Project Report

On

Solar PV in Aerial Imagery

By: Team 5

Table of Contents

1. Abstract

This should be the one paragraph that captures the significance of what you did and why you did it.

Several models were created to classify whether images of houses contain a solar panel or do not contain a solar panel. Two models were determined as effectively classifying the images into the correct categories. The first was a Histogram of Oriented Gradients Support Vector Machine, with an AUC on the test set of 84%. The second model was a Computational Neural Network, with an AUC on the test set of 99%. Together, these models provide an effective tool to allow for the classification of residential homes as either having or not having solar PV arrays. These tools can be used to further understand the distribution of solar panels in residential areas, allowing governments or other agencies to examine the effectiveness of their policy proposals as related to alternative energy and climate change.

2. Introduction

Global climate change is a serious problem with far reaching, long term consequences to all nations and economies [1]. Most energy generation in the world comes from fossil fuels, the byproducts of which are implicated in causing the current phase of global warming. Much research has gone into looking for alternative sources of energy, including wind and solar power, to try and reduce these harmful byproducts.[2] However, these technologies still continue to be very expensive compared to more traditional means of energy generation, such as coal plants. To try and counteract this, many governments, including the government of the United States, have funneled money into both alternative energy research and alternative energy production. In the US, many states also have programs designed to increase solar PV energy, such as the Duke-Energy Solar Rebate Program.[3] Despite numerous government funds, these technologies still only constitute a minority of energy generation in the US; the US Energy Information Administration states that renewables only accounted for 17.5 percent of energy generation in 2019.[4] The ultimate goal of this project will be to help the government inform best policies to encourage a broader use of renewable energy sources, particularly as applied to residential consumers. To that end, this project will focus on residential consumer's use of solar PV arrays on their houses. This is an attractive research group because it can measure both popular support for renewable energy sources (as opposed to industries complying with federal law) and the effect of government policies to encourage the use of renewable energy sources.

Specifically, this project seeks to use satellite imagery to computationally determine if a house has a solar PV array. This can help governments to determine if their policies of financially supporting solar PV arrays are yielding more solar panels on private homes. This process can then be used in conjunction with other studies to determine if government policies concerning residential solar PV arrays are effective at increasing the use of solar panels.

3. Background

Computer Vision has been a growing field in the past several decades. Recently, with the combination of feature extraction and machine learning methods, the accuracy of the models that allow computers to be able to "see" and identify specific objects from pictures has risen dramatically. This project seeks to use these methods to recognize solar panels from satellite images.

1.1 Solar panel identification from satellite images

Solar panel identification is beneficial for both government and private organizations since they can make energy policies or marketing plans by analyzing the solar development patterns. There are already some previous studies applied to machine learning methods to identify solar panels from satellite images. For example, "The DeepSolar Project" conducted by Stanford University, "constructed a comprehensive high-fidelity solar deployment database for the contiguous U.S." [4] The purpose of the DeepSolar project is to create a database for people who are interested in how the solar panels are installed in the different area in the U.S. to conduct further analysis or research. The team applied the classification method based on Google Inception V3 to identify whether there is any solar panel in the piece of the image.[4] If an image is classified as having a solar PV array, segmentation using the CAMs method would be conducted to estimate the size of the solar panels.[5] Though the purpose of the project is slightly different from this current project, the basic steps are the same as this project- detect solar panels from aerial images. Image processing and image classification are other important parts of this project to identify solar panels from the images.

1.2 Image Processing

Image processing is a subfield of signal processing, which uses computers to process digital images. This has been studied for decades since the 1950s.[6] Since the digital images are represented using matrices, this allows scaling, color conversion, image enhancement, and other useful methods to be applied to the image by adjusting the value in the matrices. Moreover, image processing can be used to filter out the information from the high dimensional features of the images. In a study of detecting plant diseases, image processing was applied to the image, for example, transforming the color into greyscale in order to filter out extraneous information then conduct image classification.[7] Similar image processing methods are conducted in this project to prepare training data for the chosen image classification models.

1.3 Image Classification

Image classification "refers to a process in computer vision that can classify an image according to its visual content".[8] However, because of the high dimensions of features of the images, it is computationally infeasible to use all these features to train models. Moreover, "look[ing] at high-resolution images it is very likely that a neighboring pixel belongs to the same land cover class as the pixel under consideration."[7] Therefore, "implementing feature extraction, and selecting suitable variables for input into a classification procedure are all important" [9] since we can make full use of the features and also reduce the dimension of the data. Some image processing methods such as Histogram of Oriented Gradients (HOG) [10] or Scale-Invariant Feature Transform (SIFT)[11] are famous as tools to extract informative features from the original image. With these filtered features, researchers can apply supervised machine learning methods to conduct image classification with high accuracy compared to using original images. One of the frequently used methods that can be applied to image classification are Support Vector Machines (SVM).[12] In the study of image classification, it is stated that SVM "can generalize well on difficult image classification problems where the only features are high dimensional histograms," by comparing the performance with KNN-based models, tuning the SVM model itself, remapping the input data.[13] The analysis process of this project would be similar to the previous work since the goal of this project is to find a good combination of transformed original data and a well-tuned model to identify solar panels in the image with high accuracy.

4. Data

The data is primarily comprised of satellite imagery of rooftops with or without a solar photovoltaic array(s). However, images without any rooftops/houses with solar array(s) installed were also part of the dataset.

Moreover, the solar array(s) had different types of structural configuration (or arrangement) and orientation across the images in the data. The images in the dataset had a resolution of 101 x 101 pixels. And, the data was provided in two sets, training, and testing data.

