Detecting Solar PVs in Aerial Imagery

Kaggle Competition Report
IDS 705 Spring 2020
Sang-Jyh Lin, Vanessa Tang, Sangseok Lee, Julio Portella, Abdur Rehman

---

Abstract

Increasing attention to renewable energy has led to the growing popularity of solar panels (*EIA - Electricity Data*, 2019) in addition to the desire to gather more information about solar panel usage. Given a dataset of aerial images, the purpose of this study is to develop machine learning models to identify solar panels in images. To do so, image preprocessing and feature extraction techniques like the histogram of oriented gradients (HOG) were used in conjunction with a support vector machine (SVM) and convolutional neural network (CNN). Using area under the curve (AUC) to compare model performance, we found that a CNN performed the best with an AUC of 0.991 while the SVM with HOG had an AUC of 0.831. This shows that CNNs specifically could be useful in identifying solar PVs in aerial imagery to provide more accurate information on solar panel usage.

Introduction

In recent years, there has been a growing adoption of renewable energy in the United States. Solar photovoltaic panels, or solar PVs, are being added every month to the roofs of commercial buildings and private properties (*EIA - Electricity Data*, 2019). Traditionally, solar PVs were connected to a part of the electric grid called the distribution grid which delivers energy "one-way" from power plants to end users. With solar PVs, there is an opportunity for a "two-way" flow of electricity on distribution grids (Rolnick *et al.*, 2019). To add solar PVs to their existing power grids, energy operators such as utilities and government agencies need detailed information about the location of solar PVs, their power capacity and energy production (Malof *et al.*, 2016). In the past, energy operators have relied on collecting information through surveys and utility interconnection filings. However, these methods are very costly, time-consuming, and limited in their spatial resolution (Malof *et al*., 2016). Given the rapid growth of solar PVs, there is a strong need to find methods that are more efficient, timely and also at a high resolution, such as county, zip code or neighborhood.

One alternative proposed to traditional methods is the use of machine learning algorithms. Once developed, algorithms can be very efficient to deploy and much cheaper and less time-consuming than surveys and utility interconnection filings. In addition, they can process images at a much higher spatial resolution, so can be operationally efficient when getting integrated into existing power systems (Malof *et al*., 2015).

In this study, we will show how machine learning algorithms can be used to detect solar PVs. We will use two approaches, namely a support vector machine (SVM) and a convolutional neural network (CNN), on satellite images, and compare their performance using area under the

curve (AUC). We will conclude by summarizing our results and suggesting areas for future research.

Background

Our work was informed by existing research that uses machine learning to interpret high-resolution imagery. In such applications, aerial imagery is usually formulated as a pixel labelling task. The goal is to produce either a semantic segmentation of the image into distinct object classes such as building, road, tree, grass, water and vehicles (Dollar *et al*., 2006; Kluckner *et al*., 2009; Saito *et al*., 2016) or a binary classification of the image into a single object class (Mnih & Hinton, 2010). Compared to binary classification, semantic segmentation models have achieved superior performance for both detecting solar PVs as well as estimating their size (Camilo *et al*., 2017). However, to simplify our analysis, we limit ourselves to the binary classification task of detecting solar PVs in aerial imagery.

With a binary classification task, our aim was to evaluate the performance of a neural network by comparing it with a non-neural network. Malof *et al*. (2015) showed how a support vector machine (SVM) could be used to detect solar PVs with a high accuracy on a small dataset of 53 hand-annotated PV images with a combined area of < 1 km^2. Malof *et al*. (2016) also showed how a more complex algorithm, a random forest (RF), could improve performance on a larger dataset of 60 hand-annotated PV images with a combined area of 135km^2. Malof *et al*. (2017) showed how a neural network offers substantial performance improvement on 16,000 hand-annotated PV images with a combined area of 400km^2. In our study, we compare the performance of a neural network with that of a non-neural network on over 1,500 hand-annotated aerial images.

To extract features, we considered the advice of computer vision experts. Minh (2013) explains why context is important in detecting objects in aerial imagery. Context is important for successfully labelling images because color hues are not sufficient for discriminating between similar pairs of object classes. For instance, solar PVs can appear to be part of a dark roof, so discriminating based on color alone is not sufficient. In addition, occlusions and shadows can make different parts of the roof appear more similar than they are. We also considered successful features of object detectors of buildings, roads and vehicles (Gleason *et al*., 2011; Hutchison *et al*., 2010; Mnih & Hinton, 2010; Youssef *et al*., 2014). Overall, our aim was to use image preprocessing and feature extraction techniques in combination with machine learning models like SVMs and CNNs to identify solar panels in aerial images.

Data

The provided dataset is composed of 1500 images for training and 588 images for testing. All images are in color and saved in .tif format. Each image is 101 pixels by 101 pixels, leading to a training dataset of (1500, 101, 101, 3) for 1500 images of 101 by 101 pixels with three channels denoting red, green, and blue (RGB) channels for each image. Training labels are also

provided to assess model performance. Within the training data, there are 505 images with solar panels (33.67%) and 995 images without solar panels (66.33%). The test data follows a similar format for the 588 images, with the exception of test labels. Within these datasets, image identification numbers are also provided, which is useful in indexing the images and can be used to individually view misclassified images when gauging model performance.

Within the training and test datasets, all images are satellite views, and most images contain satellite views of homes with or without solar panel arrays installed. However, there are some other types of images, consisting of cleared land or non-house views as shown in Figure 1. Given that solar panels are produced with different material compositions and designs related to the manufacturer, they cannot be classified easily by color. However, solar panels have a distinctive rectangular form that can be seen from above, but rooftop installation often consists of clusters of solar panels in a variety of different shapes. In addition to the irregular panel array arrangement, other objects, such as trees, may alter the view of a solar panel. Other challenges include solar panel orientation, rooftop angle, shadows, or other confounding objects that can make solar panel detection difficult. Image preprocessing and feature extraction can help address these challenges in identifying solar panel arrays.

(a) (b) (c)

Figure 1. Different aerial views for (a) clear land, (b) a roof without solar panel, and (c) a roof with solar panels.

Methods

For this study, a neural network and a non-neural network-based model were developed. The performance of the models was evaluated on AUC. Because of the high-dimensionality of the image data, we focused on support vector machines (SVM) as our non-neural network model and aimed to increase AUC through image preprocessing and feature extraction. Commonly used in image classification problems, we used a convolutional neural network (CNN) to predict solar panels in images, expecting the CNN to perform better than the SVM. In both the SVM and CNN, selection of ideal (hyper)parameters was very important, and throughout the process, selecting the best parameters relied on comparison of AUCs where the final model for each the SVM and CNN was chosen based on the highest AUC. Figure 2 details the general flowchart in selecting the two highest performing models. Comparison between kNN and SVM justifies selection of the SVM as the final non-neural network model, and the SVM then becomes the

baseline model for comparison to the CNN. Figure 2 also shows the importance of (hyper)parameter tuning in finding the optimal AUC on the validation data.

