Final Report for Kaggle Competition: Solar PV in Aerial Imagery

Group E - Scarlett Hwang, Bingying Liu, Joaquin Menendez, Nathan Scheperle,

Muxin Diao

A. Abstract

With the rapid growth of solar panel applications, institutions ranging from energy companies to government agencies are interested in finding the geographic distribution of solar panels. This competition is a mini project of solar panel detection with the goal of labeling the images with solar panels using supervised learning methods. Different binary classifiers were constructed to carry out the task, including a baseline model of K Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, XG-Boosting and, in particular, Convolutional Neural Network (CNN). The labeled set of 1,500 aerial images were split into a training set of size 1,350 and a validation set of 150, and an image generator was applied to enlarge the size of the training set. In validation, the CNN model reached an average accuracy of 96.3% and AUC of 0.962. In testing, the model achieved an accuracy of 99.19%. These results indicated that CNN model was an effective and efficient solution to this problem. However, further improvement is possible.

B. Introduction

Today, the use of drones is growing at a rapid pace in various industries. With machine-learning based computer vision now powering these drones, we can now predict and better make use of the images we gather into interpretable data that can be used to make decisions.

Specifically, in the energy sector, supervised learning techniques are deployed in routine inspections and maintenance by firms in the industry, a task which had been significantly costly and time-consuming prior to these methods due to the sheer area needed to be covered. Ascertaining defective or broken solar panels was an even more difficult process. A manual inspection method can only allow the inspection frequency of once in three months by in-field technicians. Additionally, due to certain terrains and environments in which the panels are installed, manual inspection also presents a safety hazard. However, through the use of machine-learning in aerial imaging, a company can now effectively evaluate a huge area in a matter of hours with higher accuracy and few safety risks.

In addition, with the trend of renewable energy usage in private property and of public policy supporting green energy, government agencies and energy companies can use this technique to appraise the status of solar panel installations in different regions. Further, it can help policy and decision makers to determine areas with particularly sparse distributions of installations when considering legislation or business strategies to expand solar panel adoption. Through this project we aim to use machine learning algorithms with aerial imagery data to show the effectiveness of solar panel detection.

C. Background

Automatic detection of solar panels in aerial imagery has been explored extensively, from traditional supervised learning approach with the goal to identify individual PV pixels and detect panel's precise size and shape, to state of art deep learning algorithms that could be cheaply applied to identify solar panels over large and even new geographic regions. Field of interest has gradually shifted from improving performance accuracy to increasing scalability and ability to adapt to new aerial imagery as well as inferring characteristics of PV arrays such as energy production and PV array capacity, etc.

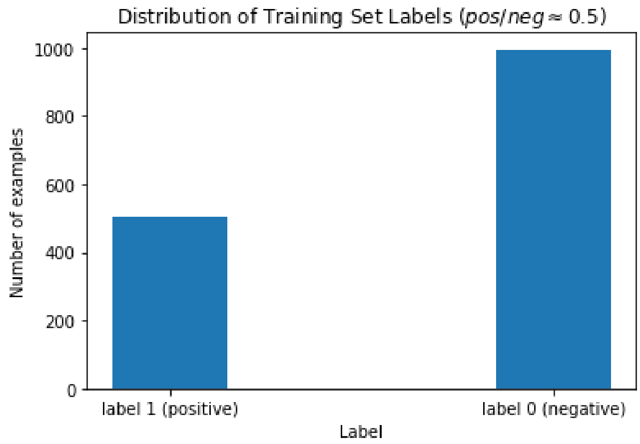
Malof et. al. (2016) transformed the original 3-channel RGB image into a feature vector. This was achieved firstly, by computing mean and variance of a 3 by 3 window surrounding each pixel. Each window centered around p0 results in 9 features and all the features were combined into a M-dimensional vector, which was the feature vector. This vector was then fed into Random Forest Classifier to output an average probability of belonging to each PV class returned from each tree. Post-processing step was applied to improve the classification accuracy by identifying local maxima and its surrounding smooth region, which not only filtered out false positives, but also helped find contiguous groups of pixels with high confidence. This method is highly effective in per-pixel basis and has established a benchmark for algorithm improvement. In addition, steps involved shows the importance of pre and post processing.

Camilo et. al. (2017) compared different performances of CNN algorithms on solar PV aerial imagery in terms of pixel-wise detection and object shape/size detection. VGG has excellent classification capabilities while SegNet (semantic segmentation) can accurately tell the true panel locations. Since the last layer of VGG network is a two-way Softmax which returns a single probability that a PV array exists, it can only determine if a solar panel is reliably identified. However, SegNet has additional "deconvolutional" layer which expands a larger feature map using the preceding layer. It returns a probability estimate at each pixel location in the original input image, therefore, with object-based scoring, it can get a relatively accurate size and shape of panel.