4.1 Data Description

The *Training* data is comprised of 1500 labeled satellite images comprising of 505 images with solar photovoltaic array(s) on the rooftop, and 995 images without any solar photovoltaic array(s). Figure 1 shows the distribution of images for each class i.e. solar and non-solar

The *Testing* data comprised of 558 unlabeled satellite image data with and without solar photovoltaic array(s).

Figure 2 shows sample images *with* solar PV array(s) installed in different locations, configurations, and orientations having different intensity in color associated with the panels in the array. Figure 3 shows sample images without solar PV array(s).

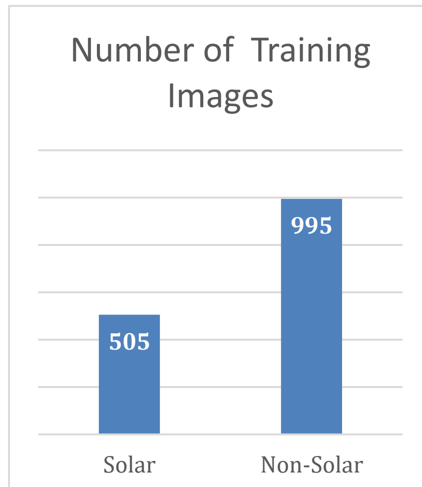


Figure 1: Number of Training Images for each label

Figure 2: Examples of images with Solar PV array(s)

Figure 3: Examples of images without Solar PV array(s)

4.2 Challenges associated with the data

The primary challenge associated with image classification for this dataset was identifying certain edges and colors which were specific to solar array(s). Since the array(s) were mounted on the rooftop, which had similar edges to the array(s), distinguishing roof and arrays was the main challenge. Another interesting challenge with detecting the images was identifying array(s) which had different colour intensity and arrangement, distinguishing them from the images of similar structures such as the top of a car in similar color, or roads and pathways. Another critical problem with image data is high dimensionality. Colored images with 101 x 101 pixels correspond to 30,603 features, which is considered as very high dimensional data, and will be computationally expensive when directly used in a machine learning algorithm. Dimensionality reduction is an important step when working with image data for faster computation.

One of the many traditional ways of solving the problem of feature extractions from these kinds of images is manipulation of the colour scheme to increase contrast between the object and noise. Contrast helps in distinguishing features (in other words the RGB or Grayscale value) of the object from the noise around it. Subsequently, dimensionality reduction methods such as Principal Component Analysis (PCA) could be used to reduce the dimensionality of the image features. Interestingly, resizing of images is a much faster way to reduce dimensionality, but it could lead to loss of important information from the image. Another method which is being widely used for dimensionality reduction and feature extraction with edge detection is Histogram of Oriented Gradients (HOG)[10] which is discussed more in the coming sections.

Convolution is another widely use process for feature extraction and dimensionality reduction, primarily with conjunction with neural networks for image classification. The process of convolution is also discussed in the methods and results sections.

5. Methods

In order to carry out the primary objective of this project to identifying and classify images with solar photovoltaic array(s), different machine learning models were created including simple logistic regression, random forest classifier, support vector machines (SVM), and Convolutional neural networks (CNN). Prior to classification, several methods for feature extraction and dimensionality reductions such as PCA, HOG and Convolution were also used.

5.1 Support Vector Classification using Histogram of Oriented Gradients (HOG) processed images

Following the classical approach of image processing, images were pre-processed in terms of color, orientation and rescaling. Following which, feature extraction with edge detection and dimensionality reduction was performed using HOG. And finally, the extracted features were used to train a SVM model for identification of images with solar array(s). *Figure 4* shows the steps in image processing and classification.

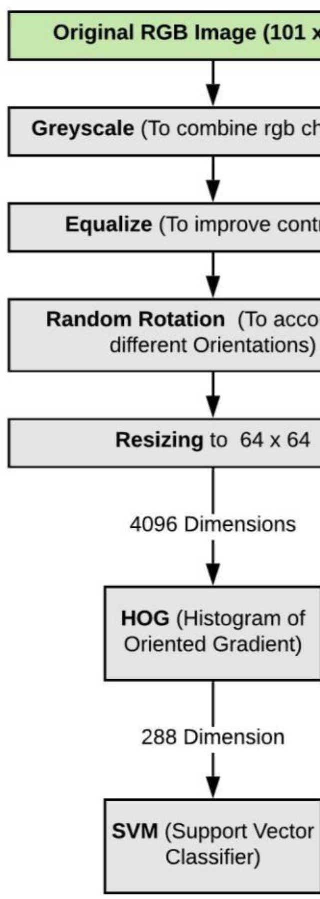


Figure 4: Process of image pre-processing for SVM

5.1.1 Image Pre-Processing using Pillow

Images were pre-processed using *Pillow* library in python (*Figure 5*). All the colored training images were converted to grayscale to reduce the dimensionality from 3 channel to 1 channel. Subsequently, images were equalized to increase the contrast between solar arrays and surroundings, following which the images were randomly rotated to capture different orientations of solar arrays. Finally, the images were rescaled from 101 x 101 pixels to 64 x 64 pixels for the purpose of dimensionality reduction.