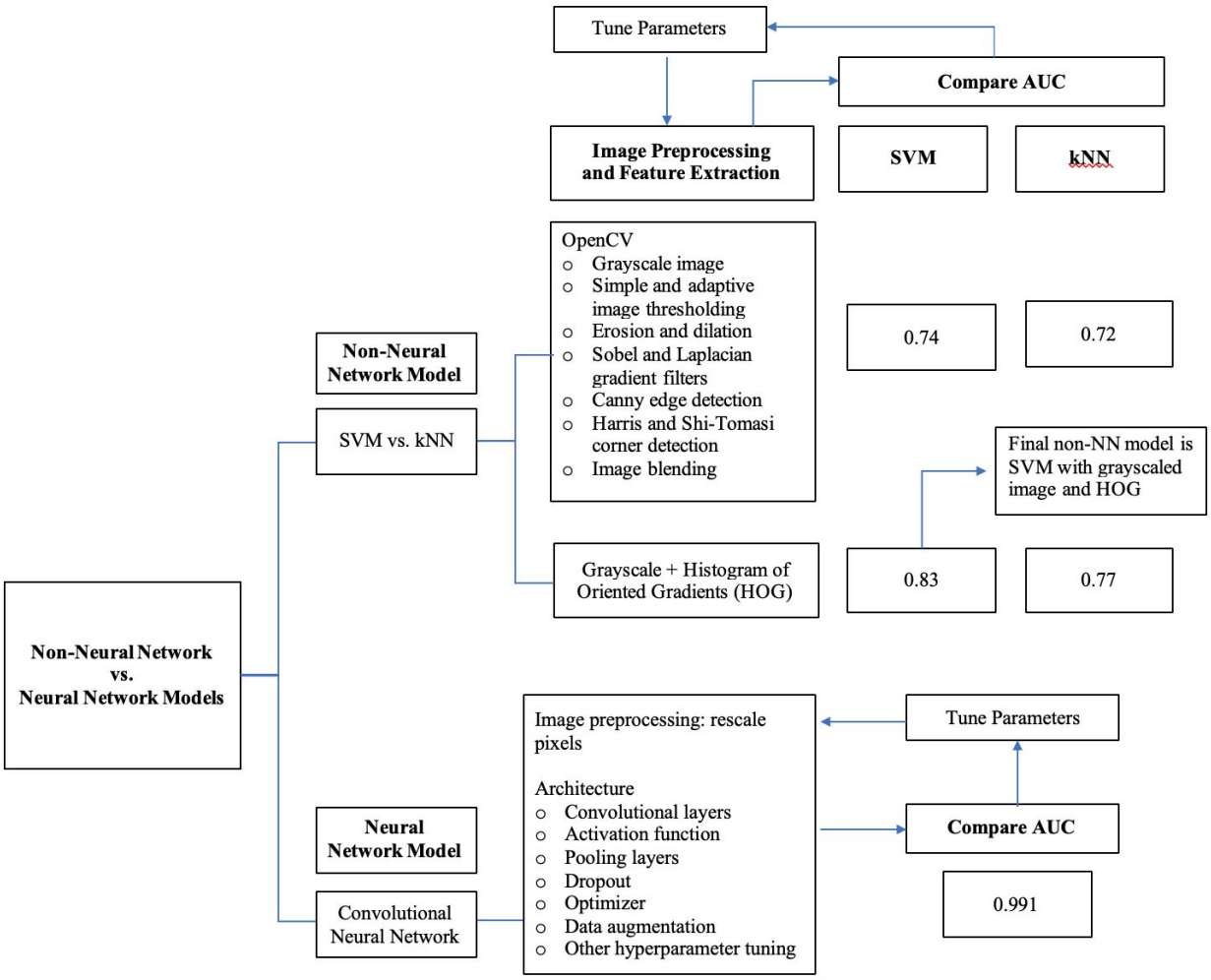


Figure 2. Flowchart of methodology in developing neural network and non-neural network models. This included image preprocessing and feature extraction in addition to AUC comparison to select the highest performing model on validation data.

