Solar PV Detection in Aerial Imagery

Authors: Abhishek Baral, Nathan Warren, Yiran Chen, Jiayue Xu, Ravitashaw Bathla

Abstract

Solar panels have been a popular source of renewable energy as increasing concerns of climate change have come to the forefront of people's minds. With the recent increase in solar panel installation, policy makers and energy producing organizations need to understand the number and distribution of these panels to regulate energy needs.

In this study, we analyze two different approaches for solar photovoltaic (PV) array detection. We present a Support Vector Machine with a feature selection based approach. To obtain higher performance, we investigated deep end-to-end Convolutional Neural Networks (CNNs). Both approaches are trained on an aerial imagery dataset of buildings in urban areas. After being cross-validated, each algorithm displayed strong classification performance, with AUCs of 0.991 and 0.823 for the CNN and the SVM, respectively. These results display excellent classification performance that may assist in future decision making in the renewable energy industry.

Section 1: Introduction

Sustainability has been one of the most important criteria as energy demand increases as well as the concerns of climate change. The reliance on renewable sources has been a major leap forward to eradicate these problems [25]. With constraints in urban spaces, there has been an increasing growth in installation of Solar Photovoltaic (PV) units on the rooftops of the buildings. The government has been making efforts to encourage installation of solar panels among house owners, which can provide sustainable power for day-to-day living needs, though tax credits and subsidies [10]. Energy organizations study the impact of the installation of these solar panels to regulate and monitor their production cost. Since there is an uneven distribution of sunlight across different regions, the number of solar panels significantly impacts the energy produced from these PV panels.

To understand the impact from the installation of these solar panels, it is imperative to correctly classify the number and distribution of PVs installed across different regions. This understanding will help the decision makers to promote sustainable living across regions. In addition, it will help power producing organizations regulate and monitor the production of electricity in their respective areas to avoid power wastage. It is extremely difficult to get a manual count of these panels as it requires substantial logistics costs and does not provide precise numbers. Several survey based approaches have been established but proven to be ineffective [9, 23]. The local residents tend not to respond to surveys and the data collection process can be prone to errors or

missing information. Hence, a more cost efficient way of solar panel detection needs to be explored. One of the ways is by exploring automatic solar panel object detection on aerial images of urban households.

Section 2 describes the background and current work that has been done for image classification. In Section 3, we describe the solar panels data and the challenges that we could face for the classification of images based on if solar panels are present. The methodologies including feature selection that have been used in this paper are described in Section 4. Section 5 presents the results that were obtained from the models and a comparison between these models have been discussed. The conclusion and future work are presented in Section 6.

Section 2: Background

There have been several traditional machine learning approaches for performing image classification that have performed well over a range of different image datasets. The most significant challenge that researchers face is feature extraction for image classification using traditional machine learning approaches [15]. Since the dimensionality of an RGB image is substantially large, with a limited dataset, the model can be disqualified in generalizing well and can potentially lead to convergence issues during training [4]. Several techniques like Principal Component Analysis (PCA) [29], Scale-Invariant Feature Transform (SIFT) [18] and Histogram Oriented Gradients (HOG) [6] have been developed for dimensionality reduction and feature extraction. PCA is efficient in extracting generalized features for the whole image but can be limited in extracting image segments. SIFT is able to produce good results in edge detection for distinctive edges in the images. On the other hand, HOG provides prominent performance in boundary detection over a background of varying color densities [6].

The traditional machine learning approaches like Support Vector Machines (SVM), Random Forest, Logistic Regression are used as classifiers which are trained on these extracted features. Since SVM is constructed sparsely with the use of kernel function by optimizing support vectors, SVM-based classifiers have better generalization performance than other widely used classifier techniques [17]. These characteristics of SVM provide substantial lead, compared to other techniques, in image classifications and consequently help avoid misclassification because it is defined by a convex optimization problem with no local minima. All these classical approaches perform well with small datasets and are computationally inexpensive [3].

In contrast to these traditional approaches, deep learning techniques which rely on the stacking of several layers of neural network have been extensively studied for a variety of tasks. For computer vision applications, Convolutional Neural Network (CNN) based approaches are considered to be state-of-the-art techniques in image classification, image segmentation, and

object detection tasks etc [13, 27]. Such methods offer promising results for image classification, however, these require a large amount of high-quality data for training to effectively function. Due to the lack of labelled training data, use of ImageNet pre-trained models to extract deep features are often employed.

Section 2.1: Related Work

The solar panel detection has been studied in [19, 20, 21] over the last couple of years by utilizing Random Forest classifier and CNN based approaches. Varying size of data has been used for the study, with substantial improvements over the years. One of the approaches for identifying solar panels was to use a Random Forest classifier, which was trained by extracting the features using the pixel-based and object-based algorithms [19]. Semantic segmentation CNNs for pixel based labels have proven to be effective in classifying the solar panels as well [20]. In the preceding study, a framework SolarMapper was introduced which uses a deep CNN for understanding the topology of the solar panels in the aerial imagery [21].

With recent advancements in different variants of CNN, in this work we present a more accurate classifier called ResNet which is pre-trained on the ImageNet [7] dataset and we further fine tune on the solar imagery dataset. We further provide an analysis on the comparison in performance of SVM with CNN based approach.

Section 3: Data

The data set consisted of 1500 RGB aerial images of the rooftops of houses and buildings. As seen in Figure 1, there were 33.6% images that had solar photovoltaic (PV) array located on the rooftops. The images without the solar PV include empty rooftops, parking lots or roadsides. These images can help to better train the model for precise detection of the solar PVs. The dimensions of the images are 101 by 101 pixels with all the three RGB color channels.

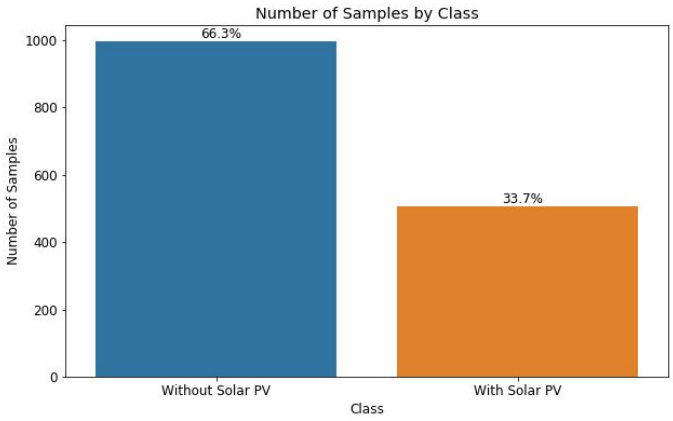

Figure 1: Distribution of label class. 66.3% of the original 1500 solar PV array images are without solar PV arrays and the remaining 33.7% of training images contain PV arrays.