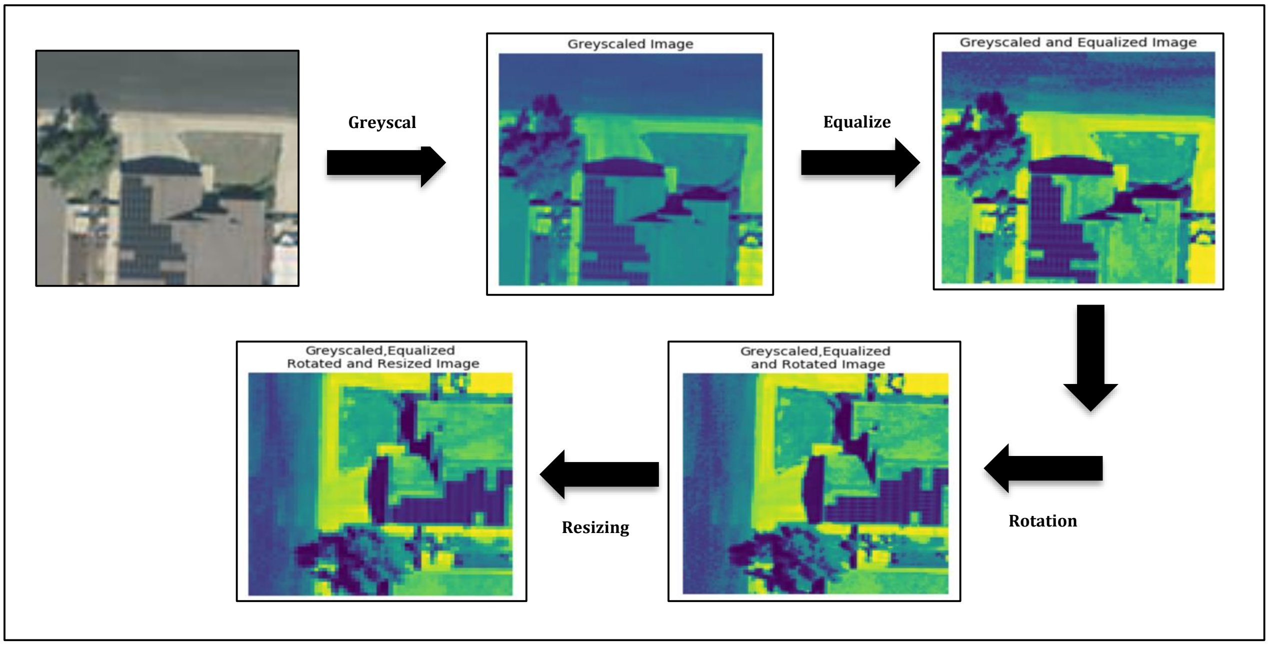


Figure 5: Image pre-processing steps

5.1.2 Histogram of Oriented Gradients (HOG) for edge detection

Histogram of oriented gradients (HOG) is a feature descriptor used to detect objects in computer vision and image processing. A feature descriptor is a representation of an image or an image patch that simplifies the image by extracting useful information and throwing away extraneous information. The HOG descriptor technique counts occurrences of gradient orientation in localized portions of an image - detection window, or region of interest (ROI). Localized portion of an image can be described as a block of consecutive pixels of the image, where the block moves over the entire grid space of an image.

In simple terms, HOG generates a histogram of orientations (which correlates to the change of intensity of corresponding pixel values in a block) for a block. A block is made of cells and each cell overlays certain number of pixels of an image. The orientations capture the direction of edges in an image, thus generating a feature space portraying the outline of an object in the image. The block moves across the image and generates histograms for a group of cells, thus reducing the dimension. *Figure 6* shows the images distributed in blocks, and gradient value and direction associated with each block.

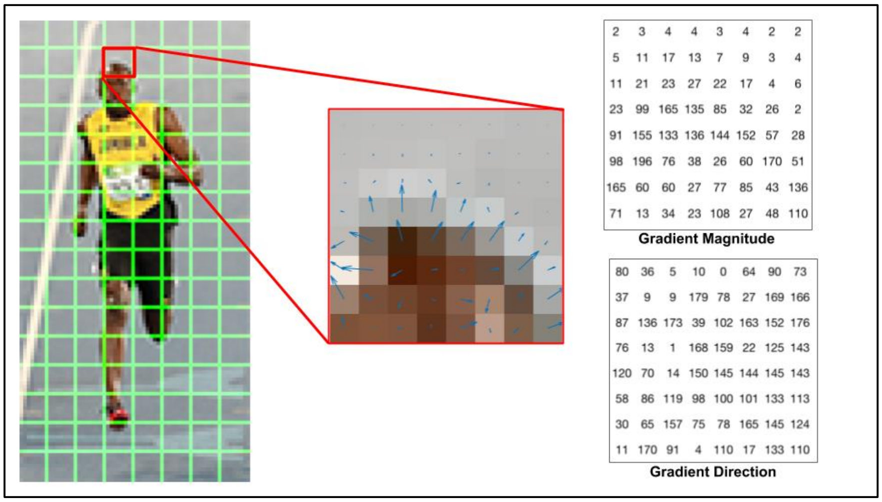


Figure 6: Blocks of images represented in terms of gradient direction and magnitude. (Source: www.learnopencv.com/histogram-of-oriented-gradients)

The 64 x 64 pixels images were processed using HOG, evaluating 16 x 16 pixels per cell with 4 cells (2x2) per block. The gradients based on their direction were binned into 8 bins for each cell and frequency of the histogram from the 4 cells (2x2) in a block were normalized using L2 norm, generating 8 features per block. Normalization is a key step which accounts for lighting variations across the image. Figure 7 shows the HOG features with oriented vector (which could be matched with corresponding edges of the original image) for a few training images.

