Project Report - Solar Photovoltaic in Aerial Imagery

Team 6 - Light Blue
Altamash Rafiq, Christy (Die) Hu, Ronald Nhondova
Srishti Saha, Tien Yu (Aaron) Liu

Abstract

In this study, we consider the task of comparing two classes of algorithms that automatically detect solar panels in aerial imagery. One of our approaches combines Gradient Boosting Machines (implemented via the Light GBM algorithm) with Histogram of Oriented Gradient, which lies in the area of classical machine learning algorithms. For the other approach, we use the current state-of-the-art model, EfficientNet with Adversarial Propagation (AdvProp) training scheme, in the class of convolutional neural networks (CNN). Our result shows that the CNN approach substantially outperforms the traditional machine learning approach in terms of feature extraction and model learnability, yielding superior generalization performance. We also discuss how our modeling approach presents a powerful solution to the challenge of estimating solar power consumption in the United States, thereby having highly scalable applications for both public and private sector industries.

Introduction

In this report, we consider the problem of using advanced machine learning algorithms, in particular gradient boosting machines and convolutional neural networks (CNN), for the identification of solar panels in aerial images. Our objective is to classify images of buildings and streets that are fed to our algorithms as either containing or not containing solar panels and to, thereby, compare the efficiency of these models for the task of image classification. We consider aerial images restricted to a size of 101x101 pixels with 3 color channels, containing either one or more solar panels each of varying sizes and shapes. Such an approach to identifying solar panels in aerial images can have important applications in a variety of sectors, preeminent among which is its potential to help estimate the scale prevalence of solar power consumption in the US.

The key benefit of our approach of using machine learning with regards to estimating solar power prevalence is that the process of labeling images via our algorithms is entirely automated making our approach highly scalable. Without the use of image data, the goal of estimating solar power usage in the US would be incredibly challenging as it would be immensely costly and time consuming. Even with image data, without the use of an algorithm such as ours, the task of identifying solar panels and quantifying their scale would be immensely arduous. As such, machine learning models present a powerful solution to the challenge of identifying solar panels and it is similarly important to determine what algorithms are most successful in this classification exercise. In this report, we compare the performance of the two aforementioned machine learning models for this problem and find that convolutional neural networks clearly outperform the gradient boosting machine, which is perhaps the most powerful of the non-neural network based models. In particular, we make use of the state of the art EfficientNet-B4 with adversarial propagation to achieve near perfect out-of-sample generalization performance on our aerial images.

Background

The objective of identifying solar panels in aerial image data using machine learning is one whose importance has been recognized for several years. There are several reasons that have motivated researchers to attempt projects with this goal. Firstly, such an approach can help approximate the extent of solar power usage in the United States. Currently, there is no directory that records the extent of solar power usage in the US, which means that a machine learning approach like ours can be critical in estimating the scale of solar power accessibility in the US and the extent to which it is contributing to its energy supply and consumption. This can be of immense importance to policy makers as they evaluate the degree to which solar power should be invested in or lobbied for. Secondly, groups such as real estate companies, that are preoccupied with accurately valuing properties and assets, would find such projects immensely useful. This is because the prevalence of solar panels in a given community or the presence of them on a certain property can lead to more accurate quantification of property values. As such, being able to estimate solar power prevalence can be immensely important for many groups/industries.

In 2016, for example, J. M. Malof et al. (2016)[8] came out with a paper on using Random Forests, a powerful non neural-network based machine learning method, for aerial image classification for solar panel identification. While these researchers found some success using this method, more recent attempts by a number of researchers, see Malof et al. (2017) [7] and Yu et al. (2018) [6] for example, have had greater success using the machine learning algorithm of Convolutional Neural Networks for solar panel detection. These approaches have yielded performance at a much higher level (when measured through metrics like accuracy, precision and recall) than older, non-neural network based methods. Yu et al. [6], for example, achieve a highly impressive accuracy of 93.3% based on their implementation over hundreds of thousands of images.

This change is the kind of machine learning algorithms being used for solar panel detection is motivated by a more general shift in the image classification and pattern recognition field from classical machine learning approaches to more modern neural network based approaches, particularly CNNs. Image data, such as that which we work with, has historically been notoriously hard to classify well. The two primary reasons for this are as follows :-

1. Image data is often high dimensional, leading to the need of a very large number of images to train a machine learning model to classify it well.
2. Objects within each image almost always have a spatial relationship to one another, meaning that the vast majority of machine learning algorithms, in which spatial relativity/interaction of dimensions has limited impact on the final classification, are likely to perform poorly when used for pattern recognition in images.