Challenges inherent to this classification problem are the vast amount of features compared to the amount of photos in the training set, which can lead to overfitting. There were challenges in the images themselves as well. In certain images, it was difficult to detect PV arrays due to similarities in the color densities between the shade and PV array in the image, as seen in Image ID 10 from Figure 2. In other situations, it was difficult to differentiate the roof from PV array, as seen in Image ID 348 from Figure 2. Additionally, there were also objects on roofs that had similar shapes and colors to those of PV arrays; these objects could easily be mistaken for solar panels, such as roof exhaust as seen in Image 882 from Figure 2. The orientations and the locations of the solar panels also vary from one image to another, this also adds a layer of complexity in terms of classification. To deal with high dimensionality of images, feature detection techniques are commonly used to extract useful information as discussed in Section 2.

While detection techniques may help in most scenarios, they have difficulty when images are less than ideal. One example of this includes how solar panels can appear white due to reflection and overexposure (see Figure 2, Image ID 1415), another is how different shades affect the quality of an image, the location and slant of the rooftop distorting the solar panel shapes. These all could potentially lead to misclassification as will be illustrated further in the Result section.

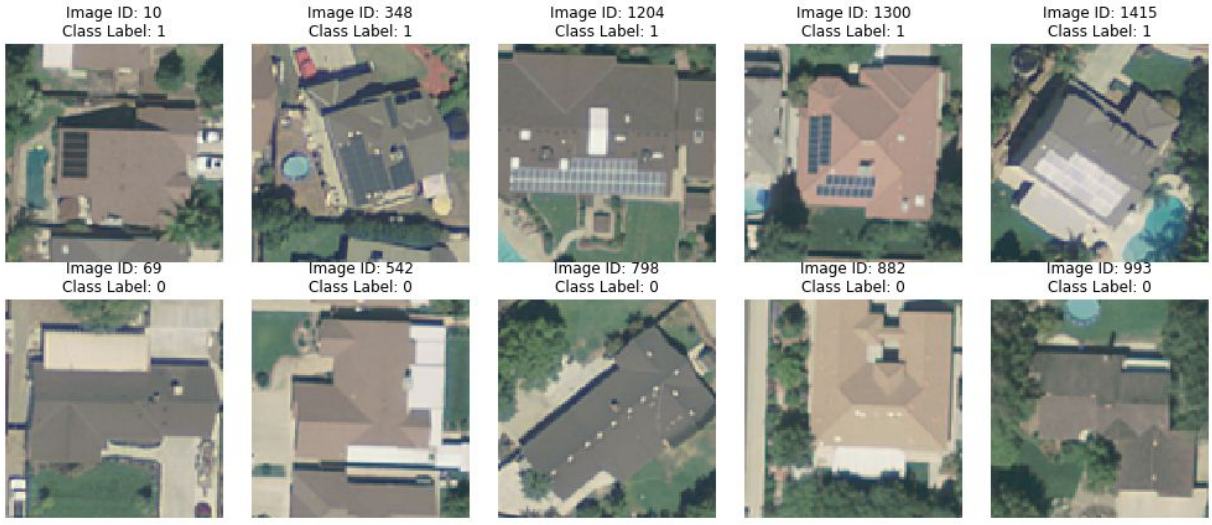

Figure 2: Example images of label class 1 (top row), which had solar PV arrays present and class 0 (bottom row), where solar PV arrays were not present.

Section 4: Methods

4.1 Support Vector Machines with HOG (Non-NN Method)

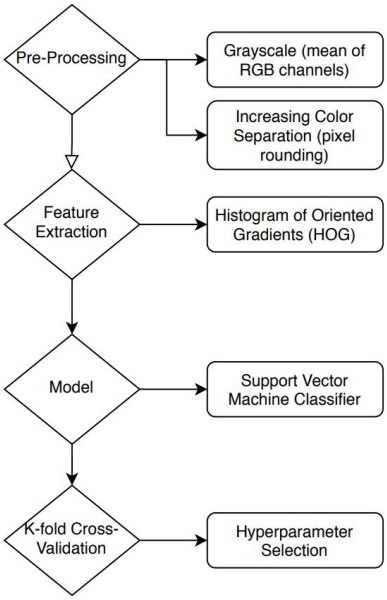

Figure 3: Flow chart of methodology of Non-Neural Network Method. The approach was broken down into four parts. During the pre-processing, the images were greyscaled and the pixels were rounded to the nearest multiple of 20. HOG was used as a feature extraction technique in the next stage. The extracted features were then fed into an SVM classifier for model training. Model hyperparameters were selected through k-fold cross validation.

4.1.1: Pre-Processing

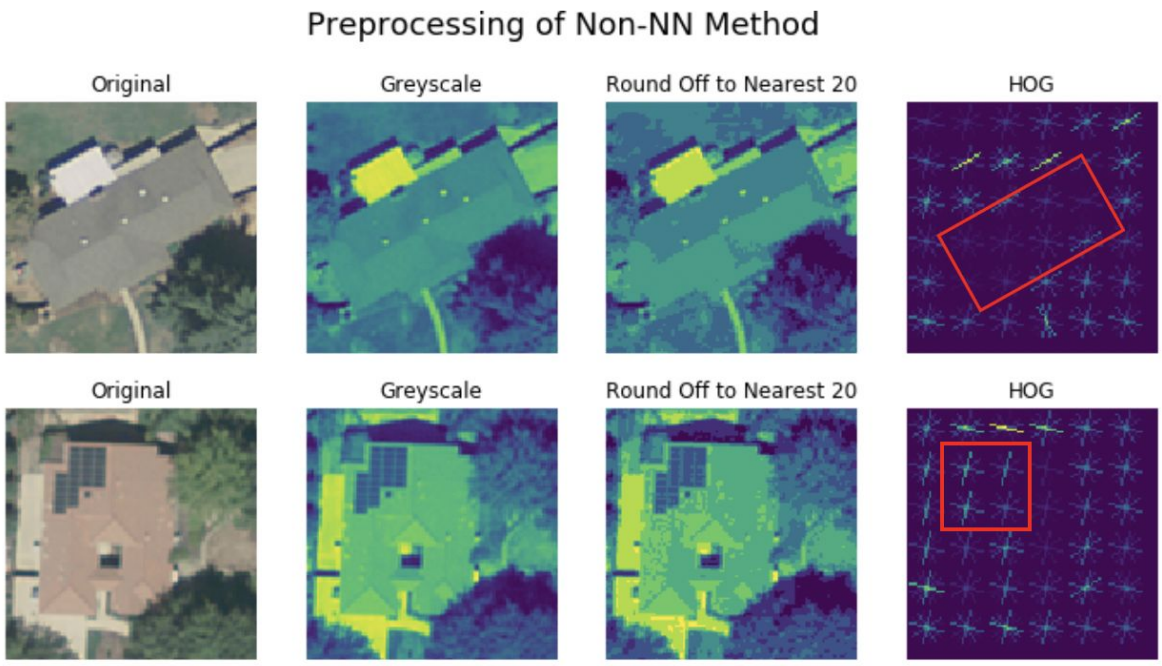

Figure 4: Pre-processing of non-neural net method. Images were first greyscaled, then had their pixel values rounded down towards the nearest multiple of 20. The final step in pre-processing was using HOG in order to extract useful features from the images.