5.1.3 Classification using Support Vector Machine (SVM)

For classification of images turned feature vectors, a Support Vector Machine or SVM classifier was used. SVM is a non-probabilistic linear classifier, and works very well with high dimensional data, which in this case was the best choice, with feature space of 288 dimensions.

An SVM classifier with the default ‘rbf’ kernel, from the SVM library by Scikit learn was used for classification. Moreover, in order to check the generalization performance of the model, a 10-Fold Cross Validation was performed using the entire training data of 1500 pre-processed images. Subsequently, model performance was evaluated in the using the test data of 588 processed images.

Model performance was evaluated using Receiver Operating Characteristic (ROC) curve with Area Under the curve (AUC), Precision-Recall curve (PR) with Average Precision value. However, since the data was un-balanced, F-1 score was computed to evaluation of model performance. Model performance is further discussed in the Section 6.

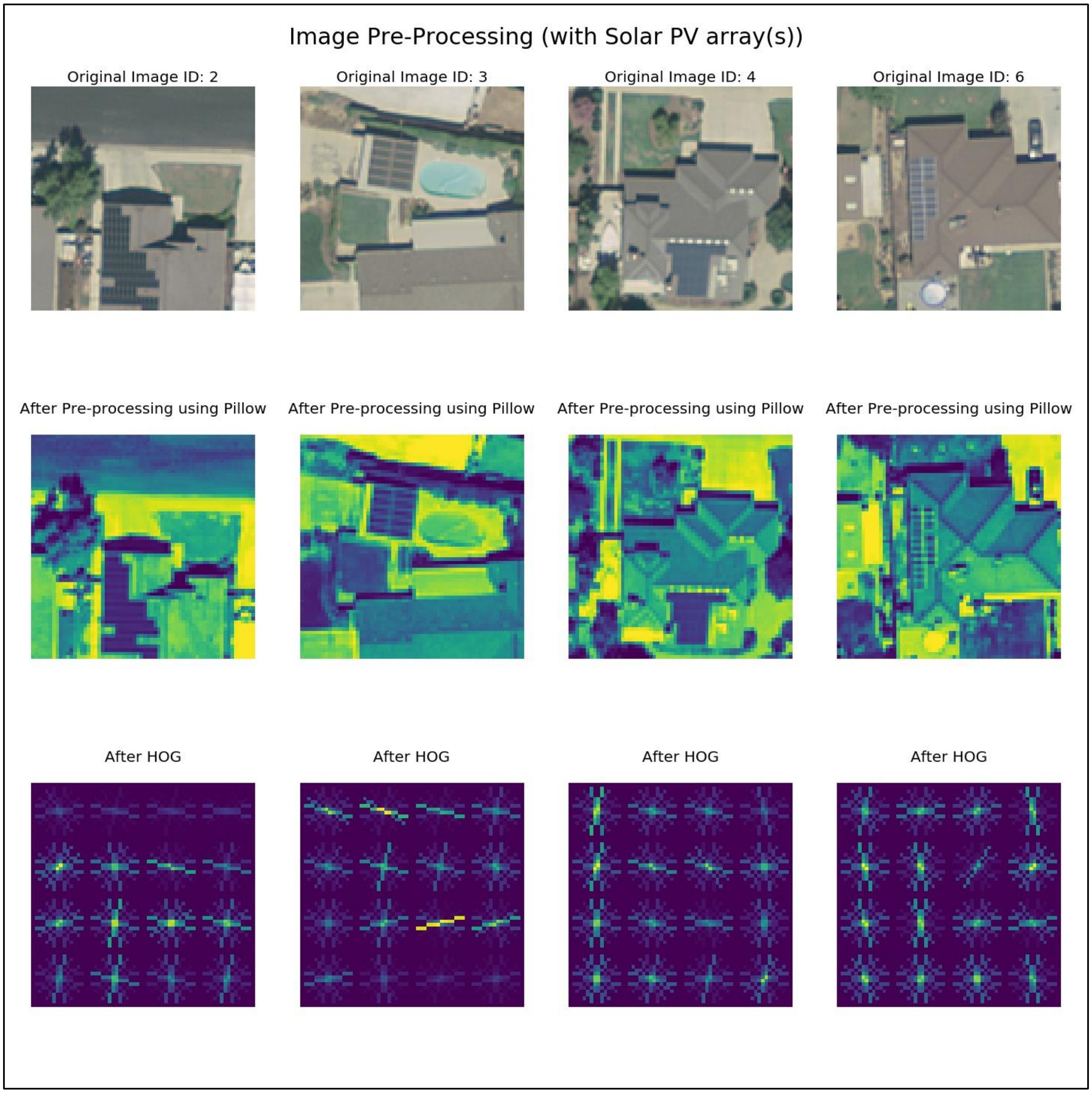


Figure 7: Image classification steps for SVM

5.2 Convolutional Neural Network (CNN)

For classification of images, another powerful classifier, Convolutional Neural Network or CNN was used. CNN is a class of deep learning network which is widely used to analyze images. In this project, our CNN model achieved high accuracy without applying feature extraction before training the model. Only normalization was conducted to the matrices that represent the images. For training the convolutional neural network, a random split of 80% to 20% of the training dataset was used results in 1200 samples for training and 300 samples for validation.

5.2.1 Structure of Convolutional Neural Network (CNN)

The CNN model architecture was constructed below to give a visual idea of the underlying process. One set of Convolution + ReLU and Max Pooling is one layer. In our CNN model, input data would go through 5 layers before reaching the stage of Flatten.