These challenges mean that predictive models, such as the gradient boosting machine that we use as our baseline model for prediction have limited success with the kind of image data we are working with. However, the increasing prevalence of deep learning and CNNs since 2012 has provided an immensely promising solution to the challenges of image classification. Deep learning has proved to be surprisingly good at managing high dimensional data, often providing incrementally increasing returns in terms of performance on larger amounts of data when non-neural network based machine learning algorithms provide diminishing returns with more data. Additionally, the ability of convolution layers in a CNN to capture the spatial relationships between objects in images is highly beneficial for image classification, leading to very impressive gains in predictive accuracy compared to more traditional machine learning methods.

On top of this, the ease of 'pre-training' CNNs on large image data repositories such as ImageNet make them suitable for problems such as ours where the sample size of images available is limited. This is known as transfer learning and is discussed further in the EfficientNet-B4 section of this report. Both Malof et al. [7] and Yu et al. [6] have used transfer learning in their approaches to identify solar panels and we take inspiration from their approaches in using transfer learning in our own CNN implementation.

In our analysis, we deal with less complex data than both the aforementioned groups who have used CNN in their version of aerial image classification. However, we try to build on their work by employing the state-of-the-art EfficientNet-B4 method. This method is discussed in greater detail in the CNN section of this report.

Data

Our data for this project is comprised of two samples, which we designate as the training and test sets. The training dataset has 1500 images with the distribution as shown in Figure 1. As the test dataset does not have available labels, the models we build either use 10-fold cross validation or create a validation set from the training dataset to assess out-of-sample performance. The training dataset comprises of 505 (33.7%)

images having solar panels and the remaining having no solar panels. The slight imbalance between the two classes is noted, and accounted for in our final fitted models and it thereby does not impact our prediciton performances.

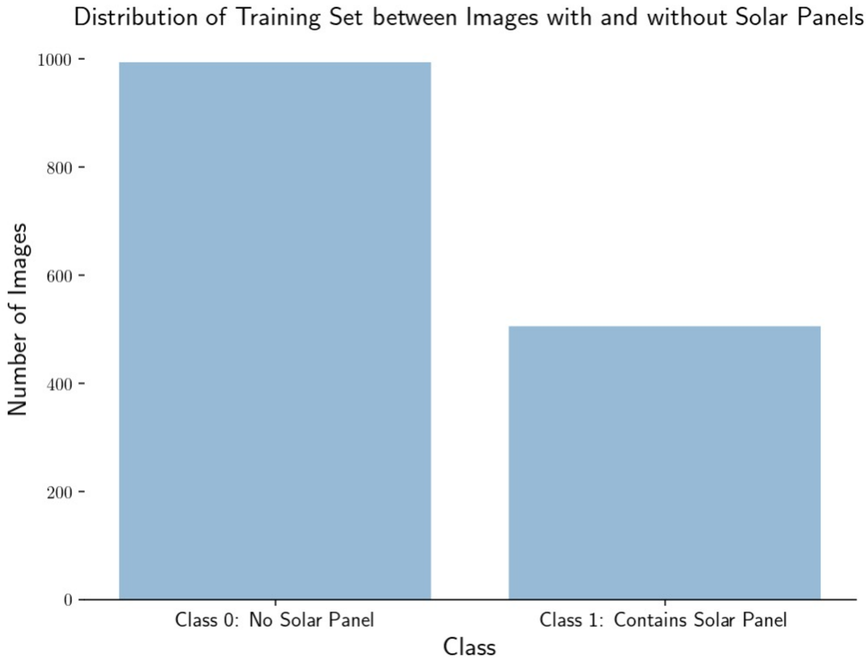
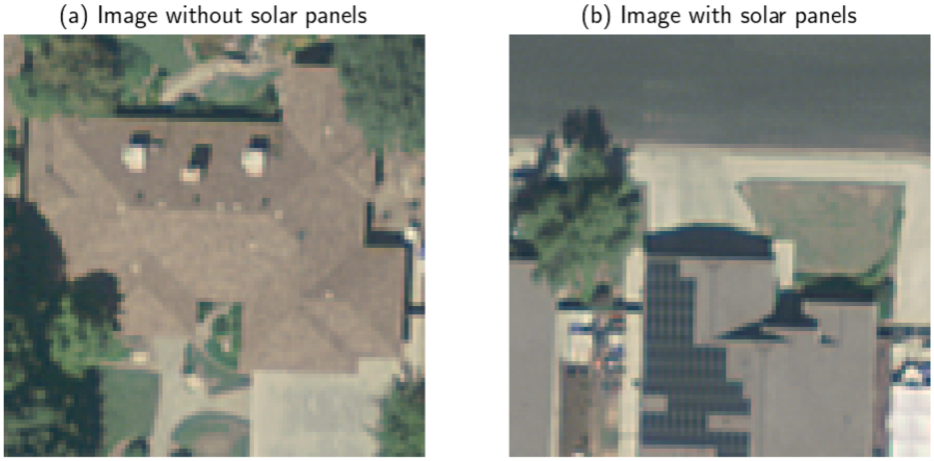