4.1.1.2: Greyscale

The training data was first transformed into greyscale images by taking mean of pixel values across the 3 coloured channels (R, G, B), to reduce the dimensionality of the data without losing too much information (Fig 3,4). This helps in extracting contrasting features from the background as well [28].

4.1.1.3: Increase Intensity Separation

As observed from the visualisation of the images with solar panels, the color contrast between the roof and solar panels could potentially aid in the detection of the solar panels. To further increase the intensity separation between the pixels of the roof and the solar panels, grayscale pixel values are further rounded down to the nearest multiple of 20. As seen in Figure 4 above, this retained sufficient information about the structures in the image with non-apparent changes to the naked eyes, while creating clearer boundaries around areas with intensity changes. Each image, with 101 by 101 pixels was then unravelled into a 1-D array with length of 10201 (101 x 101).

4.1.1.4: Histogram of Oriented Gradients (HOG)

As mentioned in the background, HOG has been selected as the dimensionality reduction technique based on its boundary detection ability over background of varying color densities.

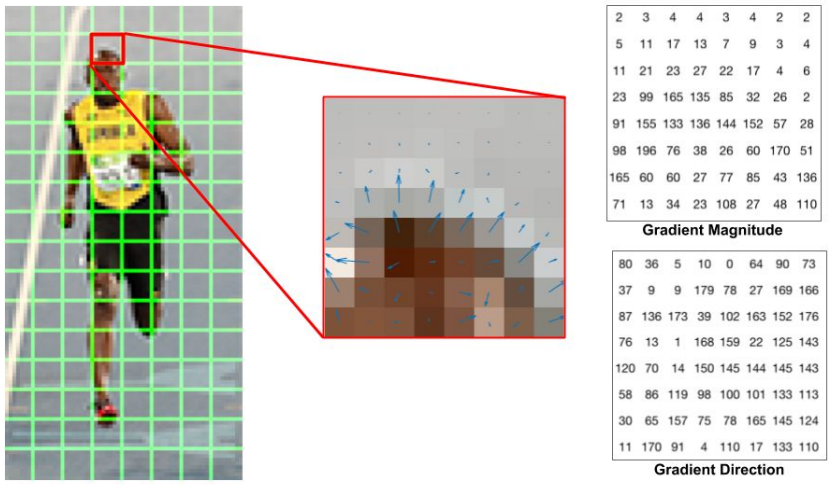

Figure 5: Histogram of Oriented Gradients.

The dimensionality reduction of HOG works by summarizing selected groups (cells) of pixels using the distribution (histograms) of magnitude and direction of variations in the pixel density (gradient) [2]. As shown in Figure 5 from [13], each image is divided into cells of a fixed size where the histogram of gradients are calculated. For each cell, the magnitude and direction matrix are determined based on both the horizontal and vertical gradients of each pixel within the cell. These gradient matrices are then consolidated into frequency tables of directions (orientations) with magnitudes as frequencies. These frequency tables are then normalized over multiple cells to reduce the lighting variation of the image, and finally extracted as features [2].

4.1.1.5: Dimensionality Reduction

Since there are only 1500 training images each with 10201 features, HOG was applied on the flattened image vectors to further reduce the dimensionality before model training. The default parameters of HOG are cell size of 8x8 pixels, block size of 3x3 cells and orientation of 9 which reduces the amount of features per image to 7200 features. This is still a large number of features for our training data with only 1500 observations. The cell size of 16x16 pixels, block size of 4x4 cells and orientation of 8 were used to achieve a desired size of feature space that is lower than the number of observations, reducing the dimensionality of each image to 1152 from 10201 features. As seen from Figure 4 above, HOG features were able to pick up the changes in the intensities between the solar panels and the roof with solar PV panels installed images as represented by the highlighted gradients, while showing mild gradients for images without any solar PV installed images, where the roof is of more homogenous intensity.

4.1.2: Modeling

4.1.2.1: Support Vector Machines (SVM)

As specified in Section 2, Support Vector Machines (SVM) has been commonly used as the binary classifier on object detection of images. When used for binary classification, SVM creates a hyperplane that best separates two classes, by maximizing the distance between the hyperplane and the closest data point [5]. The Radial Basis Function (RBF) kernel was used as it is more flexible than Linear and Polynomial kernels, allowing for a more flexible model with more complex hyperplane [24].

As seen from the data visualisations of the images, there are noticeable variances in the shapes and colour of the solar panels and significant noises in the image which corresponds to other environment objects of similar shapes and colour as the solar panels. This potentially causes the classes to not be perfectly separable with a hyperplane. The soft margins of SVM classifiers

addresses this issue by relaxing the constraint of maximizing the separation margin and allow some degree of error in the classification [5].

The degree to which the amount of violation of the margin is allowed is controlled by the inverse regularization strength C [11]. The smaller the value of C, the more violations of the hyperplane are permitted, hence leading to a less flexible classifier with higher bias and lower variance. Different hyperparameter C were explored for optimal generalisation performance.

4.2.2: Cross-Validation Approach

Using 5-fold cross validation on the entire training data, the SVM classifier was trained on the HOG features. Cross-validation approach was chosen over the train-validation split due to limited training data (1500 observations) and high feature to observations ratio (1152 features vs 1500 observations), so as to ensure all available training data is used in the model training process.

4.2.3: Hyperparameter Selection

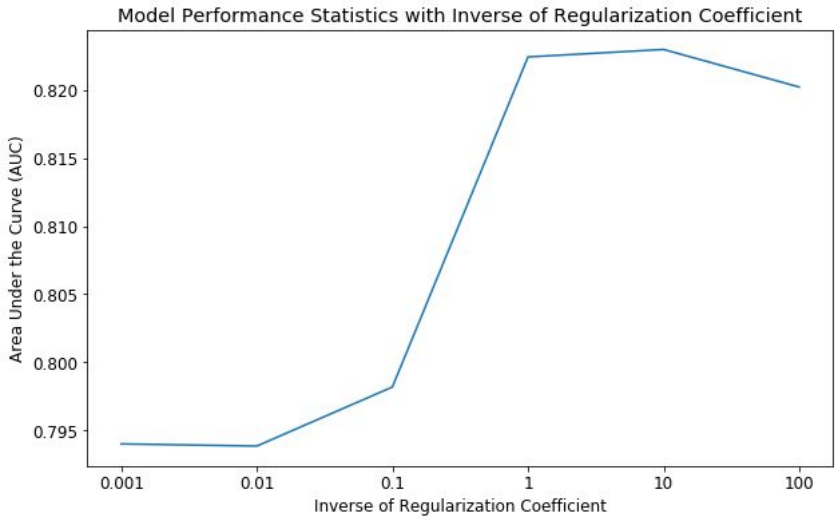

Figure 6: The optimal regularization coefficient for the SVM Classifier is 10.