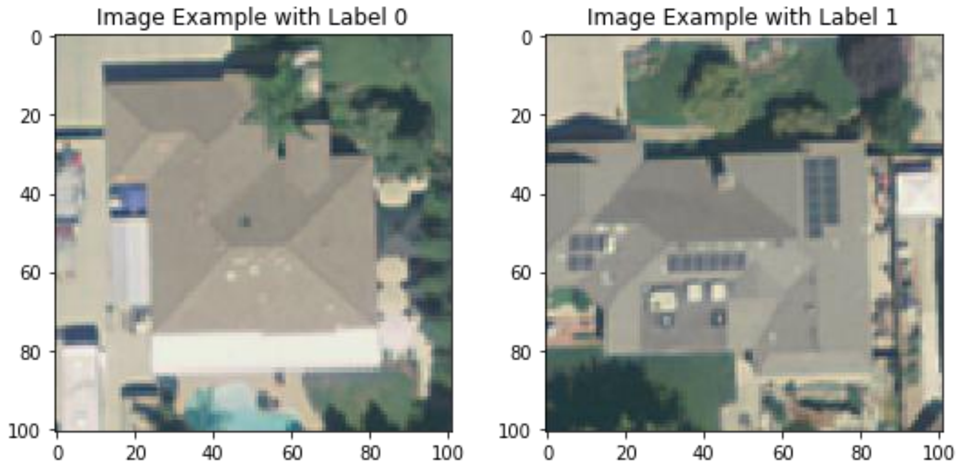
On the basis of SegNet, Malof et. al. (2019) proposed a publicly available and scalable algorithm called SolarMapper that can be applied to broader geographic regions by fine tuning the original model (with small fraction of new training data added). SegNet's ability to accurately estimate size of panel enables mapping power generation capacity over large areas possible.

D. Data

The training dataset for the project includes 1,500 digital images each labeled '1' (positive, solar panel(s) detected) or '0' (negative, solar panel(s) not detected). The distribution of training set labels is depicted as below:

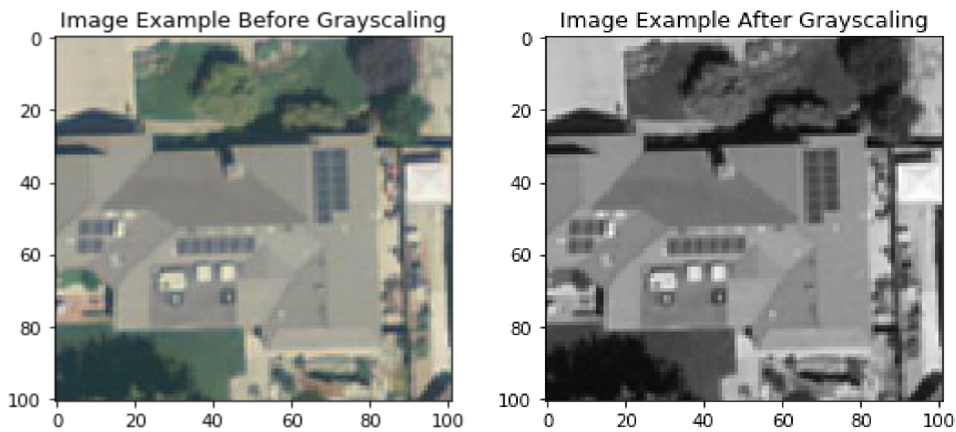


Each image is of size *101* × *101* = *10201* pixels and each pixel is composed of a 3-dimensional RGB value that ranges from 0 to 255 in each dimension. Two examples of the image set are shown as below:

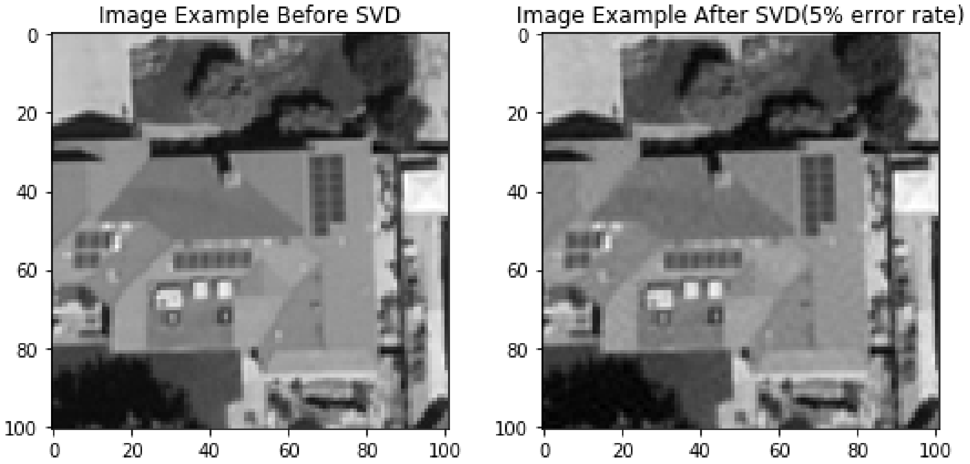


One challenge that is inherent to this problem is high dimensionality, which can lead to overfitting. Grayscaling and singular value decomposition are two proposed solutions that will be experimented with in a later stage.