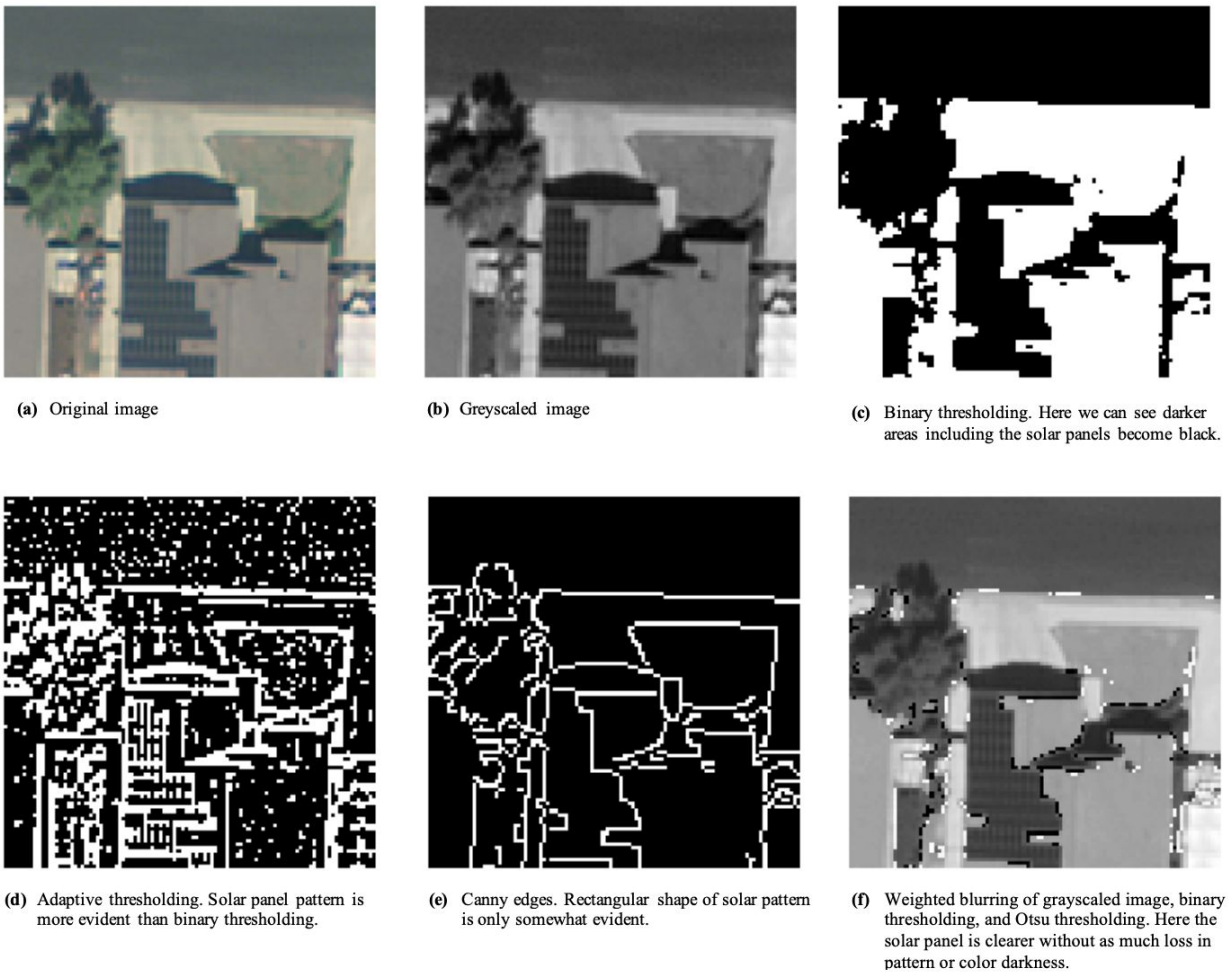
Support Vector Machine

Image Preprocessing and Feature Extraction

In working with the images, we tested a variety of preprocessing and feature extraction techniques, with the goal of making the solar panels more easily identifiable through preprocessing. First, we started with gray scaling the image and adding the mean, standard deviation, and various quantile values as different features to help extract features that could be

relevant to identifying the solar panels. Because the solar panels often appeared darker in the images, we also added the maximum gray-scaled color pixel value as a feature. Though this helped us get a better sense of the data, it did not significantly improve model performance or increase the area under the curve (AUC).

After testing these basic preprocessing techniques, we tried a variety of techniques from OpenCV, such as simple and adaptive image thresholding, erosion and dilation, Sobel and Laplacian gradient filters, canny edge detection, Harris and Shi-Tomasi corner detection, and blending to test different combinations of these preprocessing techniques (Figure 3). These preprocessing techniques were tested with both k-Nearest Neighbors (kNN) and support vector machine (SVM), generating AUC values of about 0.73 and 0.74, respectively.



(a) Original image

(b) Greyscaled image

(c) Binary thresholding. Here we can see darker areas including the solar panels become black.

(d) Adaptive thresholding. Solar panel pattern is more evident than binary thresholding.

(e) Canny edges. Rectangular shape of solar pattern is only somewhat evident.

(f) Weighted blurring of grayscaled image, binary thresholding, and Otsu thresholding. Here the solar panel is clearer without as much loss in pattern or color darkness.

Figure 3. Sample image outputs from various preprocessing techniques from OpenCV with the goal of making solar panels more easily identifiable.

In addition to image preprocessing with OpenCV, we also implemented the histogram of oriented gradients (HOG) (Figure 4). To calculate HOG for a given image, the directions of gradients are calculated for each cell and the final output consists of the calculated directions and magnitudes. Because HOG views images in smaller cell segments, it often helps deal with noise in image while reducing the dimensionality into 2 channels for gradient direction and magnitude.

Furthermore, HOG is particularly useful in identifying edges in an image, which is helpful in our case because solar panels have a clear and consistent rectangular shape.

(a) Original image (b) Greyscaled image (c) HOG. The HOG output shows edges and areas with large changes in intensity.

Figure 4. By comparing the original image to the HOG output, we can see how the gradient directions and magnitudes are represented.

After testing kNN and SVM with HOG, model performance increased to about 0.77 and 0.83, respectively. We also tested HOG with images preprocessed using various combinations of techniques from OpenCV with no improvement in AUC. Therefore, our best performing non-neural network model was SVM with HOG.

Classifier Selection and Parameter Tuning

In classification, SVMs find the optimal hyperplane that best separates the classes, which would mean separating the images with and without solar panels for this study. SVMs can also handle non-linear class separations better than other classification methods, as SVMs can be more flexible by changing the kernel (James *et al*., 2013). Previously shown to be successful in handwritten digit recognition, face detection, and object recognition, they have gained increasing attention for their ability to classify images (Hermes *et al*., 1999). Furthermore, because of the high dimensionality of the image data, SVMs were chosen as the main non-neural network model, as SVMs have been shown to handle large amounts of data well while being more flexible than other classification models (Hermes *et al*., 1999). Overall, SVMs are advantageous in image classification because they can handle nonlinear class boundaries and high-dimensional data while being flexible enough to yield high-performing models that can often generalize well in comparison to other more inflexible classifiers like logistic regression.

Though other classification techniques were tested in classifying solar panels, such as k-Nearest Neighbors and logistic regression, SVMs performed the best in terms of AUC. This observation is consistent with that of Hermes *et al*. on land usage classification where SVMs outperformed Gaussian and cluster-based maximum likelihood classifiers as well as kNN (Hermes *et al*., 1999). Therefore, after testing different image preprocessing and feature extraction techniques, we then focused on optimizing our SVM model to produce the highest AUC.

To measure generalization performance, the provided training data was split into a 70%

training set and 30% validation set, and all models were trained on the same 70% of the provided training data. Performance was measured by comparing validation set AUC values for different models. To select the best performing model, the model with specific image preprocessing and feature extraction techniques with the highest AUC was chosen. Though various preprocessing techniques were used, after thorough testing of different combinations, the model with the highest AUC was an SVM with greyscaled images and HOG.

However, in order to find the best performing model, we also needed to tune the parameters, especially for HOG and SVM. The purpose of implementing HOG was to generate gradient orientations within each image, which could theoretically help with solar panel detection by highlighting the shape of the solar panels. After testing different parameter values, we found that a 16-by-16 pixel cell with 2-by-2 cells per block with block normalization resulted in the best performance with both the SVM and the kNN models. Because the SVM performed better with this high-dimensional data, we then further turned the SVM parameters and found that a regularization parameter, *C* (squared l2 penalty), of 10 with the previously detailed HOG parameters produced the highest AUC of about 0.83 on the test data.

Convolutional Neural Network

Because of the need for high accuracy and fast predicting speed, image identifying applications, including satellite image identification, vehicle detection, computed tomography (CT) examinations, and skin cancer recognition, often apply convolutional neural network (CNN) (Kamsing *et al.,* 2019; Bragilevsky *et al.,* 2017; Wiranata *et al*., 2018; Ben-Cohen *et al*., 2018; Ly *et al*., 2018). To build a good CNN model, we need to consider the architecture of our CNN based on our data and problem. Our network architecture is shown in Figure 5. In summary, we applied 5 convolutional layers, 5 pooling layers and 2 fully connected layers. The total number of parameters is over 660,000. The details of different layers, functions, and the hyperparameter settings will be further discussed later.