Figure 1: Barplot of unbalanced data distribution

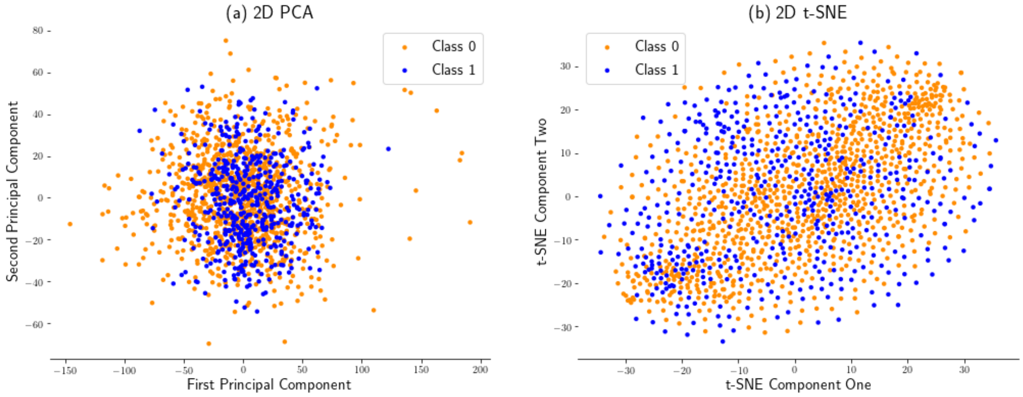
Our images were stored in the TIFF format, which has several advantages over other formats like JPEG. For example, the TIFF format will not lose any quality if edited and saved repeatedly, which can be beneficial especially if image augmentation or other image manipulation techniques are employed. As we make use of both these image transformation methods in our machine learning model implementations, the format of the images is of importance to our project. The dimensions of each of our images is 101 x 101, for each of the 3 channels (red, blue and green). Sample images for a roof top without and with solar panels are plotted in Figure 2 (a) and (b) respectively. From visual inspection of the two images, the images seem to have sufficient clarity for one to identify rooftops with solar panels, with a reasonable degree of confidence. A potential challenge however, could be distinguishing a solar panel from shadows of other objects on the roof, for example in Figure 2(a), the chimney like objects, are casting shadows which could cause issues for algorithms and eventual performance.


(a) Image without solar panels

(b) Image with solar panels

Figure 2: Example images with one from each class

When an image is flattened into a two-dimensional array, this converts it from a 101x101x3 to 1x30603, which creates a high dimensional feature problem to model. As an attempt to visualize the data set and the difficulty in separating the two classes of images, its first two principal components are calculated as shown in Figure 3(a). The illustration shows overlap between the two classes, with Class 1 (with solar panels) images more densely packed around (0,0) than Class 0 images. To confirm this observation, a t-Distributed Stochastic Neighbor Embedding (t-SNE), is performed on the data that is first reduced to 500 principal components, with the results shown in Figure 3(b). Based on the plot, there is strong evidence to suggest that the classes are not linearly separable.


(a) 2D PCA

(b) 2D t-SNE

Figure 3: Reduced dimension data is not linearly separable using 2D PCA and T-SNE

Method

We employ two different methods for this image classification problem. We first create a baseline model using feature extraction followed by LightGBM (Light Gradient Boosting Machines). To further improve the performance, we employ a neural network using EfficientNet-B4 with adversarial propogation. The sections below discuss in detail the methodologies followed for both these approaches.