Using the Radial Basis Function (RBF) kernel, different inverse regularization strength, C, were explored and cross-validated training area under ROC curve (AUC) were compared for the selection of the optimal C value. As seen in Figure 6, as C parameter increases by factor of 10, which corresponds to increased model inflexibility, AUC increases to a maximum of 0.8230 at C=10, indicating that 10 the optimal hyperparameter for generalization performance. Hence, the SVM classifier with C=10 was chosen and its cross-validated performance was evaluated in Section 5.

4.2: Neural Network based Approach (NN Method)

Variants of Convolutional Neural Networks (CNNs) have been considered to be the state-of-the-art techniques for image classification problems for several years now like AlexNet [14], ResNet [13], and VGG [27]. These models belong to the class of Deep Neural Networks and have applications in varying fields. CNNs are based on having at least one convolutional layer, which involves applying convolutional filters or kernels, similar to a sliding window, over input image [16]. Researchers observed that adding several layers of CNN on top of each other (hence the term deep) results in better prediction capability [26].

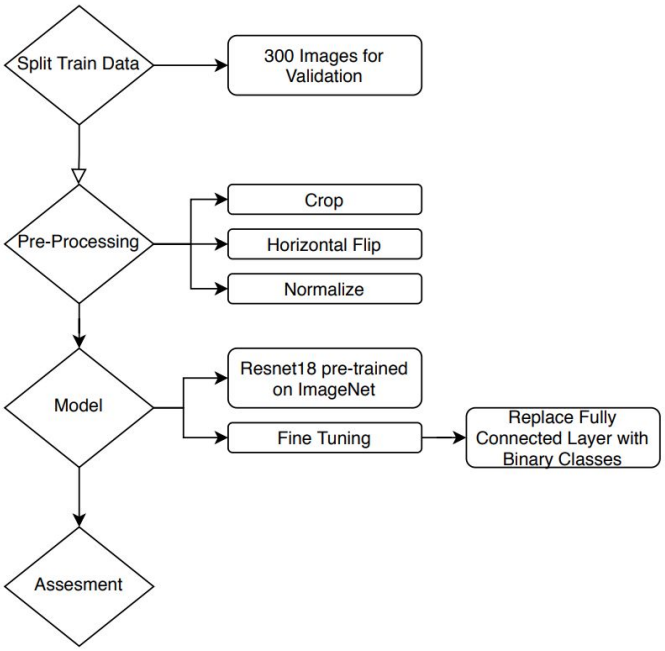

Figure 7: Flow chart of methodology: CNN based method. First the data was split into 1200 training images and 300 images for validation. In Pre-processing, data augmentation was performed with random cropping, random horizontal flips, followed by tensor conversion and normalization. ResNet-18 was used and pretrained on ImageNet with weights adjusted across all layers using the given dataset. Within the architecture, the last fully connected layer was replaced with a binary class. Lastly model assessment was performed for evaluation.

All these CNN variants are deep convolutional neural networks with several layers. Computational complexity has been one of the deciding criteria for selecting the number of layers.

Although deep neural networks provide good prediction scores, stacking several layers on top of each other results in the problem of exploding or vanishing gradients [12]. This is caused when a fraction (less than 1) is multiplied by another similar fraction across different layers, the value decreases considerably. Thus, resulting in the problem of vanishing gradient. Similarly, if the

values are very large then it can result in exploding gradient problems. Such problems were observed in AlexNet and researchers tried to solve it by modifying the architecture of CNNs in ResNet and VGG. The performance of several variants of CNN have been discussed in [25].

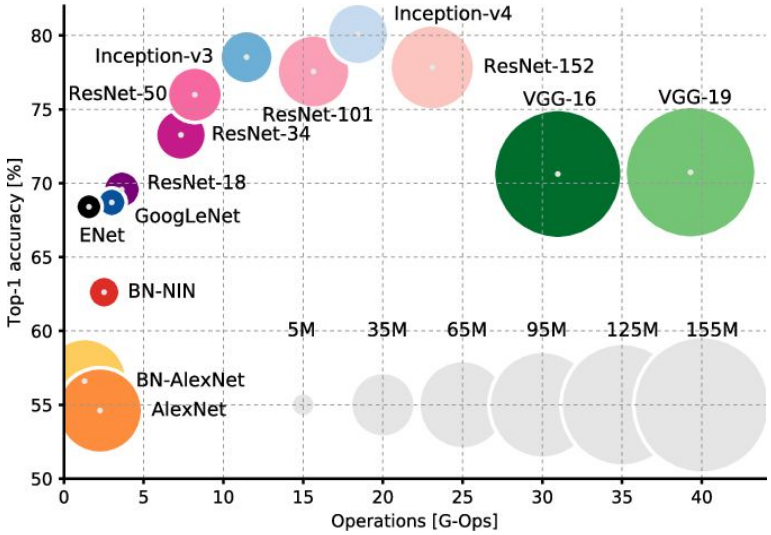

Figure 8: Evaluation of different NN architectures. Top-1 accuracy refers to the conventional accuracy where the one with the highest probability predicted by the model matches with the actual label. The size of the circle is proportional to the number of model parameters in the network.

4.2.1: Residual Network (ResNet)

ResNet are deep CNNs with varying numbers of convolutional, pooling and fully connected layers. For this problem, we used ResNet-18 which has 17 convolutional layers, 2 pooling layers and 1 fully connected layer at the end. These architecture has proven to provide state-of-the-art performance in image classification on several datasets [13]. As per He et al, these networks are based on optimizing the residual mapping which performs better than the original mapping. He et. al further elaborates how the vanishing gradient problem seems to be partially solved by introducing shortcut connections along with passing the identity mapping to the outputs. The network is modularized with residual blocks stacked together [13]. Within each building block, the input goes through two 3x3 convolutional layers (convolutional filters), after which batch normalization is performed and then passed to ReLU activation function. Lastly, the authors of ResNet elaborate that the batch normalization helps reduce internal covariate shift and network's sensitivity to initialization.

4.2.2: Preprocessing

4.2.2.1: Data Loading

The training data provided was first divided into training and validation sets by randomly selecting a subset of 300 (80-20 training-validation split) images for validation during model training (Fig. 7). Although it is ideal to conduct k-fold cross validation, such operation is computationally expensive and time-consuming when the model has a significantly large amount of parameters and hyperparameters to optimize. For model evaluation, we used 300 images for validation. We loaded multiple samples parallelly in a batch size of 30 for efficiency and updated the weights after each propagation. For the training data, batches between epochs were shuffled to make the model more robust for generalization.

4.2.2.2: Data Augmentation

Since we have limited data set for training, data augmentation was performed by creating transformed versions of images. The images were randomly scaled to 224 by 224 pixels, which matches configuration for ResNet-18. Then we performed random horizontal flips, followed by tensor conversion and normalization which is based on data from ImageNet for easier processing [7]. As most solar panels were located in the center of the image, the images were scaled larger up to 256 and center cropped to match the same size 224 x 224 for better generalization. This was followed by tensor conversion and normalization with means of [0.485, 0.456, 0.406] and standard deviations of [0.229, 0.224, 0.225] which are representative of ImageNet dataset.