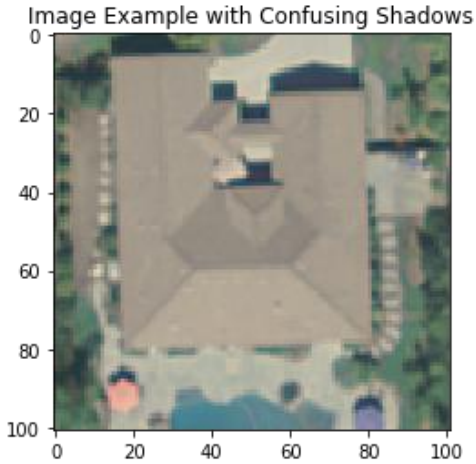
Grayscaling is a metric that transforms the 3-dimensional RGB value into a 1-dimensional range of monochromatic shades from black to white. Visually, grayscaling turns a colored image into an image with a limited number of shades of gray.



Singular value decomposition (SVD) is a method similar to principal component analysis (PCA) that uses a low-dimensional representation of a high-dimensional matrix (James, Witten, Hastie, & Tibshirani, 2013a). It allows an exact representation of any matrix, and also enables the elimination of less important parts of that representation to produce an approximation with any desired number of dimensions.



Another challenge is the shadow of the roof could potentially result in false positives since they can look similar to solar PVs, especially when the shadows are square. The solution to this problem is to increase the size of the training set or to build model sophisticated enough to make the prediction more accurate.

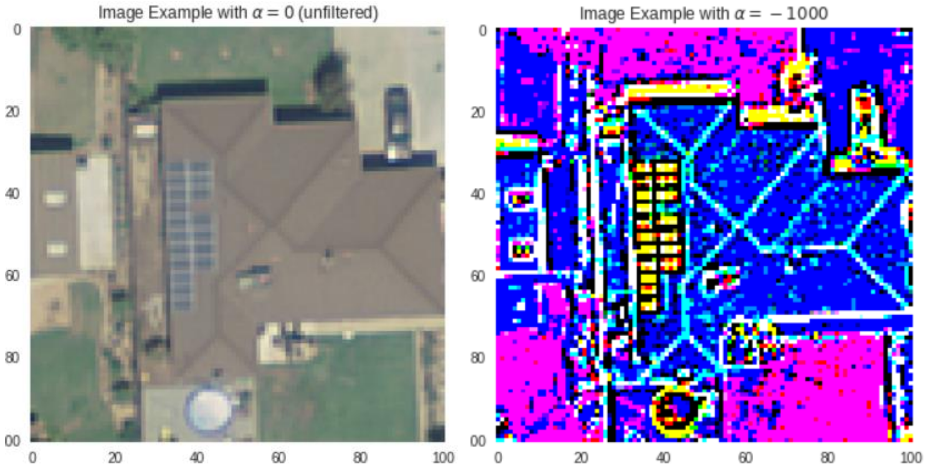


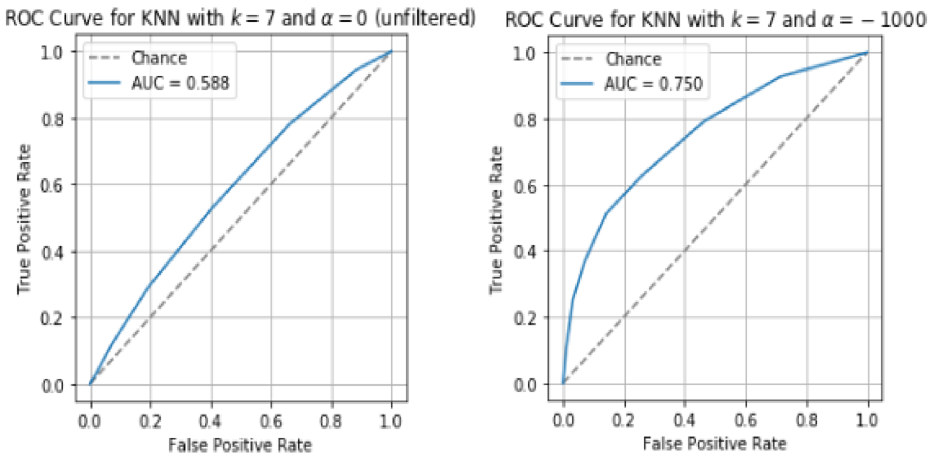
It is also a challenge to fit a model with, at most, 1,500 images as the training set. Many case studies use a much larger training set in order to get high accuracy on test set (e.g. the MNIST dataset contains 60,000 images in training set). One solution would be to expand the training set with aerial images from the Internet. However, it is extremely difficult to find aerial images with similar characteristics to this dataset and an arduous task to label the images manually. Another solution is to use an image data generator that generates additional training images from those available using transformations such as shift, rotation, flip, etc.