Light Gradient Boosting Machine Classifier

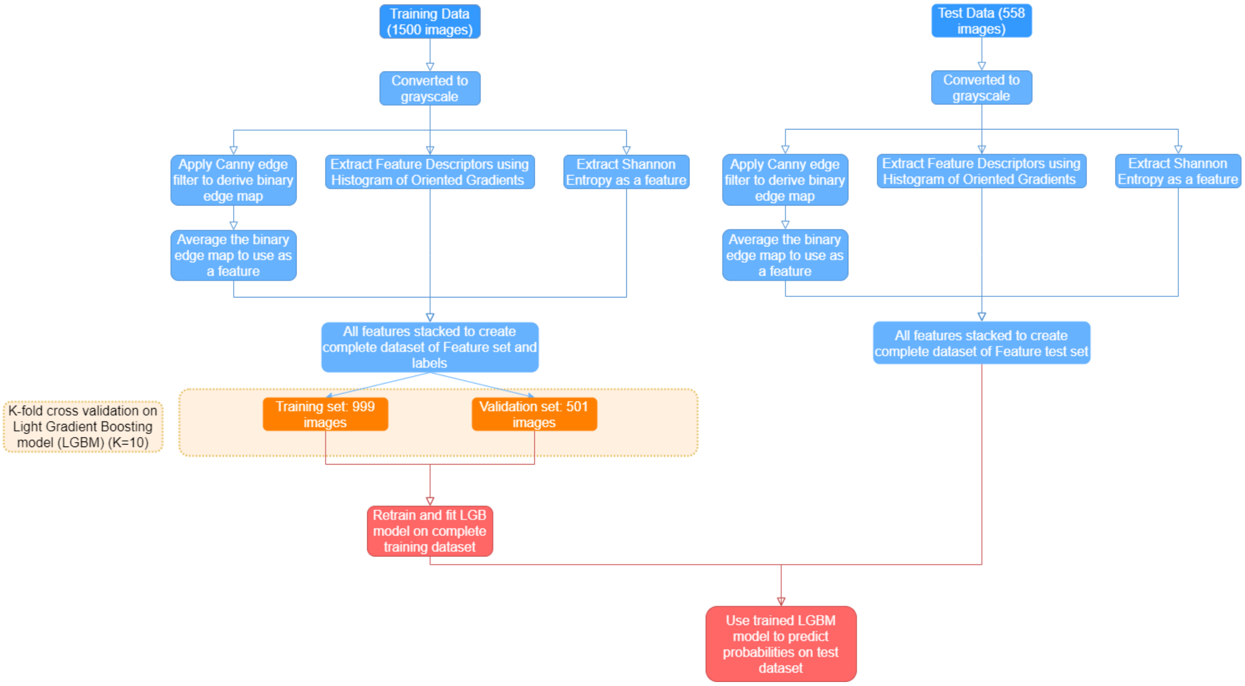

Training Data (1500 images)
Converted to grayscale
Apply Canny edge filter to derive binary edge map
Extract Feature Descriptors using Histogram of Oriented Gradients
Extract Shannon Entropy as a feature
Average the binary edge map to use as a feature
All features stacked to create complete dataset of Feature set and labels
K-fold cross validation on Light Gradient Boosting model (LGBM) (K=10)
Training set: 999 images
Validation set: 501 images
Retrain and fit LGB model on complete training dataset
Test Data (558 images)
Converted to grayscale
Apply Canny edge filter to derive binary edge map
Extract Feature Descriptors using Histogram of Oriented Gradients
Extract Shannon Entropy as a feature
Average the binary edge map to use as a feature
All features stacked to create complete dataset of Feature test set
Use trained LGBM model to predict probabilities on test dataset

Figure 4: Flow Chart of LightGBM Implementation

While deep learning and advanced techniques like CNNs have proven to be the most successful techniques for image classification, comparable results may be achievable in some situations with non-neural network based methods. The use of such alternative methods instead of CNNs is motivated by the following limitations of CNN :-

1. CNNs require a large amount of computational resources to train which may be unavailable.
2. CNNs often suffer from overfitting as they learn from and generate features tailored to the specific data they have been trained on. This means that if the features and the model are not well constructed, they are likely to not perform well for images that are significantly different from the training set. This means that the amount of data required to train a CNN well may be too immense and unfeasible.

Thus, a lot of projects make use of more classical Computer Vision algorithms to transform and extract features from images and use non-neural network based machine learning models to conduct image classification. Such algorithms are well-established, transparent and optimized for computation power and performance. They can also serve as baseline models to compare and improve the efficiency and accuracy of advanced deep learning models.

As our model for non-neural network based classification, we produced a gradient boosting machine with feature descriptors (discussed later in this section) for all the images along with additional features extracted from the image through filters. Our first step was to convert the images into grayscale, which was done by taking the weighted average of the RGB values. This process removes all color information (chrominance) and leaves a simpler characteristic (for example, luminance) of each pixel. In other words, this conversion fills every pixel with various levels of ‘gray’ (between levels 0 and 255), thereby reducing the complexity and dimensionality of the images (see Figure 5). This also improves the signal to noise ratio of the image as the unnecessary information provided by colors is removed from it. This dimensionality reduction has the added benefit of increasing the algorithm’s speed while reducing its complexity at the same time. Among the different ways of converting an image to its grayscale version, the use of luminance is popular and has demonstrated promising results. This is because Luminance matches the human perception of brightness[2]:

$$G_{Luminance} \leftarrow 0.3R + 0.59G + 0.11B \tag{1}$$

After generating new features with Luminance values instead of RGB pixel values, the Histogram of Oriented Gradient (HOG) descriptor was used to represent the images as feature descriptors. Feature descriptors are descriptions of an image patch around a certain point in the target image. They encode the most significant information in each patch using numerical vectors. HOG was selected as the feature descriptor in our case as it has been highly effective and is a popular algorithm for object detection as stated by N. Dalal et al. (2005)[15]. HOG divides the image into small connected cells and computes a histogram of gradient directions or edge orientations for the pixels within each cell. These cells are then grouped to form blocks and the aforementioned histograms are normalized for each block. This process provides more information than simple edge detection as it not only detects if the pixel is an edge or not, but also provides the edge direction by extracting both the gradient and the orientation of the edge. Thus, the feature vector from each image was extracted using the HOG algorithm which fed into the classifier as features. The HOG features were represented as an array of 1440 numerical values.