4.2.3: Modeling

4.2.3.1: Transfer Learning and Model Fine Tuning

The Deep Residual Network (ResNet-18) is a pre-trained model with ImageNet. ImageNet is one of the most popularly used datasets composing of millions of classified images. ImageNet contains one thousand classes including satellite imagery. The features learned are generic and transferable, implying they can be applied to different settings. Therefore, it can be assumed that the features learned from this pretrained network can be used in classifying solar panel satellite images. By using ResNet-18 on the given dataset and fine-tune the weights on all layers using the training dataset, we were able to generate a model with better generalization performance. Instead of predicting 1000 class scores from the pretrained ResNet model, the last fully connected layer was replaced with a new layer that is composed of only two classes, where we trained a linear classifier on top of those features. The network architecture that was used is specified in Figure 9.

The cost function used was cross-entropy loss which the network would try to minimize with respect to the weights. The optimizer during backpropagation was chosen to be stochastic gradient descent where the optimizer uses a batch of thirty observations and updates the weights each time. This is more likely to reach a global minimum with higher flexibility and outperforms other optimizers attempted such as Adam. In addition, a momentum was applied which can give

a larger effective learning rate along the directions of low curvature. We used momentum as 0.9 and the learning rate as 0.01.

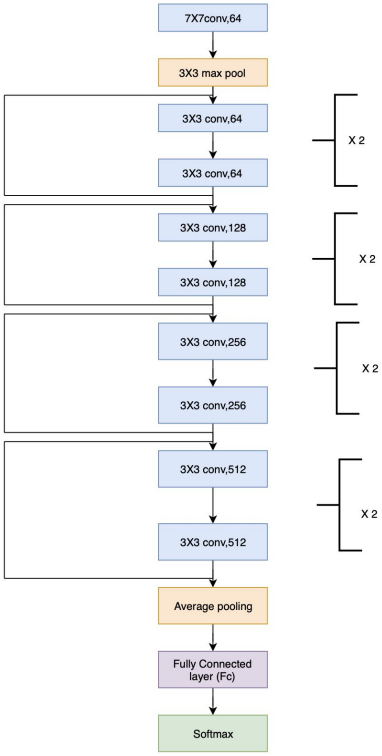

Figure 9. ResNet-18 architecture comprised of 17 convolutional layers, 2 pooling layers and 1 fully connected layer.

4.3: Performance Assessment Metrics

AUC, accuracy, confusion matrices, precision and recall were used as classification metrics for classification generalization ability of the model and were reported in the Result section. For each specific metric, AUC measures a trade-off between false positive rate and true positive rate, which is reported to show the probability of the model correctly classifying a random sample. Accuracy was shown to measure model's overall ability to correctly distinguish between two different classes. Precision-Recall (PR) curves were plotted to show the trade-offs between sensitivity and specificity. The x-axis represents the true positive rate where the model correctly classified positive images among all predicted positive images. The y-axis represents precision where the model correctly classified positive images among all labeled positives. These are useful metrics especially for imbalanced datasets to account for the proportion of positives and negatives in the given dataset. A random classifier has a constant precision of 92/300=0.3067 for all possible recall values. Threshold of 0.5 was used on the prediction scores to obtain the predicted class, where each observation is classified as class 1 (with solar PV) if the prediction scores are more than 0.5 while classified as class 0 (without solar PV) if the prediction scores are

less than 0.5. This is used for the calculation of accuracy, confusion matrices, precision and recall scores.

Section 5: Results

5.1: Performance Metrics Comparison between 2 Approaches

5-fold validation was used for the SVM Classifier which returned an AUC of 0.823 while hold-out validation was used on the neural net which returned an AUC of 0.991 (Fig. 10).

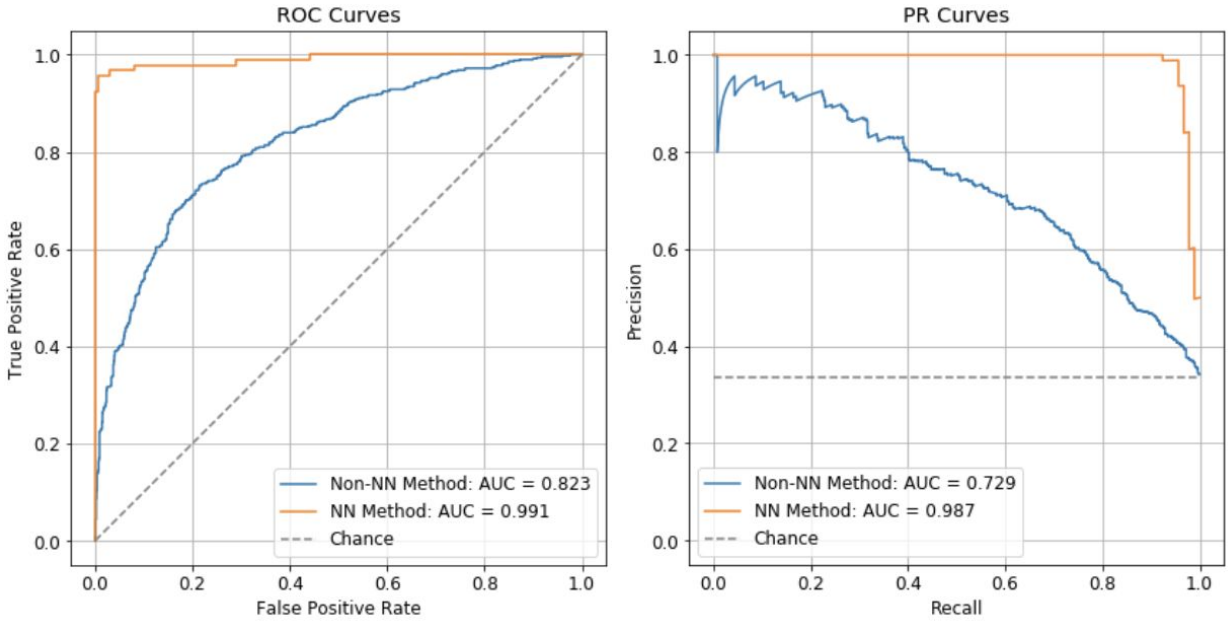

Figure 10: PR and ROC Curves of both SVM classifier and CNN classifier on respective validation data. The CNN classifier result is shown in orange while the SVM classifier result is shown in blue. For both metrics, the CNN classifier scored significantly higher than SVM classifier, achieving an AUC for the ROC and PR curve of 0.991 and 0.987, respectively.