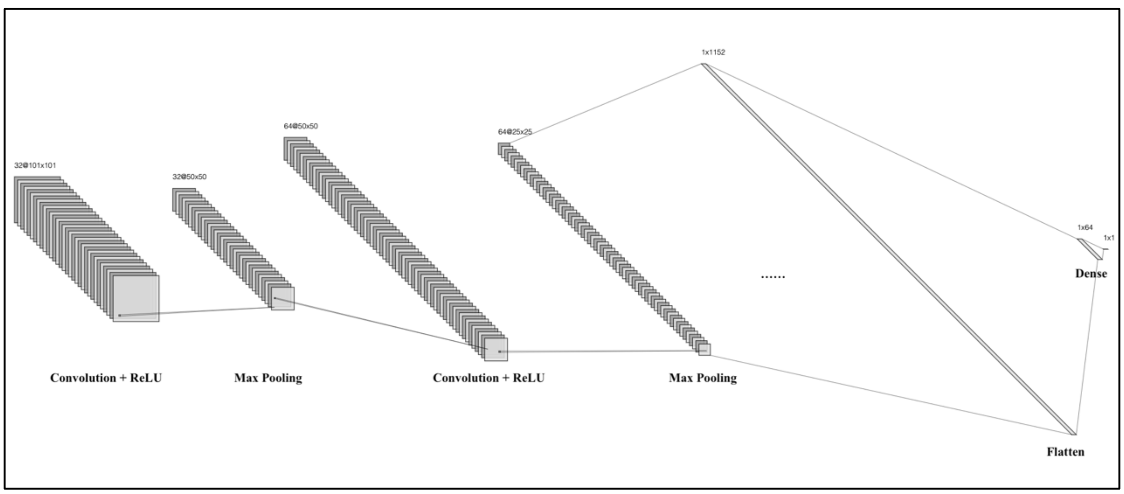


Figure 8 This architecture graph shows the first two convolution layers with ReLU and max pooling. As mentioned above, the process begins with 32 filters of the 101 by 101 input matrix, then max pooling of 2x2 frame reduces the dimensionality to 50x50. Another convolution layer was applied following with max pooling. Then additional 3 convolution layers applied following with max pooling and 50% dropout to prevent overfitting. After applying all the layers, the output matrix was flattened to a 1 dimensional matrix and reduced to 1x64 by ReLU. The last layer was to reduce the 1x64 matrix using sigmoid function (because of binary classification) to get the output result.

After normalizing the input image pixel values into a matrix, the convolution of the input matrix and the filter matrix (also called kernel) is computed starting with placing the filter matrix over the input image at the top left corner of the input image, then sliding the filter matrix by 1 pixel and for every position onto the input image. For each position, element-wise multiplication is computed and summed to get a single value. The function of the filter matrix is for feature detection over the input image.

It is required that the filter matrix must be a squared matrix and the dimension must be odd integers and it was recommended that images with less than 128 by 128 pixels to have kernel size equal or below 3. However, the weights in the filter matrix is not something that can be controlled here. The assumption is that the weights are unknown and the model itself will find its' own weights by randomly initializing the filter matrix. Rectified Linear Unit is then applied as the activation function here to learn non-linearity. It would replace all negative values with zero in the output matrix.

After feature extraction using the filter with ReLU operation, Max pooling is performed as a 2 by 2 matrix on the output matrix. The process starts with only looking at the 2 by 2 input starting at the upper left corner of the output matrix and select the maximum value with the 2 by 2 frame. Then sliding the 2 by 2 frame in the same manner as the filter matrix did, and an output matrix will be created with smaller dimension. The goal of max pooling is to reduce dimensionality while still retain the most important information.
After 5 times of the process above, flatten and dense is used respectively to reduce the output layer to 1 dimension and to a single value.

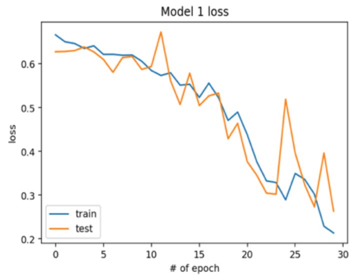
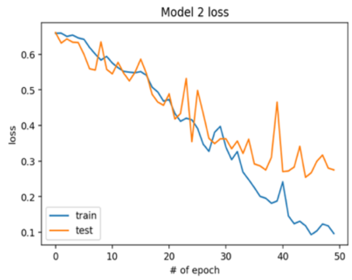
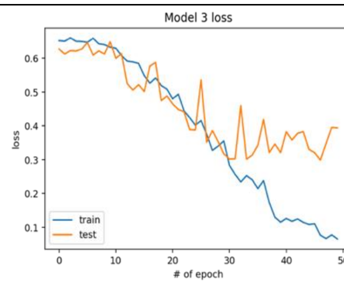
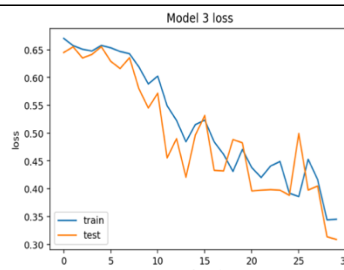
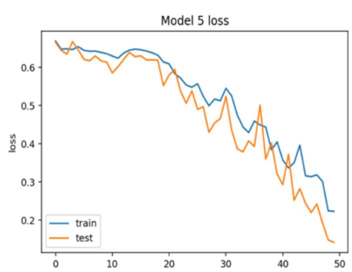
5.2.2 Padding & Dropout

Padding and dropout was applied to some layers in our CNN model. Padding was used for all the convolution layers in the model for two reasons. First, it would allow the output matrix size to be the same as the input matrix size. Second, the pixels on the edge of the input matrix would be applied by the filter the same number of times as the pixels around the center of the input matrix. Information from the edge of input matrix would be captured with the same level of certainty.