The grayscale image was also used to extract two other features: the Shannon entropy of the image and the averaged binary edge map of an image via a Canny edge detector. The Shannon entropy of an image describes the information content of an image (which can also be interpreted as a discrete signal) in the form of the log of the probability of pixels of a certain intensity ($-\log(p_k)$ *where* p_k *is the probability of pixels having an intensity* k). The higher the Shannon entropy, the higher the information content of an image. This feature was selected as it represents the probability of a certain signal value which might differentiate certain objects (for example solar panels, or trees) across all images[10]. The Shannon entropy was fed into the model as a feature. A Canny edge detector is applied to the grayscale images. The Canny edge detector[13] uses a derivative of the Gaussian filter to reduce the noise and compute the intensity of the edge gradients of each image. This then feeds into an algorithm to reduce the edge-width to 1-pixel curves by removing non-maximum pixels of the gradient magnitude. We choose the Canny edge detector because it has a low probability of failing to mark real edge points and is used extensively in object detection and image classification. The binary edge map of each image is extracted using the Canny edge detector and is averaged by the image size to create features for image classification. After adding the two features mentioned here, the total number of features per image is 1442. Thus the dataset is now represented as a 2D matrix of size: (1500x1442). Note that none of the original grayscale pixels are now used as their information has been extracted and captured in the 1442 generated features.

The above features are all stacked into one feature set to form the complete training dataset. This dataset is then fed into a K-fold cross-validation (K = 10) algorithm using LightGBM as the classifier. LightGBM is an ensemble model of decision trees which are trained sequentially on the residual errors of previous iterations[3]. LightGBM uses histogram-based split finding, which converts continuous features into discrete bins leading to more efficient tree splits, and it further reduces the algorithmic complexity by down-sampling data and features using Gradient Based One Side Sampling and Exclusive Feature Bundling. It can thus outperform other tree-based ensemble models like Extreme Gradient Boosting in terms of computational speed and memory requirements even for large datasets. Since, the feature space in this case is high-dimensional, an optimized algorithm like LightGBM is a sensible choice. Moreover, M. Ustuner et al (2019) used LightGBM for crop-detection from aerial images which proved to be an effective algorithm for this task[14]. We conduct a 10-fold cross-validation to assess the performance of the algorithm on a training and validation data split (999 samples in training-501 samples in validation). The hyperparameters of the model are chosen as follows:

1. Learning rate of 0.1
2. Number of boosted trees to fit (estimators): 1800
3. Boosting algorithm: ‘dart’ (Dropouts meet Multiple Additive Regression Trees)

These hyperparameters are chosen after manually iterating through different combinations of the parameter settings to give the best AUC (area under curve).

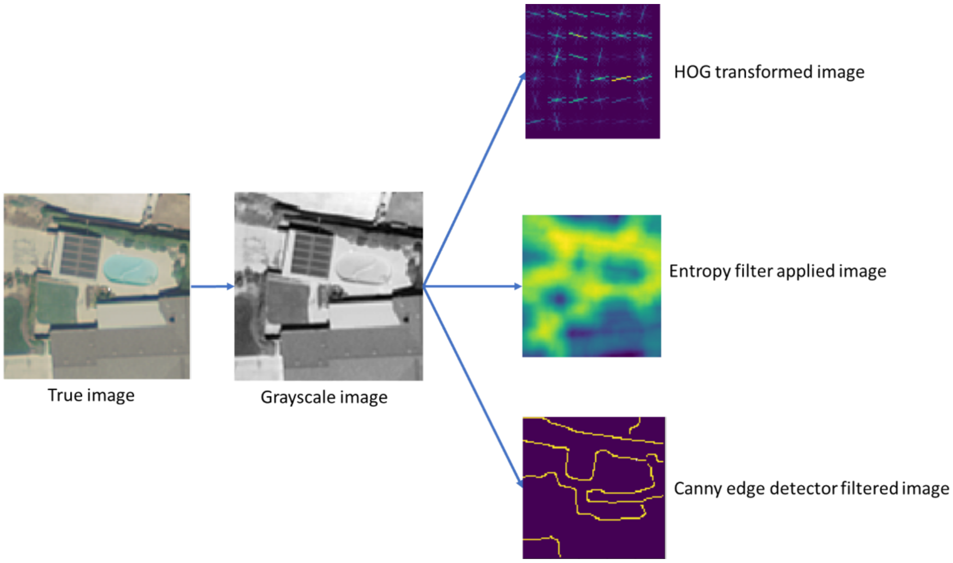

HOG transformed image
Entropy filter applied image
Canny edge detector filtered image
True image
Grayscale image

Figure 5: From the true image, grayscale transformed image was obtained by calculating weighted average as luminance; HOG transformed image was obtained after applying the HOG descriptor ('L2' normalization); Entropy filter returns the minimum number of bits needed to encode the local gray level distribution; and Canny edge detector with sigma=3 returns the edges detected in the image.