The SVM Classifier was able to classify 78.06% (accuracy) of the 1500 images correctly. The CNN Classifier was able to classify 98.33% of the 300 cross validation images correctly. Out of all the images with a true classification of 1 (recall), the non-NN method was able to classify 58% correctly and the NN method was able to classify 95.7% of these images correctly (Fig. 11). Out of all images that were predicted 1 (precision), the non-NN method was able to classify about 71.34% of the images were classified correctly, while the NN method was able to classify 99.88% correctly.

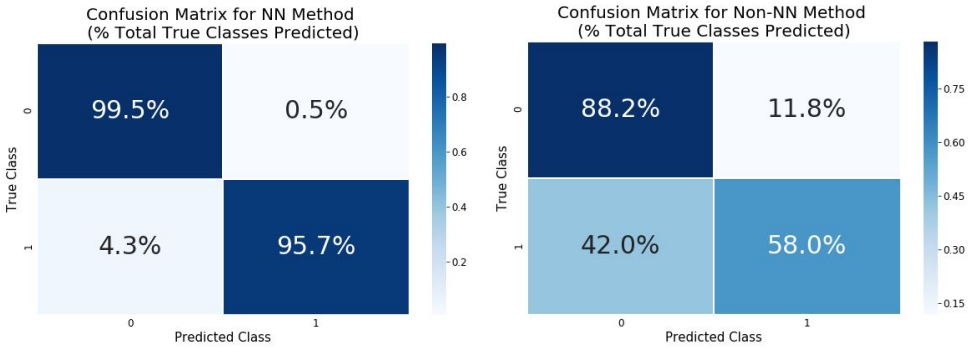

Figure 11: Confusion matrix of both CNN classifier and SVM classifier performance as a percentage of total true class.

While both algorithms were able to classify a majority of the images correctly there were multiple instances where incorrect classification occured.

5.2: Mis-Classifications of Non-NN Method

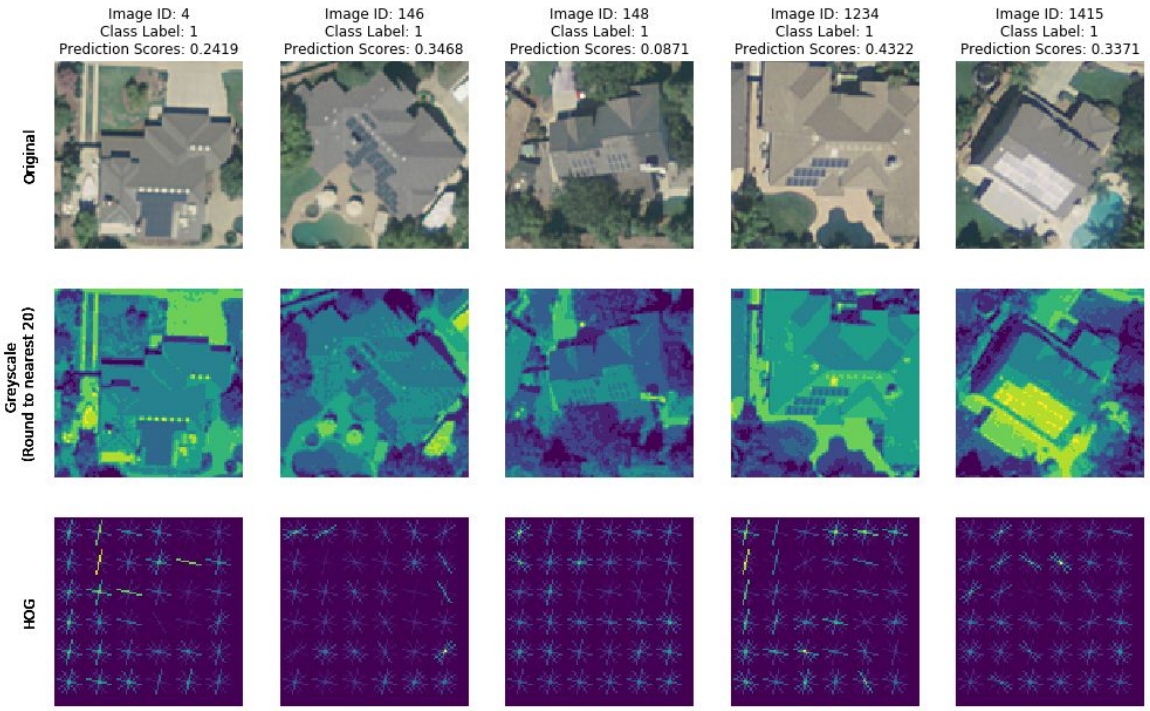

Figure 12: False negatives from SVM classifier and feature extraction. The original images are shown with caption of image ID, actual class label, and prediction scores on top (top row). Images after grayscale and pixel color rounded down to their nearest 20 to increase contrasts (middle row). After HOG feature extraction, the output images are shown (last row).

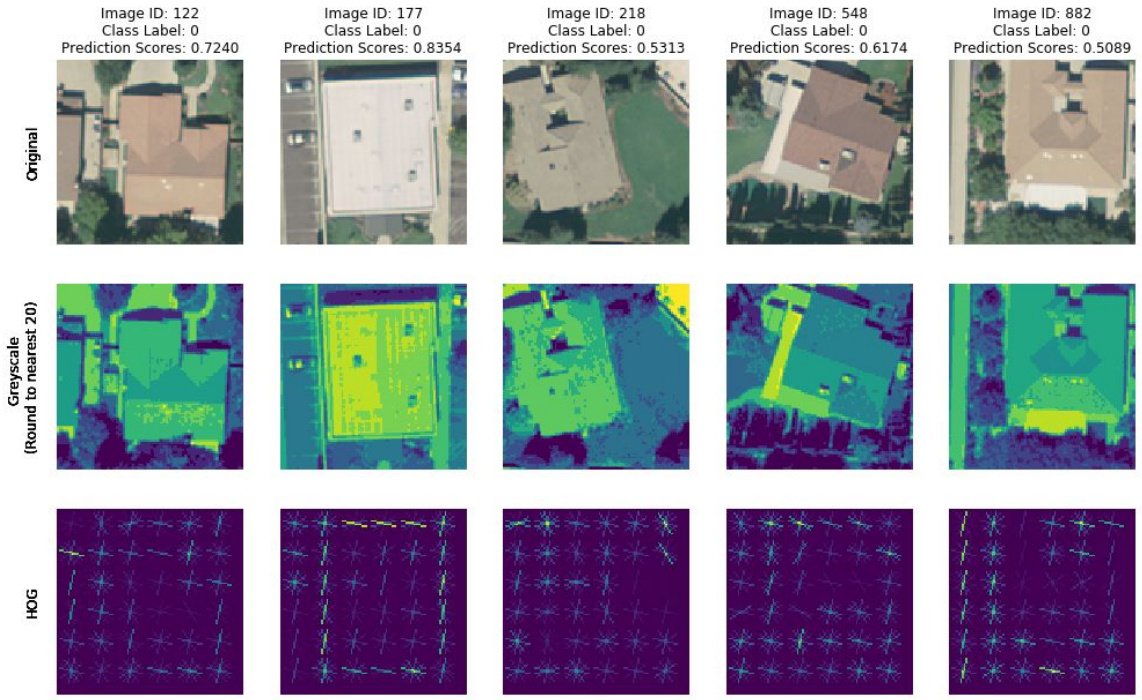

Figure 13. False positives from SVM classifier and feature extraction. The original images are shown with caption of image ID, actual class label, and prediction scores on top (top row). Images after grayscale and pixel color rounded down to their nearest 20 to increase contrasts (middle row). After HOG feature extraction, the output images are shown (last row).