Dropout was used for the 3rd, 4th and 5th layer of the model for the purpose of preventing overfitting. The concept was to set a random portion of the input units (in this case 50%) after max pooling to zero. According to Srivastava's paper, dropout has proven to be the most effective way over other regularization method to prevent overfitting in neural networks. The reason to use dropout only after the first 2 layers is because information was captured. [16]

5.2.3 Model training

The Convolutional neural network was trained using different combinations of hyperparameters. The following table shows the process of choosing the final model parameters.

Model	# of layer	# of epoch	Dropout	Padding	Training Accuracy	Validation Accuracy	Learning Curve
1	4	30	50% of 3rd Layer	No	0.9275	0.9167	
2	4	50	50% of 3rd Layer	No	0.9667	0.9100	
3	4	50	50% of 3rd & 4th layer	Yes	0.9792	0.8567	
4	5	30	50% of 3rd & 4th layer	Yes	0.8475	0.88	
5	5	50	50% of 3rd 4th & 5th	Yes	0.9040	0.9533	

With 4 layers of convolution neural nets and 30 epochs, the model is doing well with just slight overfitting. When Increasing the epoch to 50, clear overfitting is showing on the learning curve graph. Testing set loss discontinued to decrease around epochs 30. A third model was tested with adding a dropout at the 4th layer to prevent overfitting, and also added padding. The model's performance was worse than the second model, with a clear separation of training and testing loss at epochs of 30. This could potentially imply that with 4 convolution layers, the best it can do is with 30 epochs without overfitting the data. In order to try to boost the overall accuracy, a fifth layer was added to the model and again tested starting with 30 epochs, padding, and dropout of 3rd and 4th layers. The testing accuracy was around 88% only and it was 4% above the training accuracy. This indicates no overfitting was occurred and probably more epochs could be added to test what point overfitting could occur while keep trying to increase accuracy. The fifth and final model was showing at the last row of the table above. As one can see both the training and validation loss curve continues to go down as it reaches the 50th epoch. The validation curve was also below the training curve showing no problem of overfitting and both the training and validation accuracy was very optimal. The final model was again training over all of the 1500 samples and generated a prediction for the testing set to submit on Kaggle. Since each time CNN initialize different random filter matrix, the results would be slightly different. It achieved around 96% accuracy of the test data set on private leaderboard.

6. Results

Overall, both the models SVM and CNN performed very well in identifying images with solar PV array(s). However, as expected the Convolutional Neural Network outperformed every other model used for classification. Following is the evaluation of model performance of both the models.

6.1 Model performance for SVM

Confusion Matrix for SVM		
	Predicted Solar	Predicted Non-Solar
Solar	291	214
Non-Solar	97	898

After image pre-processing followed by HOG, SVM classifier performed with an **AUC of 0.84** over the training data when cross validated to check model generalization performance, and **0.86** over the test data. As can be seen from the confusion matrix, model performed relatively better in identifying images without solar array(s). However, the model was able to correctly identify 60% of the images with solar PV array(s).

Further analysis of the classified images revealed several interesting insights. Based on the visual inspection of the True positives, False positive and False negative images (*Figure 8*), following conclusion can be made about the model.

- The model was able to capture the array(s) which had a good contrast with the background/surface on which they were installed and also had a linear structure.
- However, the model was misclassifying the images as positive class (with solar) which contained shadows (which have a same appearance as that of solar array(s) and also had repeated rectangular edges).
- The model failed to identify images having low contrast between the solar array(s) and the background or had a whiter boundary in the solar array(s), which was giving the same appearance as that of a rooftop.

Model performance was visualized using ROC curve and Precision-Recall curve (see model comparison). Also, since the data was unbalanced, F-1 score was also evaluated to check the model performance. The model had an F-1 score of 0.78.

Figure 9: Classification performance of SVM showing images classified as True Positive, False Positive, and False Negatives (at threshold 0.5)

6.2 Model Performance for CNN

Convolutional neural networks performed extremely well as compared to the SVM model. The CNN had an AUC of **0.99** for the training data**, 0.97 for the test data**. As can be seen from the confusion matrix, the model had a high precision, with only 5 false positives. Also, as compare to SVM, CNN was correctly able to identify almost 75% of the images with solar array(s).

Confusion Matrix for CNN		
	Predicted Solar	Predicted Non-Solar
Solar	398	107
Non-Solar	5	990

Further analysis of the classified images revealed several interesting insights. Based on the visual inspection of the True positives, False positives and False negative images (*Figure 9*), the following conclusions can be made about the model.

- The model was able to capture different arrangement of solar array(s) with different orientations and shadows.
- However, the model did falsely classify images with blue colored objects arranged closely, blue surfaces or images with regular rectangular arrangements as a positive class.
- Also, it appears the model was not able to identify images with noise such as vegetation, and images having low contrast between the array(s) and background.

Figure 10: Classification performance of CNN showing images classified as True Positive, False Positive, and False Negatives (at threshold 0.5)

ROC and Precision-recall curve was plotted for the CNN model. The model had an F-1 score of 0.92.

6.3 Comparison of model performance

As stated earlier, CNN had much better performance as compared to the SVM model. Table 1 compares and contrast the performance metrics for both the models. Interestingly, since the data was unbalanced, even though CNN had an AUC of 0.99 for training data, the true model performance metric can be measured as the F-1 score, which was 0.92.