EfficientNet-B4 under AdvProp Training Scheme

Compared to non-neural network based machine learning approaches, it has been shown that CNNs have superior performance in processing image data as they can effectively capture the spatial relationships among image pixels and learn both low-level and high-level features automatically through optimization. Among several architectures of this class of neural networks, EfficientNet, developed by Tan et al. (2019)[11], achieved state-of-the-art performance on ImageNet, achieving a top-1 accuracy of 84.4% (EfficientNet-B7), as well as on several transfer learning benchmark datasets (CIFAR-100, Stanford Cars and Flowers) with reduced parameters compared to existing CNNs. These breakthroughs are mainly attributed to a comprehensive model scaling method: compound scaling. Unlike conventional scaling practices that only focus on a single aspect among the width, depth and resolution in a neural network, this compound scaling approach balances those dimensions with a set of fixed scaling coefficients and further enables EfficientNet to learn more efficiently and accurately compared to other network architectures. Built upon EfficientNet, Xie et al. (2019) created a new training scheme, Adversarial Propagation (AdvProp), which seperates adversarial samples and clean samples into two different batch normalization layers[1][18]. This allows the network to fully utilize the information from both of the samples and achieves the current state-of-the-art performance on ImageNet with a top-1 accuracy of 85.5%. As a result, in this project, we use EfficientNet-B4 (containing 19 million tunable parameters) under the AdvProp training scheme with pre-trained weights on ImageNet as our model and fine-tune it via training on our aerial images.

As we are using a pre-trained model, standardization is performed to the whole aerial images data set. Using these samples, the model is trained on 1,425 images and validated using the other 75 images. However, considering that the size of the training set is relatively small compared to the number of parameters, and aerial images are normally taken vertically from an aircraft, it is reasonable to assume that they may possess different orientations in the real world scenario. As a result, online data augmentation is applied to the training set, including random rotation with 90, 180 and 270 degrees. Furthermore, since color jittering may also happen and impact the resolution of aerial images, we use random color jittering to incorporate slight changes to the color values of the images. In addition, to help inspect the generalization performance of our model, the random rotations are also applied to the validation set, yielding 300 images for validation.

To train our model, since Keskar et al. (2017) found stochastic gradient descent (SGD) yields better generalization performance compared to other adaptive optimization methods, such as Adam, RMSprop or Adagrad [17], we use SGD as our optimizer, with Nesterov momentum 0.9, weight decay 1e-5 and initial learning rate 1e-4. In terms of objective function, we employ cross entropy loss, which is commonly-adopted in binary classification tasks.

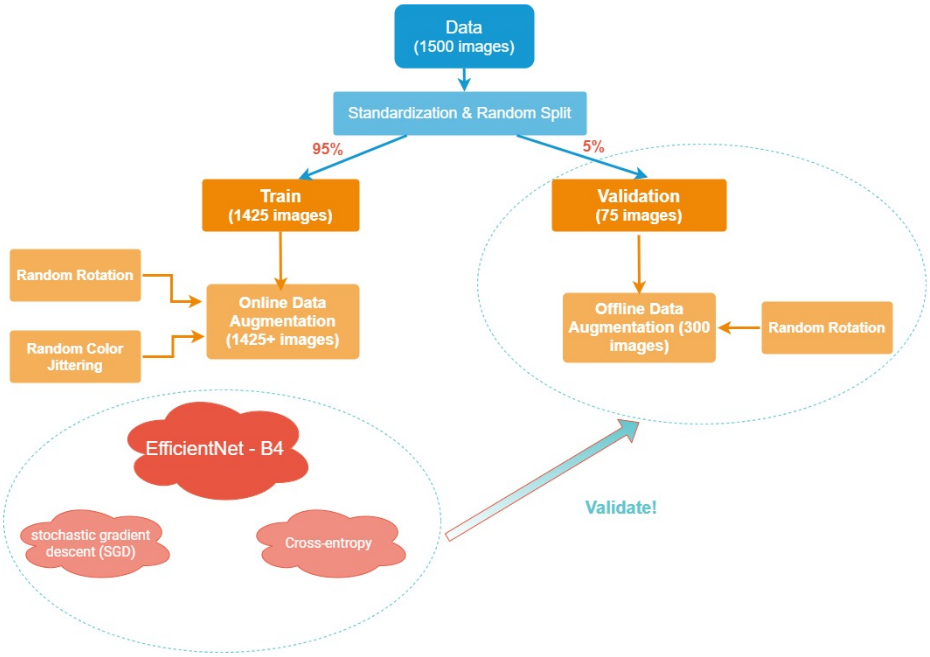


Figure 6: Flow Chart of Convolutional Neural Network Implementation

Results

After training our models, we evaluated their performance on the validation sets and compared them to assess where one model performed better than the other and where the models struggled. In Figure 7, we present the validation data Receiver Operating Curve (ROC) for our models. As can be seen, we achieved an Area Under the Curve (AUC) of 1.0 on the test data with the EfficientNet-B4 and 0.8189 using the GBM, both performances that are significantly better than random chance.