Figure 12 and Figure 13 depict false negatives and false positives based on SVM classifier prediction scores, with threshold. In particular, the top rightmost image in Figure 12 shows white solar panels which is unusual. The idea of a white solar panel is counter-intuitive as the color white reflects all visible wavelengths. We hypothesize that the white solar panels seen here are actually due to solar reflection and overexposure. Correctly classifying an image like this is extremely difficult as it is such a rare anomaly in our training data.

5.3: Classification Improvement on Non-NN Method by NN Method

Figure 14 shows four examples of images where the SVM classifier produced false negatives while the neural net appropriately labeled the images as having PV arrays. We hypothesize, that in the top left image, the similarity in color of the roof and the solar arrays led to the misclassification as HOG features are based on the gradient shift with respect to colors. In the top right image, the white grid lines within the solar panel area create high color contrast with surrounding colors in very small sections. Since HOG used 16 x 16 patch (cell per block) and calculated a histogram of gradients for each cell, it might be size sensitive and not able to capture features on a more granular scale leading to failed detection.

Figure 15 depicts images where the SVM classifier incorrectly classified images with no solar arrays while the neural net was able to properly classify these images. A common detail in the top left, top right, and bottom right image are extremely dark shadows along the roof borders. We hypothesize that these shadows may have been misclassified as they would produce similar HOG features as the solar panels. On the other hand, CNN has strong and automatic feature extraction abilities which utilize deep layers and learns from many complex hidden functions.

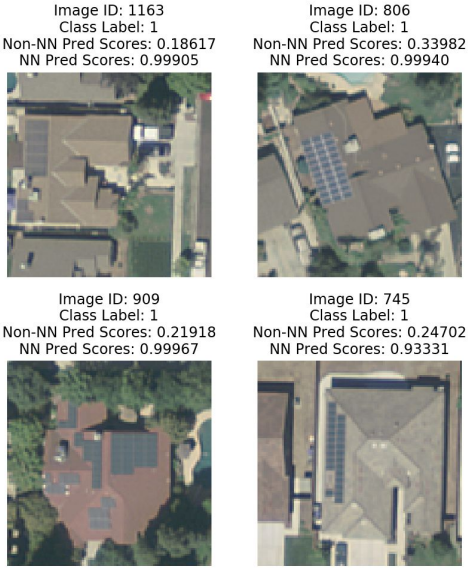

Figure 14: Example false negatives from SVM classifier while correctly classified by CNN classifier. The original images are shown with caption of image ID, actual class label, and prediction scores from two approaches on top.

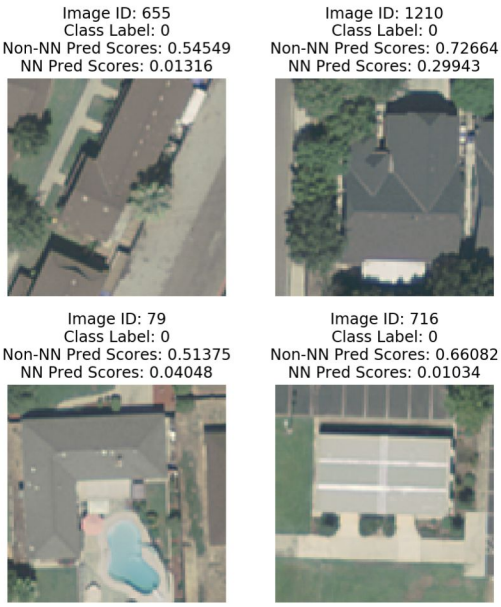

Figure 15: Example false positives from SVM classifier while correctly classified by CNN classifier. The original images are shown with caption of image ID, actual class label, and prediction scores from two approaches on top.

5.4: Mis-Classifications of NN Method

Figure 16 depicts false negatives predicted by CNN where the Non-NN correctly classified 3 of these images (top left, top right, bottom left). In particular, we suspect that the top left image is a mislabeled in the given dataset unless the PV arrays are actually located in the shadows and invisible to naked eye. In the top right image, there are two sets of PV arrays which blend in with their respective surroundings very well and may be mistaken by the CNN as parts of a rooftop instead. For both the bottom left and the bottom right image, the misclassification could be a result from the fact that the images are too small to be detected by the algorithms or maybe got truncated during preprocessing.

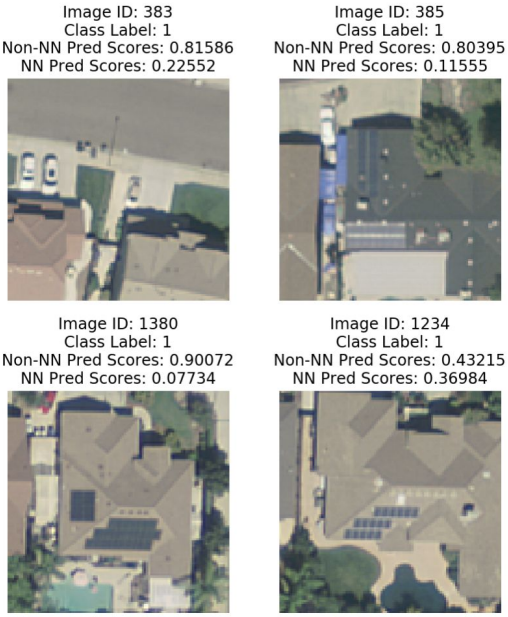

Figure 16: Four total false negative images classified by CNN classifier from validation set. The original images are shown with caption of image ID, actual class label, and prediction scores from two approaches on top.

Figure 17 depicts the only false positive in the validation set from CNN. As the CNN model has been trained on contrasting surfaces, the large area of rooftop could be mistaken as background and the white rectangular shape on the right or blue pool at the bottom of the image could have been mis-recognized as solar panels on top of the background.

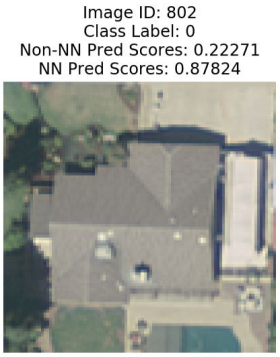

Figure 17: The only false positive image classified by CNN classifier from validation set. The original images are shown with caption of image ID, actual class label, and prediction scores from two approaches on top.