Table 1: Performance metrics for SVM and CNN

Model	AUC Training (with CV)	AUC Testing	Average Precision	F-1 Score (for threshold 0.5)
SVM (w/ HOG)	0.84	0.86	0.74	0.78
CNN	0.99	0.97	0.98	0.92

Precision-recall curve (*Figure 10 (left)*) shows the PR curve for SVM and CNN model. As can be seen the CNN had a much higher average precision as compare to SVM. Also, the ROC curve (*Figure 10 (right)*) shows the ROC curve for the SVM and CNN models.

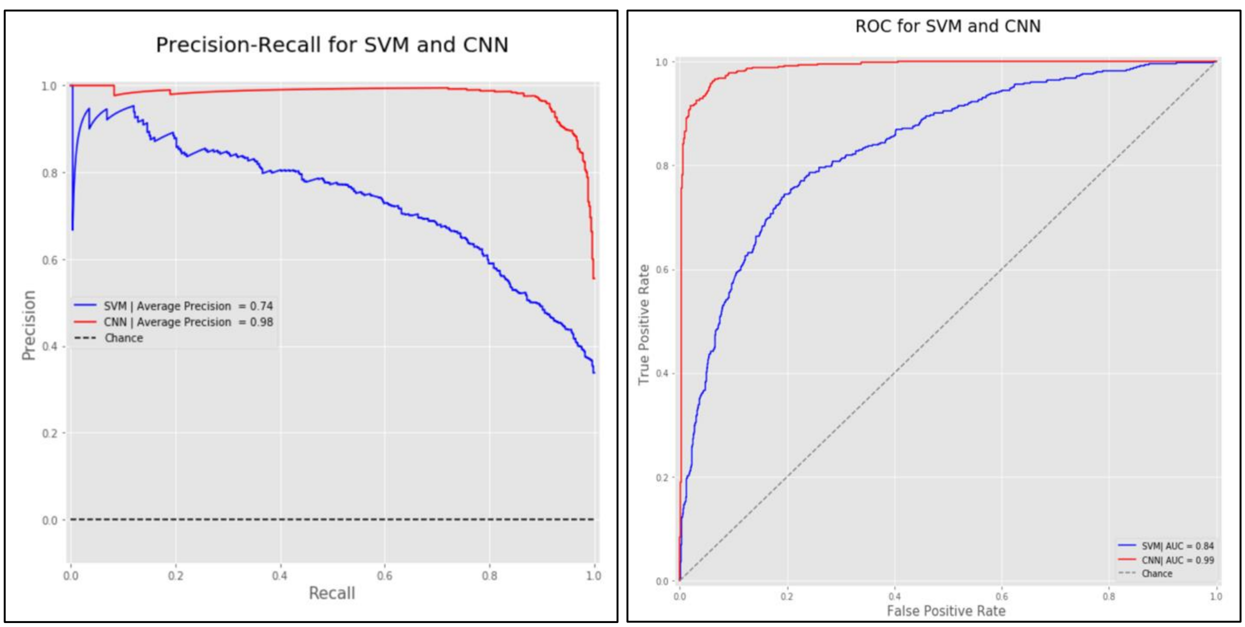


Figure 11: (Left) Precision-Recall Curve for SVM and CNN, (Right) ROC curve for SVM and CNN

The main advantage of CNN over the other models is because there is a really minimum level of feature extractions for a human needed to supervise to the model. It detects the important features itself. In the solar PV case, the features from the original pictures are 101 by 101 by 3, and the sample size available is only 1500. For the models that require human supervised feature extractions, successfully extracting the important features can be a challenging task. In addition, the image features are much larger than the sample data that is available to train. For models doing much less feature extractions, the amount of training data presented is far under the needs. With only a fraction of the training samples than the features, it becomes a hard task for the model to find the important features. Thus, CNN is outperforming the other models.

However, there were cases when both the models failed to identify few images with solar array(s). Looking at the false negative images (*Figure 11*), it appears that both the models were not able to capture the features when the contrast between solar array(s) and the background was very low.