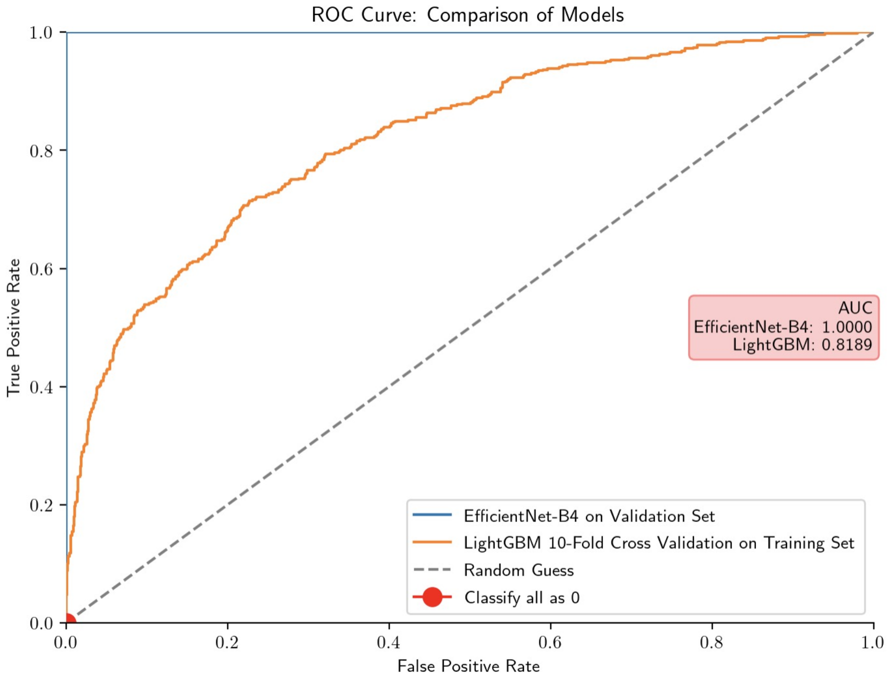


Figure 7: ROC curve for LightGBM and EfficientNet-B4 model

In addition, we plot the precision recall curve, on the test data, of our classifiers in Figure 8, which suggests that while both models are strong, the EfficientNet-B4 is significantly better. To make the class predictions, the default probability threshold is set at 0.5 i.e. probabilities above 0.5 are translated to class 1 and the ones below 0.5 are classified as class 0 (no solar panel).

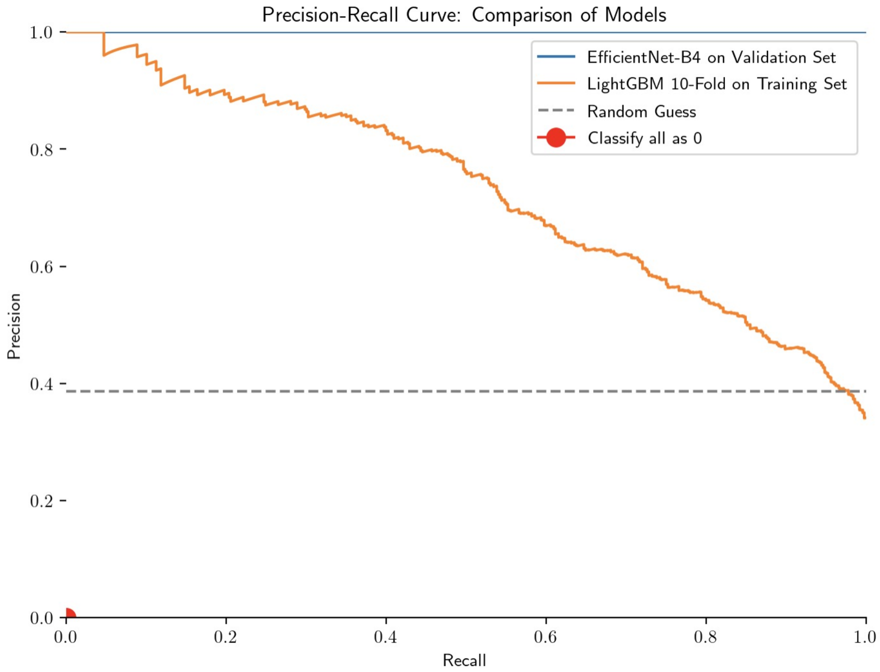


Figure 8: Precision Recall curves for LightGBM and EfficientNet-B4

Clearly, the LightGBM model has a higher proportion of false positives and false negatives as compared to the EfficientNet-B4 model, which has no misclassifications on the test data. See Table 1 and Table 2 for a comparison of the confusion matrices on the test data for each model. These clearly show the generalization superiority of the EfficientNet-B4 compared to the LightGBM, the former of which generalized perfectly on the validation set. The confusion matrix for the EfficientNet-B4 was calculated not only using the original validation data of 75 images but also with various rotations applied to these images as well as color jitters, which meant that a total of 300 different images were used for validation. Despite these transformations the EfficientNet-B4 correctly classified all the validation samples showing how it is rotation invariant. Note that the performance of the LightGBM is evaluated using 10-fold cross validation while that of the EfficientNet-B4 is only evaluated on a single validation set due to the large amount of time needed to fit each EfficientNet-B4 model.