Section 6: Conclusion

6.1: Key Takeaways

We have investigated and compared two types of approaches for solar array detection in aerial imagery data; SVM classifier with HOG feature selection and deep CNN model from transferred learning. The result shows that the SVM classifier is able to achieve an ROC AUC of 0.823 and a Precision-Recall AUC of 0.729 from k-fold cross validation. While this type of manual feature selection which is focused on color contrasts and edge detection shows satisfactory metrics, the CNN based model has exhibited significant improvements over the Non-NN approach.

The CNN based model demonstrated outstanding classification performance with high generalization ability to correctly identify presence of solar panels in a fully automated mannar. The convolutional layers and the pooling layers have effectively extracted image features during the training process. We are able to achieve an ROC AUC of 0.991 and a PR AUC of 0.987 on the validation set with substantial less false detections than seen in the SVM classifier.

Based on the comparison of the performance of both approaches, it can be seen that traditional feature extraction techniques are not able to capture specific characteristics of the solar panels very well. On the other hand, the CNN classifier proved to be able to accurately identify solar panels with high performance metrics. It is able to capture unique features of solar panels successfully and generalizes well with new datasets.

In addition, with the SVM classifier, we found that occasionally, adding features helped us improve the training ROC AUC. However, when the model was tested on the test data, we saw

the AUC score decreased. The key takeaway from this is that adding features in an attempt to simply increase the training AUC would lead to overfitting that in turn would decrease the test AUC.

6.2: Future Research

For both Non-NN and CNN approaches, further improvements could be obtained. For the Non-NN model, different feature extraction techniques could be used to better encode the images and extract the relevant features. Some examples include: Laplacian edge, Sobel edge, or Canny edge detection can be attempted in the context of detecting edges of rectangular shapes of solar panels and avoid non-target edges.

Regarding the non-NN approach, grid search on the hyperparameters of both HOG and SVM can be considered to further optimize the performance, including adjustment on the cell size, block size and number of orientations for HOG, and the type of kernels and gamma values for SVM.

Regarding the CNN-based approach, several possible improvements can be considered on top of the existing model, including adjustment on data augmentation to introduce more variance to the limited data, and the addition of deeper layers within CNN architecture for better training. To avoid possible overfitting, techniques such as dropouts and L1 or L2 regularization should be applied.

It should be noted that we cannot pinpoint which part of the images that our CNN model recognized to classify as solar panels within true positive predictions. In such scenarios, the model might take part of a dark rectangular shaped rooftop as a solar panel instead of the real one on a positive labeled image. Likewise, we are not able to formulate possible reasons for misclassifications due to limited data.

Lastly, it would also be helpful to obtain more information of the solar panels such as their capacity or types for better approximation of energy generation to provide energy production estimates for energy organizations [22].

6.3: Challenges Faced that Lead to Solve New Problems

For SVM classifier, one challenge faced was determining what the cell and blocks sizes should be for HOG feature extraction. One method that could have been used was to run grid search with different cell and block sizes to see which configuration would encode the data the best.

Another challenge faced was the issue of models taking a significant amount of time to run. Because of this issue, it restricted our time and thus reduced the number of techniques and models that could have been used on the dataset. A solution to this would have been to train the model on smaller sizes of data first and then make further decisions based on previous results.

Section 7: Roles

Preprocessing of the data was done by all members of the group: Nathen Warren (NW), Yiran Chen (YC), Jiayue Xu (JX), Ravitashaw Bathla (RB), and Abhishek Baral (AB). Each of the members tried different preprocessing techniques and reported their results as to what method of preprocessing provided the best results for classification. For example, SIFT with flattened vectors as features were performed by RB, JX, and YC. Another method of preprocessing was using HOG which was performed by JX, NW, and AB.

Moving to classification methods, for CNNs, a CNN with dropout was created by JX and RB while YC used transfer learning. For non neural net algorithms, multiple combinations of classifiers were used by JX, AB, NW. Lastly, for the manuscript all parts were shared equally.

References

[1] Canziani, A., Culurciello, E., & Paszke, A. (2017, May). Evaluation of neural network architectures for embedded systems. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1-4). IEEE.

[2] Chen, D., Cao, X., Wen, F., & Sun, J. (2013). Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3025-3032).

[3] Chi, M., Feng, R., & Bruzzone, L. (2008). Classification of hyperspectral remote-sensing data with primal SVM for small-sized training dataset problem. *Advances in space research*, *41*(11), 1793-1799.

[4] Cortes, C., Jackel, L. D., Solla, S. A., Vapnik, V., & Denker, J. S. (1994). Learning curves: Asymptotic values and rate of convergence. In *Advances in Neural Information Processing Systems* (pp. 327-334).

[5] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.

[6] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (Vol. 1, pp. 886-893). IEEE.

[7] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.

[8] Duan, K., Keerthi, S. S., & Poo, A. N. (2003). Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, *51*, 41-59.

[9] Electric Power Monthly. 2016.

[10] Galinato, G. I., & Yoder, J. K. (2010). An integrated tax-subsidy policy for carbon emission reduction. *Resource and Energy Economics*, *32*(3), 310-326.

[11] Hastie, T., Rosset, S., Tibshirani, R., & Zhu, J. (2004). The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, *5*(Oct), 1391-1415.

[12] Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(02), 107-116.

[13] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[14] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[15] Kumar, G., & Bhatia, P. K. (2014, February). A detailed review of feature extraction in image processing systems. In *2014 Fourth international conference on advanced computing & communication technologies* (pp. 5-12). IEEE.

[16] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

[17] Li, X., Wang, L., & Sung, E. (2008). AdaBoost with SVM-based component classifiers. *Engineering Applications of Artificial Intelligence*, *21*(5), 785-795.

[18] Lindeberg, T. (2012). Scale invariant feature transform.

[19] 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.

[20] Malof, J. M., Li, B., Huang, B., Bradbury, K., & Stretslov, A. (2019). Mapping solar array location, size, and capacity using deep learning and overhead imagery. *arXiv preprint arXiv:1902.10895*.

[21] Malof, J. M., Hou, R., Collins, L. M., Bradbury, K., & Newell, R. (2015, November). Automatic solar photovoltaic panel detection in satellite imagery. In *2015 International Conference on Renewable Energy Research and Applications (ICRERA)*(pp. 1428-1431). IEEE.

[22] Malof, J. M., Collins, L. M., Bradbury, K., & Newell, R. G. (2016, November). A deep convolutional neural network and a random forest classifier for solar photovoltaic array detection in aerial imagery. In *2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA)* (pp. 650-654). IEEE

[23] Net Generation from Renewable Sources: Total (All Sectors), 2006-February 2016. 2016.

[24] Park, J., & Sandberg, I. W. (1993). Approximation and radial-basis-function networks. *Neural computation*, *5*(2), 305-316.

[25] Sari, A., & Akkaya, M. (2016). Contribution of renewable energy potential to sustainable employment. *Procedia-Social and Behavioral Sciences*, *229*, 316-325.

[26] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85-117.

[27] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[28] Vincent, L. (1993). Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *IEEE transactions on image processing*, *2*(2), 176-201.

[29] Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, *2*(1-3), 37-52.