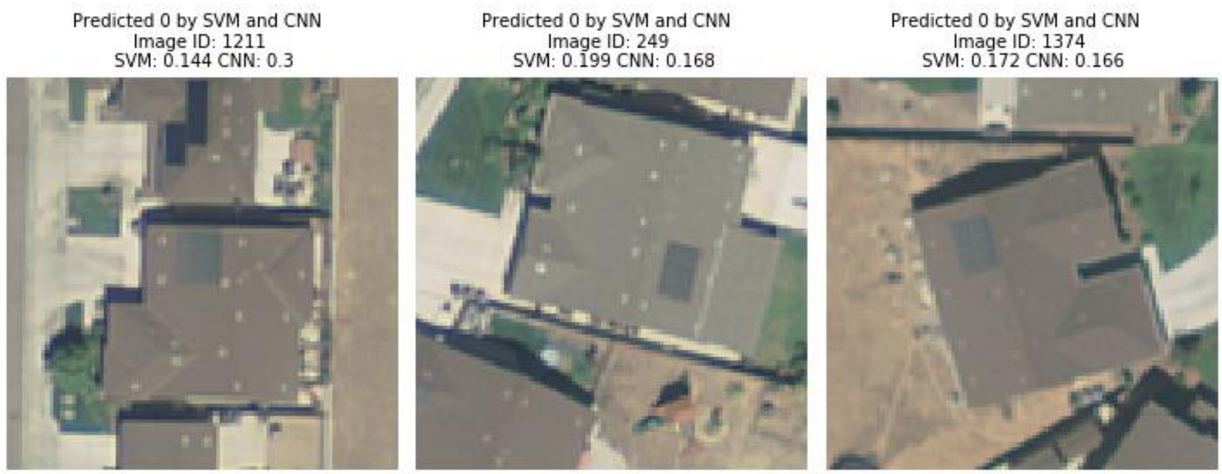


Figure 12: False negative images for both SVM and CNN

7. Conclusions

Out of the models created, the two best models were the HOG SVM and the CNN models. The HOG SVM performed acceptably, achieving an AUC of 86% on the test dataset. This model is much faster than the CNN, and could have applications where speed is necessary, but accuracy is not as much of a concern. In terms of strict accuracy, the CNN performed much better, with AUC of 96% at classifying images as correctly having or not having a solar panel on the house. The CNN clearly has an advantage over all other models at correctly classifying the images, however, it was much slower than any other model. This model would be preferred when accuracy is required, but speed is not a concern. As described in the introduction above, the solar panel image classification goal of this project could help inform policymakers or other stakeholders as they seek to experiment with and determine the best method to increase solar panel use. With an AUC of 96%, the CNN model could be used to effectively determine the efficacy of policies targeted at increasing solar panel usage. This can be done by feeding satellite images into the model, allowing any level of government to analyze their policies effectiveness. Using this information, the government can then tailor the policies to further experiment with policies and legislation, with the ultimate goal of increasing solar panel usage, reducing reliance on fossil fuels, and combating climate change. Further research could include ways to boost the accuracy and AUC of these models even further, and to apply these image classification techniques to further questions of interest.

8. Roles

Following is the descriptions of individual contribution in the project by the members to the team.

1. **Akshay** worked on image processing and used Histogram of Oriented Gradients, to develop the SVM model. Akshay also wrote the methods and results section for this model, and the data section.

2. **Melody Li** created a Computational Neural Network model for the images, which was then submitted as part of the final part of the competition.

3. **Tzu Chun** created and tested several models including KNN, CNN, and SVM combine with feature extraction methods- greyscale and Histogram of Oriented Gradients(HOG). The models are submitted as part of the Kaggle competition. Moreover, Tzu-Chun also wrote the background and revised section 5.2 Convolutional Neural Network (CNN) of the report.

4. **Derek Wales** worked with Andrew in report framing and development and made the baseline Linear regression model and applied several other methods using Random forest, and Naïve Bayes.

5. **Andrew Patterson** created and tested several models with Derek while using the grayscale component reduction. Andrew also wrote the abstract, introduction, conclusion, references, and edited other parts of the rest of the report.

9. References

1. Mora, C. et al. Broad threat to humanity from cumulative climate hazards intensified by greenhouse gas emissions. Nat. Clim. Change 8, 1062–1071 (2018).
2. Owusu, P. A. & Asumadu-Sarkodie, S. A review of renewable energy sources, sustainability issues and climate change mitigation. Cogent Eng. 3, 1167990 (2016).
3. Database of State Incentives for Renewables and Efficiency. DSIRE. Accessed March 1, 2020. https://programs.dsireusa.org/system/program?zipcode=27517.
4. What is U.S. electricity generation by energy source? - FAQ - U.S. Energy Information Administration (EIA). https://www.eia.gov/tools/faqs/faq.php?id=427&t=3.
5. Yu, J., Wang, Z., Majumdar, A. & Rajagopal, R. DeepSolar: A Machine Learning Framework to Efficiently Construct a Solar Deployment Database in the United States. Joule 2, 2605–2617 (2018).
6. Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba 2016. Learning Deep Features for Discriminative Localization. Retrived from: ”http://cnnlocalization.csail.mit.edu/Zhou_Learning_Deep_Features_CVPR_2016_paper.pdf”.
7. Lipkin, L. <strong>Picture Processing by ComputerAzriel RosenfeldAcademic Press, New York, 1969. Computer Science and Applied Mathematics. Science 169, 166 (1970).
8. Jayme Garcia Arnal Barbedo 2013. Digital image processing techniques for detecting, quantifying and classifying plant diseases. Retrived from: “https://link.springer.com/article/10.1186/2193-1801-2-660”.
9. David Kaeli, Perhaad Mistry, Dana Schaa, Dong Ping Zhang 2015. Heterogeneous Computing with OpenCL 2.0- Chapter 9.
10. Lu, D. & Weng, Q. A survey of image classification methods and techniques for improving classification performance. Int. J. Remote Sens. 28, 823–870 (2007).
11. Navneet Dalal and Bill Triggs 2005. Histograms of Oriented Gradients for Human Detection. Retrived from: “https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf”.
12. Lowe, D. G. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60, 91–110 (2004).
13. Corinna Cortes and Vladimir Vapnik 1995. Support-Vector Networks. Retrived from: “http://image.diku.dk/imagecanon/material/cortes_vapnik95.pdf”.
14. Olivier Chapelle, Patrick Haffner, and Vladimir N. Vapnik 1999. Support Vector Machines for Histogram-Based Image Classification. Retrived from: “http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.459.9821&rep=rep1&type=pdf”.
15. Bradbury, Kyle; Saboo, Raghav; Johnson, Timothy; Malof, Jordan; Devarajan, Arjun; Zhang, Wuming; et al. (2016): Modesto Aerial USGS Imagery from the Distributed Solar Photovoltaic Array Location and Extent Data Set. figshare. Dataset. https://doi.org/10.6084/m9.figshare.3385789.v1.
16. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. 2014.