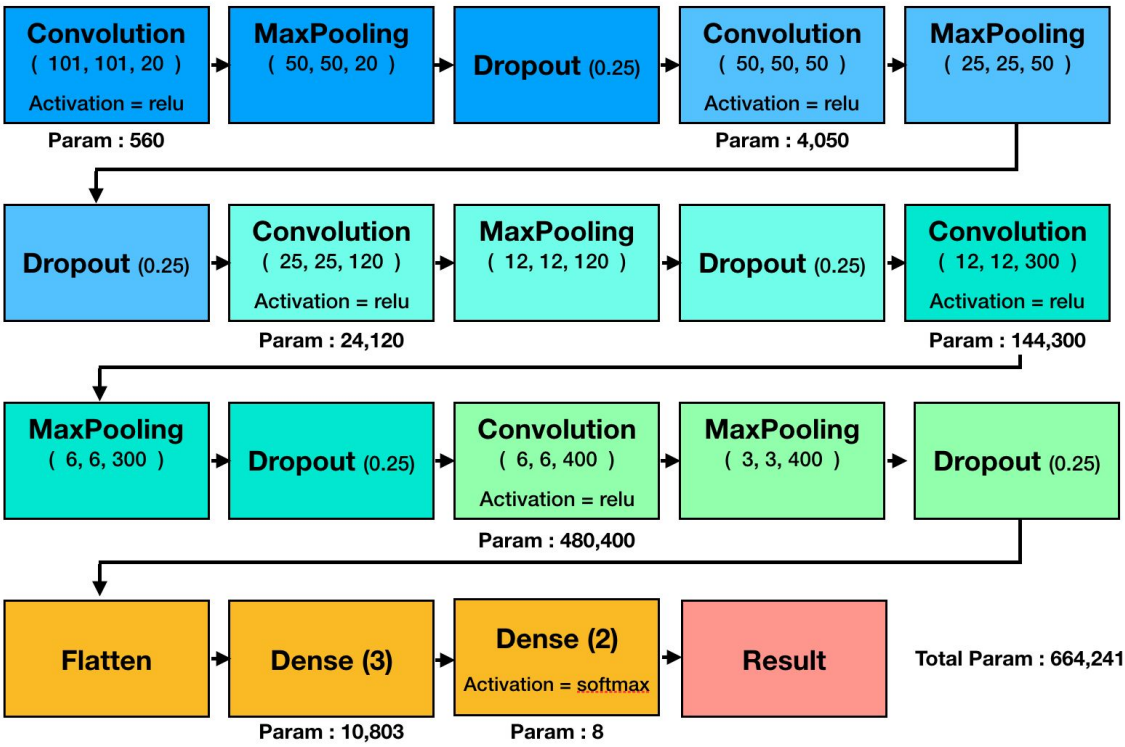


Figure 5. Architecture and Summary of Convolutional Neural Network: Each block represents a layer. The parentheses under the layer name is the output size of that layer. Under the parentheses is the chosen activation function for that layer. Param means the number of parameters in that layer. For example, there are 560 parameters in the first convolutional layer.

Data preprocessing

Although we do not need to normalize our data if we use rectified linear unit (ReLu) as the activation function in CNN, research results show that normalization can improve optimization and thereby produce better model performance while avoiding overfitting. (Krizhevsky *et al*., 2012; Ly *et al*., 2018) Therefore, we rescaled our data from range [0,255] to range [0,1]. After this data transformation, our networks obtained over 0.85 AUC with only one convolutional layer.

As shown in Figure 5, we have around 660,000 parameters in our network, but only 1500 total images are in the training data. This is insufficient in learning high numbers of parameters without overfitting. Therefore, we apply data augmentation to combat overfitting. There are many ways to achieve data augmentation. The easiest and most common method is to randomly rotate, shift, flip, crop, and sheer the original image dataset by creating an image generating function. Since these augmentations only slightly increase computational cost, as these generated images do not need to be stored on disk, we do not sacrifice computation ability or necessitate choosing a smaller network (Scott *et al*., 2017).

Since training CNN is time consuming, we did not apply cross validation for CNN training. Instead, we split our training dataset into 70% training data and 30% validation data.

Convolutional layer

The convolutional layer is the core building block of CNN. It has a small receptive field in the input image and computes the output by convolution of the receptive field with a linear filter. In general, the number of filters and feature detectors in each convolutional layer should be twice that of the previous convolutional layer. Higher the number of filters lead to higher the number of extractions, and enable the network to extract more features from image data. Hence, in order to reduce the number of parameters to avoid overfitting, we set only 20 filters for our first convolutional layer, and 400 for the fifth convolutional layer, which should be 600 if the doubling rule was followed. Moreover, our model testing supports this concern, as a CNN with 500 filters in the fifth convolutional layer resulted in lower AUC performance. Overall, finding the optimal number of filters for each convolutional layer was crucial in developing a high-performing CNN.

Another hyperparameter that we adjusted is the padding function, applying the same-padding scheme in our final CNN model. This was done because same-padding can help prevent information loss (Wiranata *et al*., 2018).

Activation function

The selection of activation function will affect model classification performance (Ertam *et al*., 2017; Zhao *et al*., 2018). Activation functions include ReLu, exponential linear unit, hyperbolic tangent (TanH), sigmoid, SoftPlus, Softsign, and Softmax (Ertam *et al*., 2017; Zhao *et al*., 2018; Ide et al, 2017). Among these functions, recent research shows that developing deeper or wider architecture applied ReLU to improve model performance (Szegedy *et al*., 2015; Simonyan and Zisserman, 2015; He *et al*., 2016; Zagoruyko *et al*., 2016; Zeiler and Fergus, 2014). In addition, Glorot *et al*. (2009) also advocated that ReLu can solve the gradient problems that other saturated activation functions, such as sigmoid or tanh, often suffer from. Another commonly used function is Softmax. Softmax is usually used in the dense layer, since we can classify our data directly if we use this function (Salakhutdinov *et al*., 2009; Zhao *et al*., 2018). Therefore, we chose Softmax for the last dense layer.

Pooling layer

Pooling layers in CNN summarize the outputs of neighboring groups of neurons in the same kernel map. These neighboring groups are summarized by adjacent pooling units (Cire *et al*., 2012; Jarrett *et al*., 2009; LeCun *et al*., 2010). We applied max pooling in our final model to reduce the complexity of our final CNN. Furthermore, max pooling is a widely used pooling method and has been successful in improving model performance in prior studies (Zhai *et al*., 2017; Chou *et al*., 2019; Krizhevsky *et al*., 2012).

Optimizer

Optimizers play an important role in minimizing loss during the training process. In general, Adaptive Moment Estimation (Adam) has the best performance when compared to other types of optimizers, such as root mean square prop (RMSProp), stochastic gradient descent

(SGD), adaptive gradient (Adagrad), or other gradient descent algorithms (Taqi *et al*., 2018; Kamsing *et al*., 2019; Vani *et al*., 2019; Lee *et al.*, 2019; Ben-Cohen *et al*., 2018). Therefore, since our problem has only 2 classes and the size of our data is relatively small, we applied Adam as our optimizer to improve generalization performance.