E. Methods

i) Preprocessing

The image data are loaded in via imread and converted to numpy arrays. When each image is loaded, a Gaussian filter is applied, originally with the intent to remove noise from the image. However, we found that subtracting the filtered image from the original image and multiplying by negative scalar led to a significant improvement in even very simple models. Though unscientific, this method inadvertently "highlighted" solar panels within the image.





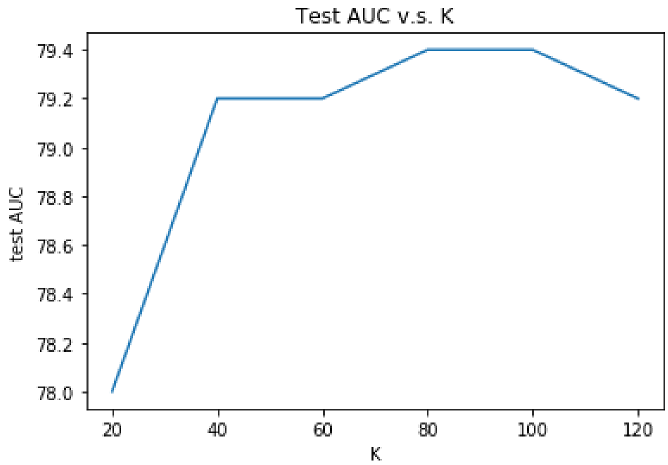
As mentioned in the previous section, grayscaling and SVD are two useful tools for dimension reduction in the hope of preventing overfit and reducing computational complexity. However, they might lead to underfitting due to the loss of information. The efficacy of these methods is assessed by comparing the performance of a model fit using them to one without such dimension reduction. For models including KNN, Random Forest, XG-Boosting and CNN, the accuracy on the test set is lower when dimension reduction is conducted, which is likely because the color of the image is an important distinguishing feature of a solar panel. Therefore, the models demonstrated below were fit on data without dimension reduction.

Standardization is applied to pixel features to eliminate the impact of the scales of different predictor values.

ii) Baseline Model – K Nearest Neighbors (KNN)

A baseline model is a model that is relatively straightforward to implement and used for the evaluation of the performance of more sophisticated models later on by comparison. The baseline model chosen for this project is K Nearest Neighbors (KNN), since it is extremely easy to implement in its most basic form, and yet can perform quite complex classification tasks (James, Witten, Hastie, & Tibshirani, 2013b). KNN assigns the classification to an item based on its K-nearest data points using Euclidean distance between the image arrays.

Although KNN is a non-parametric model, the hyperparameter K has to be determined. The test AUC on K from 20 to 120 are experimented with an interval of 20 and the result is depicted as below.



The AUC on test set doesn't fluctuate much with the change of K, and the optimal K is between 80 and 100.

iii) Support Vector Machine (SVM)

A support vector machine is a discriminative classifier defined by a separating hyperplane. More specifically, given labeled training data set, the algorithm outputs an optimal hyperplane which categorizes new examples (James, Witten, Hastie, & Tibshirani, 2013c).

The time complexity for SVM is quadratic thus it is hard to apply it on a huge data set. The 'kernel' argument of SVM defines the type of kernel to be used for segmentation. 'Linear' outperformed 'sigmoid', 'poly' and the default 'rbf' during experimentation. The 'gamma' argument defines the kernel coefficient for 'sigmoid', 'poly' and 'rbf' kernels. Since a linear kernel was used, 'gamma' remained at default 'scale'.

iv) Random Forest

A random forest is an integration of single classification trees. It adds additional randomness to the model while growing the trees and uses averaging to improve the predictive accuracy and control over-fitting.(James, Witten, Hastie, & Tibshirani, 2013d)

The 'n_estimators' argument defines the number of trees in the forest and the default value of 100 was used. The 'max_depth' argument defines the termination condition of the fit – the maximum depth of the tree. It was set to be 2 to stop the nodes from expanding until all leaves are pure, which, in other words, is to prevent overfitting.

v) XG – Boosting

Boosting classifiers combine thousands of simple trees with low accuracy into a high-accuracy model. Xg-boosting, being the abbreviation of eXtreme Gradient Boosting, excels in flexibility and speed among boosting methods.

