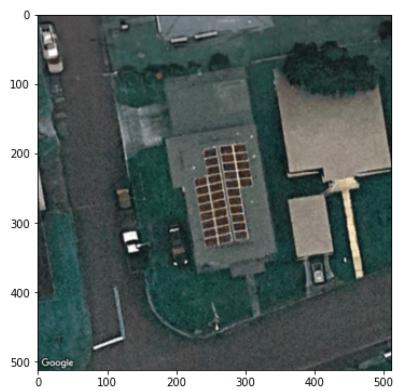
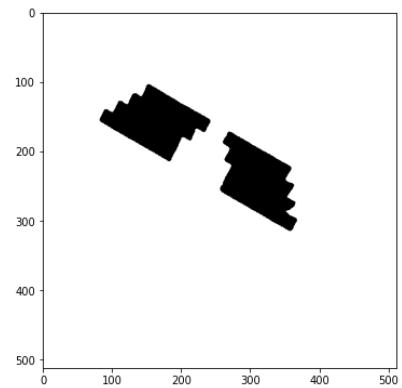
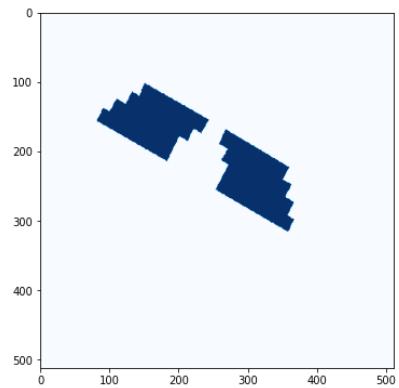
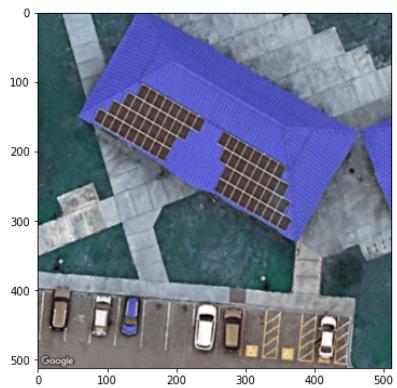


E. HyperionSolarNet semantic segmentation model training plots



F. Examples of segmentation mask predictions from HyperionSolarNet, along with the original images and labels.





G. Examples of images misclassified by HyperionSolarNet. The title above the image denotes the actual class, but our model classified as the opposite.



H. Examples of segmentation predictions with an IoU score < 0.4 from the Berkeley testset