Adam allows us to apply time-varying learning rate by setting decay rate. Setting a high learning rate can avoid convergence at local minima in earlier epochs while increasing the chance of convergence in later epochs. Because the learning rate affects suitable batch size options, batch size, learning rate, and decay all need to be optimized in order to produce a high-performing CNN. The choice of the initial learning rate will also affect our training curve and the acceptable range of decay function of our learning rate in each epoch.

Learning rate can also affect other hyperparameters. Because larger batch sizes can prevent finding the global minimum, higher learning rates can be used to help encourage faster convergence to the actual minimum. Moreover, larger batch size means fewer training iterations per epoch, so higher decay value should be used to converge at a minimum cost point. However, higher learning rate does not guarantee better network performance. As shown in the right hand side in Figure 6, higher learning rates can also lead to higher minimum loss or divergence (Smith *et al*., 2017; Surmenok, 2017). For our CNN, learning rates ranged from 0.001 to 0.000001 based on different architecture and other hyperparameter settings. The learning rate, decay, and number of epochs in our best performing CNN are 0.0004, 0.0004, and 1,000, respectively. Overall, finding optimal values for the various hyperparameters required careful tuning, as many hyperparameters are interdependent.

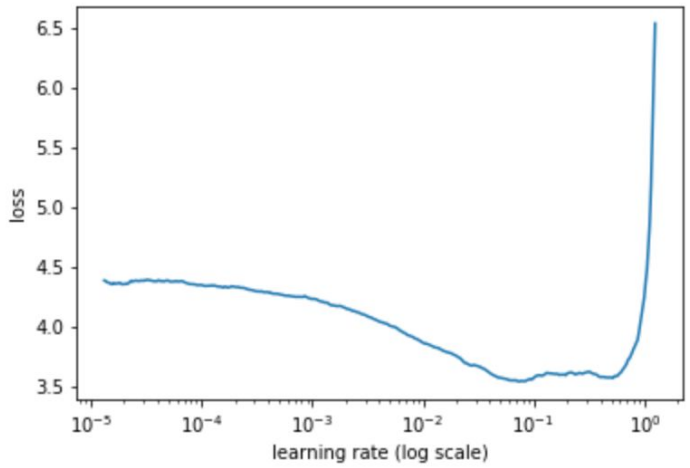


Figure 6. Graphing learning rate versus loss shows that finding the optimal learning rate is necessary to minimize loss (Surmenok, 2017).

To measure generalization performance, we used the AUC of the validation data of 0.991. In applying this model to the entire test data, our AUC was 0.994. This relatively successful generalization performance on the unseen test data may be because our initial learning rate is not too high, enabling both a smoother learning curve and the ability to train more epochs without overfitting. However, using high epoch numbers does not guarantee better performance. Throughout our testing of hyperparameters, we have a network that got 0.986 AUC from a 300 epoch training process, but we have one another got 0.982 AUC from a 600 epoch training

process. This shows the importance of selecting the optimal number of epochs to generate higher performance, measured by AUC for our model.

Besides the learning rate, the loss function and data quality can also affect the speed and the quality of optimization. In order to optimize the learning process, we weighted for different classes by penalizing class 1 predictions, leading to a less-biased binary cross entropy loss function (Ben-Cohen *et al*., 2018). Lastly, in general, since high accuracy models lead to high AUC performance, and the metric for this competition is AUC, we chose accuracy as our metric when model fitting and AUC for comparing model results.

Dropout layer

Dropout layers are a relatively recent development for deep learning modeling and provide a regularization method to prevent CNNs from overfitting. According to Zhao *et al*.(2018), Krizhevsky *et al*.(2012), Ly *et al*.(2018), and Zhu *et al*.(2018), this technique can force our models to learn more robust features to avoid overfitting. Without including dropout layers, some of our initial CNNs overfit to the training data. As with other hyperparameters, finding the optimal dropout rate in a hidden layer is extremely model dependent. Through thorough testing, we found that a dropout rate of 0.25 produced the best-performing model while enabling good generalization performance. This means that 25% of the inputs will be randomly excluded from each update cycle. Dropout is usually applied to the fully connected layers, but Park & Kwak (2017) found that adding dropout layers can yield a more accurate model. Therefore, based on thorough testing supported by research, we added dropout layers to our final CNN.

In summary, generating a high-performing CNN with good generalization performance required thorough testing of all hyperparameters. Comparing various trial CNN models based on AUC enabled selection of a final, best-performing CNN model with the structure detailed in figure 5.

Results

As previously stated, the provided training data was randomly split into 70% training and 30% validation sets. Both the SVM and the CNN were trained using the same 70% training subset, and model results including AUC, ROC curves, and PR curves were generated on the 30% validation set. This method of splitting in 70-30 training and validation sets was used to help gauge generalization and model performance. Cross validation was not used in this case because of computational expense when training the CNN. Therefore, to enable model performance comparison, the SVM was also trained on the same subset of data.

To compare model performance, we used the area under the curve (AUC) of the receiver operating characteristics (ROC) curve. The AUC measures a model's ability to separate the two classes where an AUC of 1 indicates that the model perfectly separates two classes. The ROC curve plots false and true positive rates at various thresholds, where the optimal ROC curve hugs the upper left corner. To compare the performance of our different models, we used the AUC of the ROC curve. For our baseline SVM with HOG model, the AUC was 0.83 on the validation data (Figure 7). In comparison to random guessing with an AUC of 0.49, the SVM model

performs much better (Figure 7). However, the CNN performs even better than the baseline SVM model with an AUC of 0.991 on the validation data (Figure 7). By comparing AUC values and ROC curves, we can conclude that the CNN performs the best.

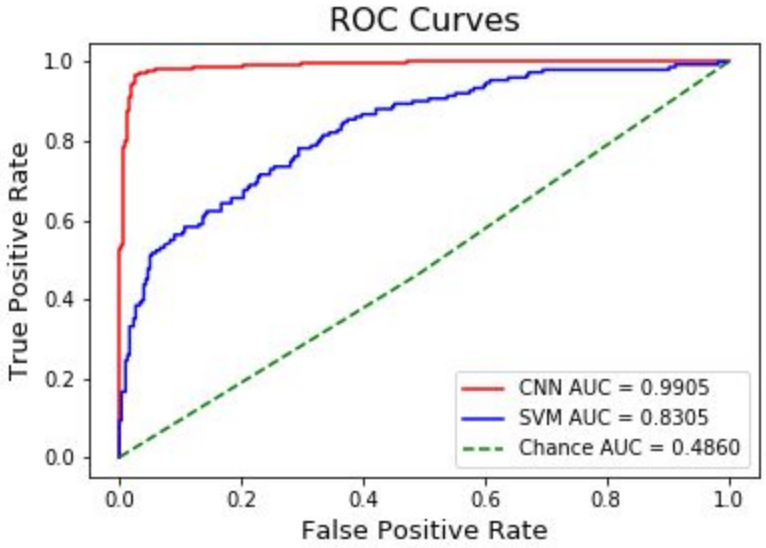


Figure 7. Plot of ROC curves for CNN, SVM, and random guessing. Based on the AUC, we can see that the CNN performs the best, and the SVM provides a baseline and performs much better than random guessing.