As a derivative of tree model, xg-boosting shares a bunch of arguments with Random forest. The 'n_estimators' and 'max_depth' arguments are set under same consideration and value, the 'learning_rate' is the optimizing speed towards the opposite direction of gradient descent and was set to be 0.1.

vi) Resnet50, VGG19 and InceptionV3

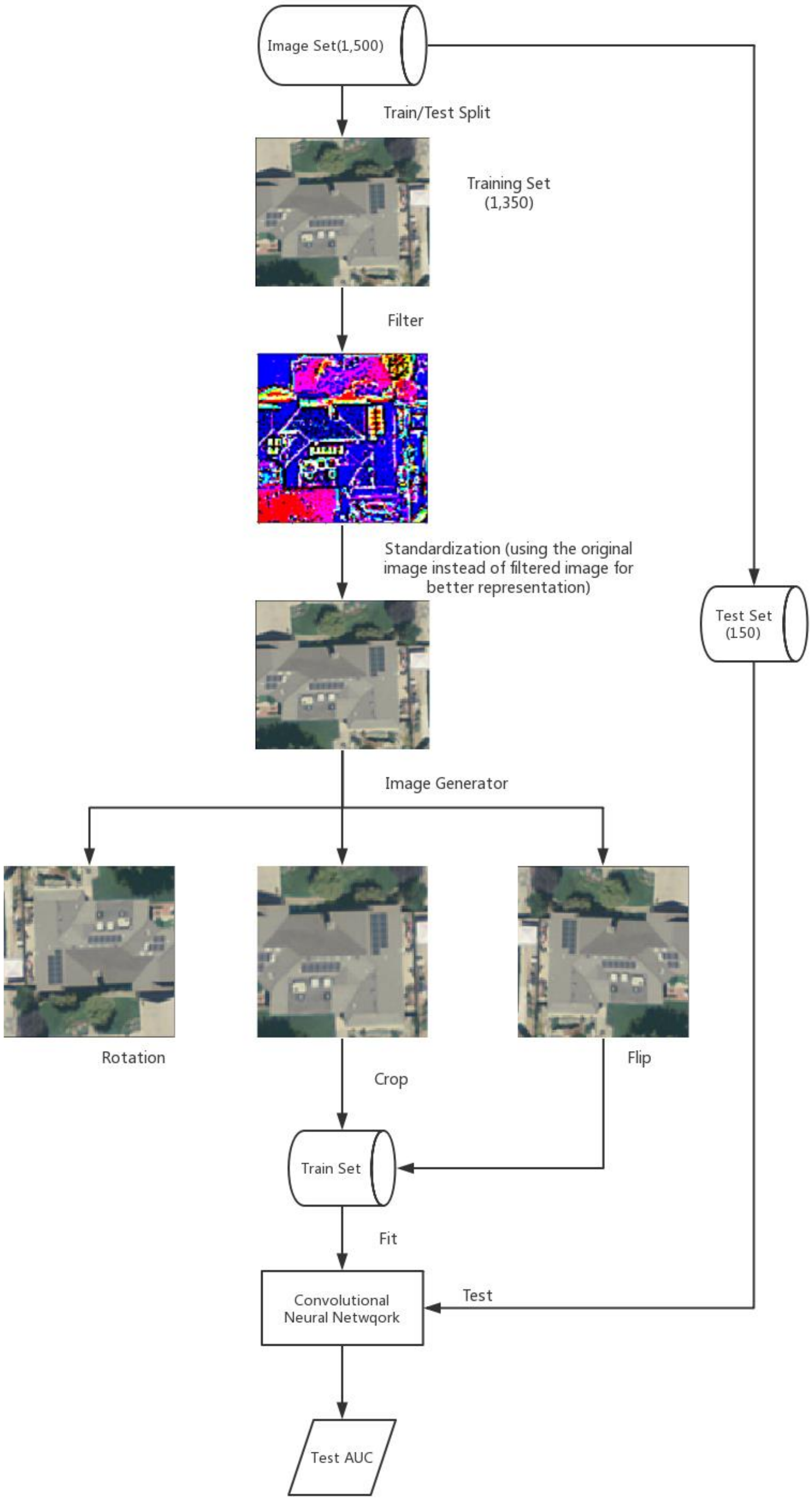
Pretrained models like Resnet50, VGG19 and InceptionV3 were experimented with but the improvement over a custom-built model was not great enough to warrant the additional training time required.

vii) Convolutional Neural Network (CNN)

Convolutional Neural Networks, like ordinary neural networks, are made up of neurons that have learnable weights and biases. Our model consists of five convolutional layers followed by three densely connected neural network layers. Following each convolutional layer, a maximum pooling layer is used to combine the layer's output into a single neuron as well as a batch normalization layer to maintain a mean activation close to 0 and an activation standard deviation of 1. The output of the convolutional layers is then flattened prior to the fully connected layers. Two of these subsequent dense layers use a rectifier activation function while the final dense layer uses a softmax activation. Additionally, prior to each fully connected layer a dropout layer is used to prevent overfitting.

Finally, the model was trained on batches of size 8 with Adadelta – an adaptive learning rate method – as the optimization algorithm. The model was trained for 20 epochs which was enough for the training accuracy to converge to 100% across most of the batches. The epoch number was chosen empirically and reached a balance between the fit of the model and efficiency.

The flowchart of the methodology is shown as below:



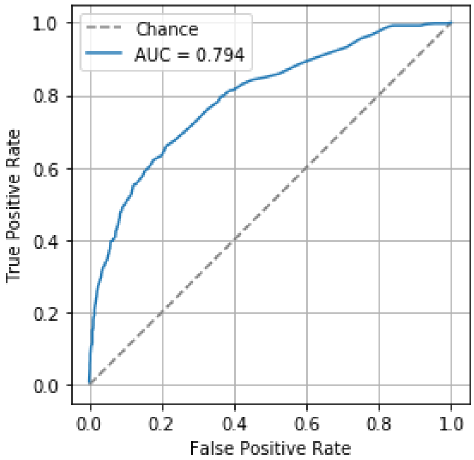
F. Results

For the KNN, SVM, Random Forest, and XG Boosting models, Stratified K-Folds cross-validation was used to assess their performance. For the CNN model, the training set is split into a training set and a validation set on the portion of 9:1.

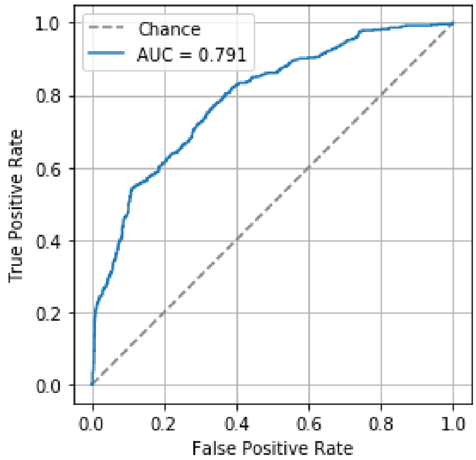
To make up for the small size of the training set, when training the CNN model, ImageDataGenerator from the keras package was used to generate more images from the original set using methods like rotation, flip, shift, etc. This turned out to be a useful tool and helped with the problem of underfit.

The ROC curve of implemented models are shown as below.

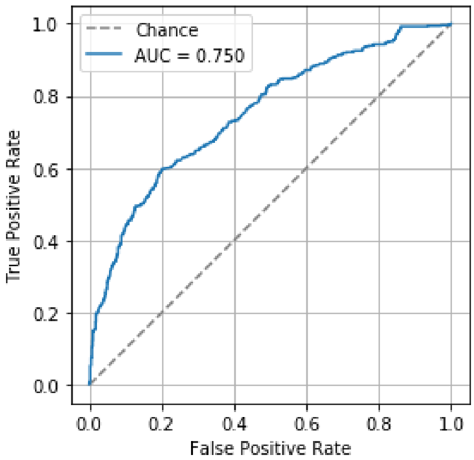
KNN (baseline)