In addition to comparing model performance with AUC, we can also look at precision-recall (PR) curves for the three different models (Figure 8). Perfect classifiers hug the upper right corner of the PR curve, and based on Figure 8, we can see that again the CNN performs the best in terms of recall and precision. The baseline cross-validated SVM with HOG model still performs better than random guessing but not nearly as well as the CNN. In addition to comparing PR curves (Figure 8), we can also calculate precision and recall based on the confusion matrices (Figure 9). The precision and recall for the SVM on the validation data are 0.573 and 0.781, respectively. This indicates that when the SVM predicts the presence of a solar panel, it is correct 57% if the time, while it correctly identifies solar panel presence 78.1% of solar panel-containing images. Likewise, the precision and recall for the CNN are 0.973 and 0.940. This indicates that when the CNN predicts presence of a solar panel, it is correct 97.3% of the time, and the CNN correctly identifies 94% of all PV-containing images. Again, based on precision and recall calculations, we can see that the CNN performs better than the SVM.

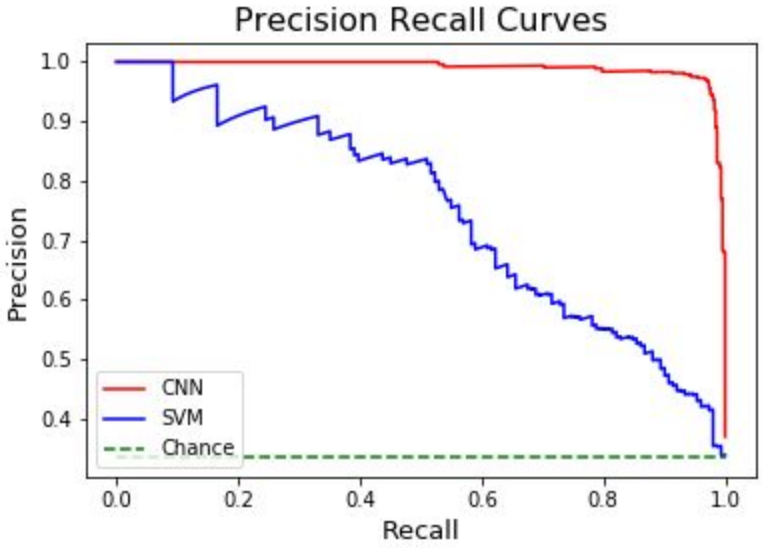


Figure 8. Plot of precision-recall curves for CNN, SVM, and random guessing. Again, we can see that the CNN performs the best while the SVM still outperforms random guessing.

While the ROC and PR curves provide a good sense of model performance in terms of true and false positive rates and precision at different thresholds, it is also helpful to identify misclassifications for both models. Figure 9 (a) and figure 9 (b) show the confusion matrices for the SVM and CNN, respectively, on the validation set. To generate binary predictions from the probability predictions from each of the models, the optimal threshold was found by determining the maximum point on the ROC curve (closest to upper left corner) where the false positive rate is minimized but the true positive rate is maximized. The predicted probabilities from the SVM and CNN were then thresholded based on the maximum point of the ROC curve, where probabilities greater than the threshold were predicted to be 1 (solar panel present) and those less than the threshold were predicted to be 0 (no solar panel present). The threshold for the SVM and CNN are 0.27 and 0.57, respectively. After thresholding the probabilities to generate binary predictions, these predictions were compared to the actual labels in the validation set to generate the confusion matrices in figure 9.

Comparing the confusion matrices in figure 9, we can see that overall, the CNN performs much better, with only 13 misclassifications relative to the 121 misclassifications from the SVM. Interestingly, the SVM has more false positives than false negatives while the CNN has more false negatives than false positives. Therefore, it is possible that the SVM struggles when an image has characteristics that seem similar to a solar panel, leading to the higher false positive rate. However, the converse may be true for the CNN, as the CNN has more false negatives than false positives. In order to better determine instances in which both the SVM and CNN misclassify, we can look for patterns in the input images.

SVM		Predicted	
		Solar Panel	No Solar Panel
Actual	Solar Panel	118	33
	No Solar Panel	88	211

(a) SVM confusion matrix

CNN		Predicted	
		Solar Panel	No Solar Panel
Actual	Solar Panel	142	9
	No Solar Panel	4	295

(b) CNN confusion matrix

Figure 9. (a) Confusion matrix from SVM with HOG model and (b) Confusion matrix from CNN. Both (a) and (b) are generated from the validation set.

First, we can observe misclassified images from the SVM predictions to identify areas where the SVM struggled to correctly classify. Figure 10 shows examples of false negatives and false positives from the SVM predictions. First looking at the false negatives from the SVM (Figure 10 (a)), we can see that the model often misses solar panel detection when the roof is similarly colored. Looking at some examples of the false positives from the SVM (Figure 10 (b)), the model may incorrectly predict solar panel array presence when the image has a lot of shadows, often caused by trees in the bottom left and bottom right images of figure 10 (b). However, in looking at examples of both false positives and negatives, it is possible that the SVM has difficulty distinguishing pools and shadows from solar panels. Given that the SVM model predicts from HOG output, this is understandable considering pools are often relatively lightly colored, which may lead to large changes in gradient magnitude that resemble those of solar panels. Likewise, shadows also can cause similar behavior because shadows and solar panels are often both darkly colored, leading to large changes in intensity captured by HOG.

(a) Examples of false negatives from the SVM.

(b) Examples of false positives from the SVM.

Figure 10. Examples of images that were incorrectly classified by the SVM model, including false negatives (a) and false positives (b). False negatives often occur when the solar panel and roof are similarly colored, while false positives can occur because of shadows often caused by trees. Generally, the model also has problems distinguishing pools from solar panels.

We can also identify patterns in misclassifications for the CNN predictions to further clarify when the CNN performs poorly. First looking at example images of false negatives from the CNN predictions (Figure 11 (a)), we can see that the CNN may have difficulty when the solar panels and roof are very closely colored, as in the top left image. Furthermore, the CNN may miss the solar panel array if it is small, as in the top middle and right images. Lastly, it is possible that solar panels on the edge of the image may miss detection. Looking at example images of false positives from CNN predictions (Figure 11 (b)), we can see that the CNN may have difficulty distinguish shadows from solar panels, leading to falsely predicting solar panel presence. However, similar to the pattern observed in SVM misclassifications, it seems as though the CNN has difficulty when images have swimming pools, shadows, and/or similar coloring between solar panel arrays and roofs. After re-examination, we found that approximately 15% of the images contain pools, and the positions and shapes of the pools vary. Therefore, future methods could train on more images with pools to help models better distinguish pools and solar panels.

(a) Examples of false negatives from the CNN.

(b) Examples of false positives from the CNN.