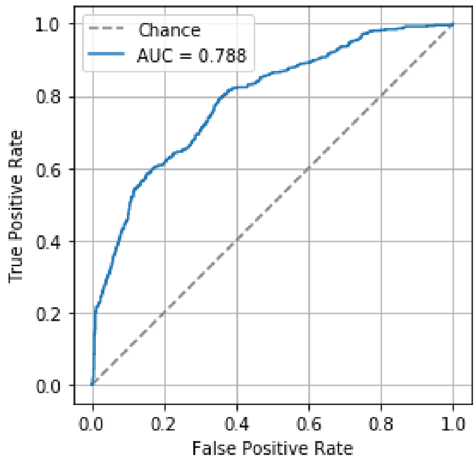
SVM



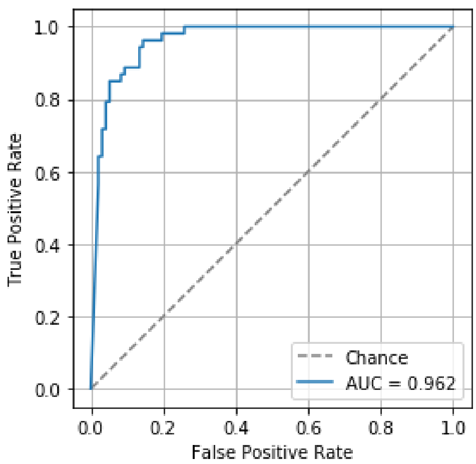
Random Forest



XG – Boosting



CNN



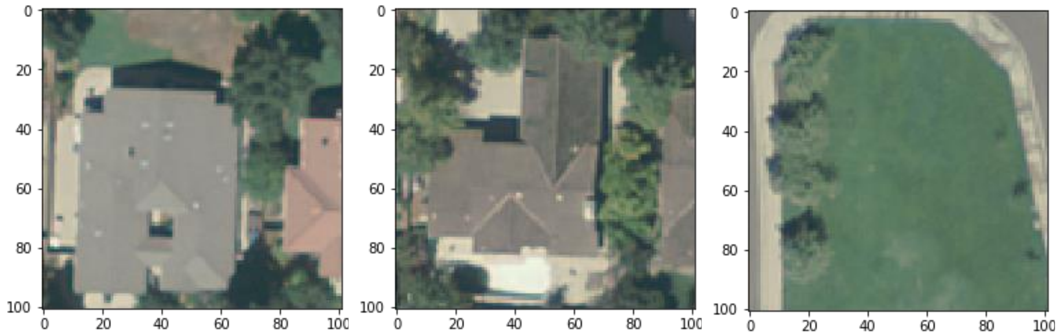
The Area Under Curve (AUC) on the validation set is a measure of the model performance based on true positive rate and false positive rate. Although KNN is a simple baseline model, the performance is a little better than those of SVM, Random Forest and Xg-boosting, with each has an AUC between 0.75-0.80. The CNN model outperformed the others with a test AUC of 0.962.

The confusion matrix for the CNN model is

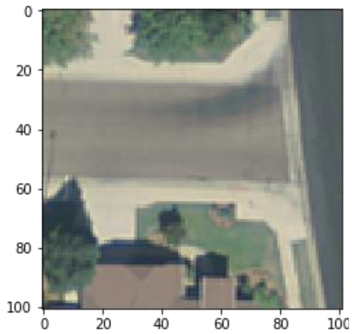
Prediction Class / **Real Class**	**Positive**	**Negative**
Positive	52	1
Negative	4	93

Some examples of the classification result are shown as below:

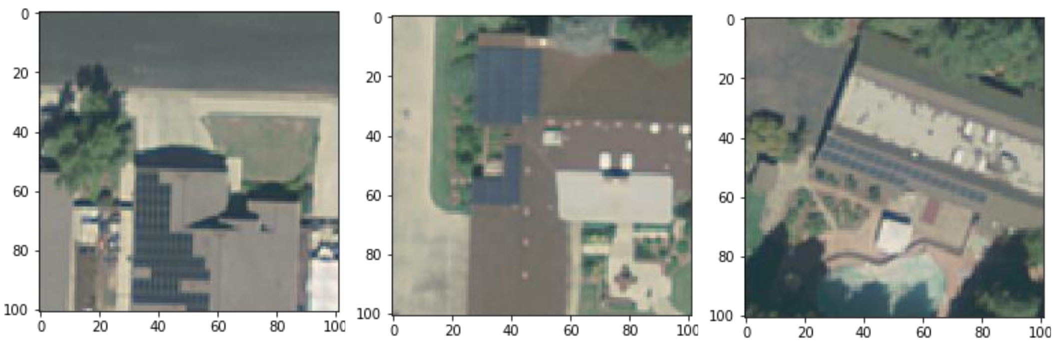
True Negative (Label 0, Prediction 0)



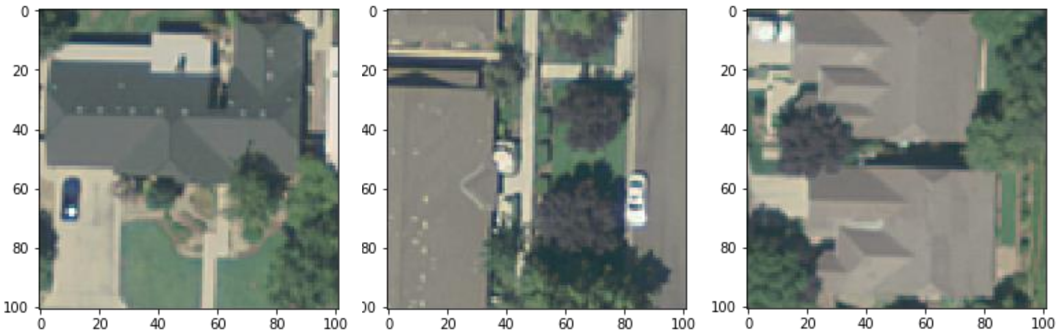
False Negative (Label 1, Prediction 0)



True Positive (Label 1, Prediction 1)



False Positive (Label 0, Prediction 1)



The misclassified images validated the idea in the ‘data’ sector that the shadows of the roofs would be a confusing item in solar panel detection. The CNN classifier performed best on images with a light roof color and sufficient light where the solar panels make a great contrast with its surroundings. This also validated the experiment result that the performance of the classifier would be better if the images are ungrayscaled since color and light play an important part in the detection process. These errors could potentially be fixed or eased by adjusting the RGB color values to make the contrast sharper. More importantly, the prediction accuracy would be more promising with a larger training set.