Figure 11. Examples of images that were incorrectly classified by the CNN model, including false negatives (a) and false positives (b). False negatives often occur when the solar panel and roof are similar colored, the solar panel array is relatively small, or if the panel is on the edge. False positives can occur because of shadows or pool presence.

In looking at incorrectly classified images from both the SVM and CNN, we can see a general pattern where both models have difficulty when images have pools and/or shadows or when the solar panel and roof colors are similar. While the CNN handles these issues better than the SVM, leading to overall higher accuracy and better performance, both of the models suffer from these general patterns. Future work could be done to better distinguish pools and shadows from solar panel arrays to further improve model performance.

In addition to identifying areas where the models perform poorly, it is also important to identify instances where the models perform well. In addition to simply looking at correctly classified images, we can also identify images where models were more confident in the correct prediction by looking at predicted probabilities for both models for true positives and negatives. First looking at the correct predictions from the SVM (Figure 12), we can see that the SVM most confidently identifies presence or absence solar panels when the roofs are lighter colors. Theoretically, this enables HOG to show greater magnitude changes between light and dark colored areas.

(a) Examples of high-confidence true positives from the SVM.

(b) Examples of high-confidence true negatives from the SVM.

Figure 12. Examples of high-confidence true positive (a) and true negative (b) predictions for the SVM. Generally, the SVM performs well with light colored roofs.

Likewise, when looking at correct image predictions from the CNN (Figure 13), we can see that the CNN performs best with light-colored roofs as well. In these cases, the predicted probabilities are almost 1.0, indicating that the CNN is almost 100% confident that there is a solar panel in the true positives (Figure 13 (a)) or that there is not a solar panel in the true negatives (Figure 13 (b)). Interestingly, in looking at high-confidence true positives (Figure 13 (a)), we can also see that the PV arrays in these cases are all rectangular with no irregularities. It is possible that this pattern also helps produce more accurate, high-confidence predicted probabilities for the CNN.

(a) Examples of high-confidence true positives from the CNN.

(b) Examples of high-confidence true negatives from the CNN.

Figure 13. Examples of high-confidence true positive (a) and true negative (b) predictions for the SVM. Generally, the SVM performs well with light colored roofs.

Overall, in assessing the performance of the SVM and CNN, we compared AUC on a random 30% validation set taken from the provided training data, where both models were trained on the same subset of images. Comparing the ROC and PR curves, it is clear that the CNN performs the best, but the SVM still performs better than random guessing in terms of true and false positive rates and precision. Generally, both models struggle when images have pools or shadows, leading to misclassification. However, both models perform well when images contain light-colored roofs, as both models can more confidently classify these images.

Conclusion

With the growth of renewable energy and increasing concern of climate change, solar panels have become more popular, leading to the installation of PV arrays on homes and buildings (*EIA - Electricity Data*, 2019). However, with this growth comes the desire to gauge solar panel prevalence, as this could provide helpful information to climate change activists, governments, utility companies, and policy makers. How many homes have solar panels? How

fast is solar panel installation spreading? In what areas is renewable energy less popular? What are the implications of this growth of renewable energy? While current efforts rely on surveys to gather information about solar panels, these surveys are costly and often have low response rates (Malof *et al*., 2016). More efficient, effective, and successful identification of solar panel arrays can help provide more detailed and accurate information regarding solar PVs.

Machine learning algorithms have been developed to address the issues of inefficient and ineffective solar PV detection. The current state of the art solar PV detection algorithms use different types of CNNs or hybrid models with other machine learning algorithms, such as a CNN with a SVM (Dong *et al*. 2018; Meng *et al*. 2016), and are successful in identifying solar PVs regardless of their location. Our results validate the finding that CNNs achieve high accuracy on solar PV classification on large datasets at high resolutions (Malof *et al.*, 2017; Camilo *et al*., 2017; Mnih & Hinton, 2010).

In this study we compared the ability of two machine learning models, SVMs and CNNs, in classifying satellite imagery of solar PVs. First, we focused on image preprocessing and feature extraction and found that HOG captured details of the satellite imagery better than other image preprocessing techniques. Because HOG enables better detection of edges and changes in pixel intensity, it helped address the previously described challenges in inconsistent panel array shape. Using the HOG output with an SVM trained on 70% subset of the provided training data, our validation set AUC was 0.83, which is an improvement from random guessing at 0.49. Though SVM with HOG performs moderately well in part due to SVM's ability to handle high dimensional data and HOG's ability to encode changes in image gradients, we also compared this model to a CNN because CNNs are well known to successfully classify images. For our CNN model, we implemented data augmentation to generate more training data. We also created a deeper network and applied ReLu function and dropout layers to avoid overfitting. In addition, we use Adam as our optimizer to adjust our learning rate accordingly. With these techniques, our network performance increased from 0.85 AUC to 0.991 AUC.

After tuning hyperparameters and then training the two models on the same 70% subset of the provided training data, we compared the CNN to the SVM model. We found that the CNN performs much better, with an AUC of 0.991 on the validation set, a significant improvement from the AUC of 0.83 of the SVM model.

Based on these results, we can conclude that the CNN performs significantly better than the SVM in terms of true and false positive rates in addition to precision. However, there is still room for improvement. Both the CNN and SVM struggled to correctly classify images with swimming pools or shadows in addition to when solar panels and roof coloring were similar. Future work could improve model performance by accounting for pools, shadows, and roof coloring. However, despite this difficulty, both models performed relatively well, indicating that in the future, more widespread implementation and use of machine learning models like CNNs for solar PV classification in aerial imagery could provide valuable information about the adoption and use of renewable energy.

References

A. Ben-Cohen, E. Klang, M. M. Amitai, J. Goldberger and H. Greenspan, "Anatomical data augmentation for CNN based pixel-wise classification," 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, 2018, pp. 1096-1099.

A. M. Taqi, A. Awad, F. Al-Azzo and M. Milanova, "The Impact of Multi-Optimizers and Data Augmentation on TensorFlow Convolutional Neural Network Performance," 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Miami, FL, 2018, pp. 140-145.

A. Wiranata, S. A. Wibowo, R. Patmasari, R. Rahmania and R. Mayasari, "Investigation of Padding Schemes for Faster R-CNN on Vehicle Detection," 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Bandung, Indonesia, 2018, pp. 208-212.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in CVPR, 2015, pp. 1–9.

Camilo, J., Wang, R., Collins, L. M., & Malof, J. M. (2017). *Application of a semantic segmentation convolutional neural network for accurate automatic detection and mapping of solar photovoltaic arrays in aerial imagery*. 8.

D. Cire¸san, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. Arxiv preprint arXiv:1202.2745, 2012.

Dollar, P., Zhuowen Tu, & Belongie, S. (2006). Supervised Learning of Edges and Object Boundaries. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*, *2*, 1964–1971. https://doi.org/10.1109/CVPR.2006.298

EIA - Electricity Data. (2019, December). US Energy Information Administration. https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=epmt_1_17_b

F. Chou, Y. Tsai, Y. Chen, J. Tsai and C. Kuo, "Optimizing Parameters of Multi-Layer Convolutional Neural Network by Modeling and Optimization Method," in IEEE Access, vol. 7, pp. 68316-68330, 2019.