G. Conclusions

In this work, we employed a Convolutional Neural Network (CNN) to detect the presence of solar arrays on the roof of residential houses. We tried different transformations over the images or feature. We 'grayscaled' the images and also applied a dimensionality reduction technique (SVD). Neither was more effective than using the default images, and in fact we obtained a lower accuracy on our validation set. We also probed several models as XgBoosting, Random forest, KNN, SVM, and CNN. Among these techniques, CNN was the one with the highest accuracy, and therefore our final model.

Given the reduced training dataset (we had only 1,350 images to train the CNN) we decide to increase this number using synthetic data generated using the Keras library. By doing this we were able to reached an average accuracy of 96.3% and AUC of 0.962 for our Validation test and an accuracy > 99.1% over the test set.

For this assignment, we used images with the same size (101x101 pixels). All these images were preselected in order to show only one house. In future research, we would like to expand our classification algorithm for not edited images. In order to do so, we aim to develop a method to isolate individual houses from satellite images as in (Ghaffarian & Ghaffarian, 2014) in order to apply this method to real satellite data. Another possibility could be developing a detection algorithm that identifies the presence and position of solar arrays on aerial imagery as in Malof, Bradbury, Collins, and Newell (2016). By performing this we would be able to quantify the number of solar arrays on different geographical regions. This method would permit taking accountability of the regions with fewer solar arrays in order to implement public policy to promote more installations. By using public aerial imagery, every state could perform this analysis in a rapid and cheap way instead of relying on surveys that can be expensive and drawn-out.

Further, the method of repeatedly subtracting a Gaussian filtered version of the image from itself should be approached in a more rigorous manor.

Also, developing a method to detect solar arrays would be the first step in order to perform more complex analyses as it could be estimating the energy generation capacity of certain geographical regions (Malof, Li, Huang, Bradbury & Stretslov, 2019) or determine solar panels that need or would need maintenance in the future.

H. Roles

Scarlett Hwang: Preprocessing the data; building SVM model; final report.
Bingying Liu: Literature review; Building KNN and XG-Boosting model; final report.
Joaquin Menendez: Literature review; building Random Forest model; final report.
Nathan Scheperle: Preprocessing the data; image generator; building CNN model; final report.
Muxin Diao: Experiment with SVD; fitting pre-trained model; building CNN model; final report.

I. References

Allibhai, E (2018, October 16). Building a Convolutional Neural Network (CNN) in Keras. Towards Data Science, retrieved from https://towardsdatascience.com/building-a-convolutional-neural-network-cnn-in-keras-329fbbadc5f5

Camilo J, Wang R, Collins LM, Bradbury K, Malof JM. Application of a semantic segmentation convolutional neural network for accurate automatic detection and mapping of solar photovoltaic arrays in aerial imagery. IEEE Appl. Imag. Pattern Recognit. Work., 2017.

Ghaffarian, S.,& Ghaffarian, S. (2014). Automatic building detection based on Purposive FastICA (PFICA) algorithm using monocular high resolution Google Earth images. ISPRS Journal of Photogrammetry and Remote Sensing, 97, 152-159.

High-Resolution Aerial Maps Come to Solar (2018,July 26) Greentech Media, retrieved from https://www.greentechmedia.com/articles/read/high-resolution-aerial-maps-come-to-solar#gs.sh7nqB5L

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013a). 10 Unsupervised Learning In *An introduction to statistical learning* (pp. 374-385). New York: Springer.

James, G., Witten,D., Hastie, T., & Tibshirani, R. (2013b). 2.2.3 The Classification Setting In *An introduction to statistical learning* (pp. 37-42). New York: Springer.
James, G., Witten,D., Hastie, T., & Tibshirani, R. (2013c). 9 Support Vector Machines In *An introduction to statistical learning* (pp. 337-356). New York: Springer.
James, G., Witten,D., Hastie, T., & Tibshirani, R. (2013d). 8 Tree-Based Methods In *An introduction to statistical learning* (pp. 303-321). New York: Springer.
K. Bradbury, L.M. Collins, J.M. Malof, and R.G. Newell (2016). Automatic Detection of Solar Photovoltaic Arrays in High Resolution Aerial Imagery. CoRR, abs/1607.06029.
Kaila, G. (2018, June 6). How to easily do Object Detection on Drone Imagery using Deep learning. Medium. Retrieved from https://medium.com/nanonets/how-we-flew-a-drone-to-monitor-construction-projects-in-africa-using-deep-learning-b792f5c9c471
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.
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.