F. Ertam and G. Aydın, "Data classification with deep learning using Tensorflow," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, 2017, pp. 755-758.

G. Zhao, Z. Zhang, H. Guan, P. Tang and J. Wang, "Rethinking ReLU to Train Better CNNs," 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, 2018, pp. 603-608.

Gleason, J., Nefian, A. V., Bouyssounousse, X., Fong, T., & Bebis, G. (2011). Vehicle detection from aerial imagery. *2011 IEEE International Conference on Robotics and Automation*, 2065–2070. https://doi.org/10.1109/ICRA.2011.5979853

G. J. Scott, M. R. England, W. A. Starms, R. A. Marcum and C. H. Davis, "Training Deep Convolutional Neural Networks for Land–Cover Classification of High-Resolution Imagery," in IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 4, pp. 549-553, April 2017.

G. Zhu, B. Li, S. Hong and B. Mao, "Texture Recognition and Classification Based on Deep Learning," 2018 Sixth International Conference on Advanced Cloud and Big Data (CBD), Lanzhou, 2018, pp. 344-348.

H. Ide and T. Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2017, pp. 2684-2691.

Hermes, L., Frieauff, D., Puzicha, J., & Buhmann, J. (1999). *Support Vector Machines for Land Usage Classification in Landsat TM imagery*. ResearchGate. https://www.researchgate.net/publication/3803218_Support_vector_machines_for_land_usage_classification_in_Landsat_TM_imagery

Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Mnih, V., & Hinton, G. E. (2010). Learning to Detect Roads in High-Resolution Aerial Images. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Computer Vision – ECCV 2010* (Vol. 6316, pp. 210–223). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-15567-3_16

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning* (Vol. 103). Springer New York. https://doi.org/10.1007/978-1-4614-7138-7

K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in CVPR, 2016, pp. 770–778.

K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In International Conference on Computer Vision, pages 2146–2153. IEEE, 2009.

K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in ICLR, 2015.

Kluckner, S., Mauthner, T., Roth, P. M., & Bischof, H. (2009). Semantic Image Classification Using Consistent Regions and Individual Context. *Procedings of the British Machine Vision Conference 2009*, 25.1-25.12. https://doi.org/10.5244/C.23.25

Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.

L. Bragilevsky and I. V. Bajić, "Deep learning for Amazon satellite image analysis," 2017 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Victoria, BC, 2017, pp. 1-5.

L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, 2017, pp. 464-472.

M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in ECCV, 2014, pp. 818–833.

Malof, J. M., Bradbury, K., Collins, L. M., & Newell, R. G. (2016). Automatic detection of solar photovoltaic arrays in high resolution aerial imagery. *Applied Energy*, *183*, 229–240. https://doi.org/10.1016/j.apenergy.2016.08.191

Malof, J. M., Rui Hou, Collins, L. M., Bradbury, K., & Newell, R. (2015). Automatic solar photovoltaic panel detection in satellite imagery. *2015 International Conference on Renewable Energy Research and Applications (ICRERA)*, 1428–1431. https://doi.org/10.1109/ICRERA.2015.7418643

M. Geng, Y. Wang, Y. Tian and T. Huang, "CNUSVM: Hybrid CNN-Uneven SVM Model for Imbalanced Visual Learning," *2016 IEEE Second International Conference on*

Multimedia Big Data (BigMM), Taipei, 2016, pp. 186-193.
Mnih, V., & Hinton, G. E. (2010). Learning to Detect Roads in High-Resolution Aerial Images. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Computer Vision – ECCV 2010* (pp. 210–223). Springer. https://doi.org/10.1007/978-3-642-15567-3_16.
P. Kamsing, P. Torteeka and S. Yooyen, "Deep Convolutional Neural Networks for plane identification on Satellite imagery by exploiting transfer learning with a different optimizer," IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 2019, pp. 9788-9791.
P. Ly, D. Bein and A. Verma, "New Compact Deep Learning Model for Skin Cancer Recognition," 2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York City, NY, USA, 2018, pp. 255-261.
R. . Salakhutdinov and G. . Hinton, "Replicated softmax: An undirected topic model," Adv. Neural Inf. Process. Syst. 22 - Proc. 2009 Conf., pp. 1607–1614, 2009.
Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A. S., Milojevic-Dupont, N., Jaques, N., Waldman-Brown, A., Luccioni, A., Maharaj, T., Sherwin, E. D., Mukkavilli, S. K., Kording, K. P., Gomes, C., Ng, A. Y., Hassabis, D., Platt, J. C., … Bengio, Y. (2019). Tackling Climate Change with Machine Learning. *ArXiv:1906.05433 [Cs, Stat]*. http://arxiv.org/abs/1906.05433
S. Lee *et al*., "Improving Scalability of Parallel CNN Training by Adjusting Mini-Batch Size at Run-Time," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 830-839.
S. Vani and T. V. M. Rao, "An Experimental Approach towards the Performance Assessment of Various Optimizers on Convolutional Neural Network," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 331-336.
S. Zagoruyko and N. Komodakis, "Wide residual networks," in BMVC, 2016.
S. Zhai *et al*., "S3Pool: Pooling with Stochastic Spatial Sampling," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 4003-4011.
Saito, S., Yamashita, T., & Aoki, Y. (2016). Multiple Object Extraction from Aerial Imagery with Convolutional Neural Networks. *Electronic Imaging*, *2016*(10), 1–9. https://doi.org/10.2352/ISSN.2470-1173.2016.10.ROBVIS-392
Solar PV in Aerial Imagery, Kaggle, Feb. 2020.[Online]. Available: https://www.kaggle.com/c/ids705sp2020/data
Sungheon, P., Nojun, K. (Feb. 2017) "Analysis on the Dropout Effect in Convolutional Neural Networks".
Surmenok, P. (2017). Estimating an Optimal Learning Rate For a Deep Neural Network. https://towardsdatascience.com/estimating-optimal-learning-rate-for-a-deep-neural-network-ce32f2556ce0
T. Dong, X. Qi, W. Li and M. Qin, "Target Feature Recognition Based on Wavelet Transform and CNN-SVM," *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, 2018, pp. 347-350.
Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, pages 253–256. IEEE, 2010.
Youssef, M. M. S., Mallet, C., Chehata, N., Le Bris, A., & Gressin, A. (2014). Combining

top-down and bottom-up approaches for building detection in a single very high resolution satellite image. *2014 IEEE Geoscience and Remote Sensing Symposium*, 4820–4823. https://doi.org/10.1109/IGARSS.2014.6947573

X. Glorot, A. Bordes, and Y. Bengio (2017), “Deep sparse rectifier neural networks.” 2011.