Table 1: LGBM Cross Validation Confusion Matrix

	Predicted	
	Negative	Positive
Negative	59.33%	7%
Positive	15.47%	18.2%

To illustrate the comparison of performance of both the models, we have taken one example each from false positives, false negatives, true positives and true negatives, as classified by each model on the train and test data combined, and compared them (see Figure 9 and Table 3). Note that the EfficientNet-B4 model does not have any false positive predictions in either the train or the test data so the image in the 'false positive' category is only discussed in light of the LightGBM model.

The LightGBM model incorrectly predicts the example as positive with a very high probability (0.984). The predicted probability for the same from EfficientNet-B4 is 0.00004 which is a very confident prediction of a

Table 2: EfficientNet-B4 Validation Set Confusion Matrix

	Predicted	
	Negative	Positive
Negative	61.33%	0%
Positive	0%	38.67%

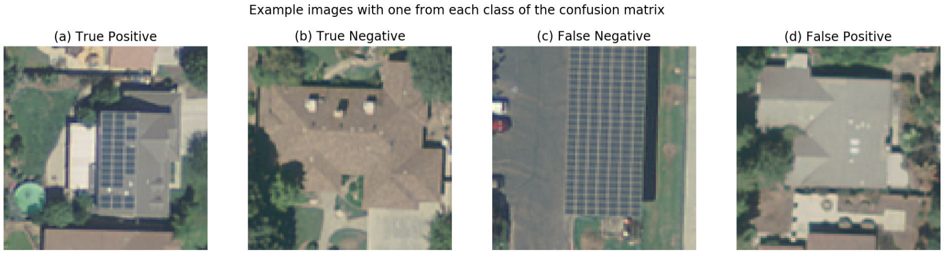


Figure 9: Four examples of each class of prediction

Table 3: Comparing Predicted Probabilities from Both Models - Full Training Data

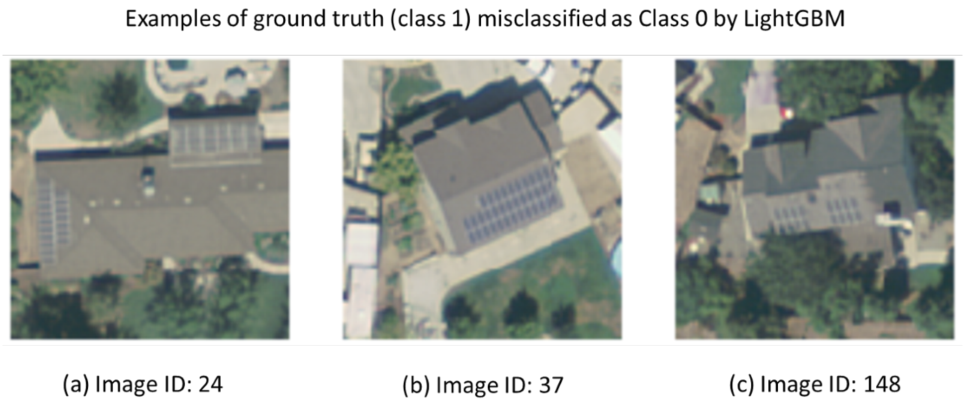
	Image ID	LGBM	EfficientNet-B4
True Positive	1499	0.928775	0.999559
True Negative	1	0.000928	4e-06
False Negative	1399	0.004401	0.356189
False Positive	47	0.984	-

class 0 (no solar panel). This shows how drastic of difference these two models have in terms of their mapping of the data. As for the example for False Negatives, EfficientNet-B4 has only 1 misclassified case. Image ID 1399 is misclassified as class 0 (no solar panel) with a probability score of 0.356 by the EfficientNet-B4 model. However, LightGBM shows a stronger misclassification as it assigns a lower probability score of 0.004. This shows that, on this specific image, as well as many others like it, LightGBM has a tendency to misclassify more strongly than the EfficientNet-B4 i.e. it is more wrong on average in misclassifications than the EfficientNet-B4.

When we look at the true positives and true negatives, EfficientNet-B4 outperforms LightGBM by a huge margin. If we look at the True Positive depicted by Figure 10(a), the predicted probability score from EfficientNet-B4 is 0.99 and that from LightGBM is 0.92. While the probability score from EfficientNet-B4 is higher than that from LightGBM, both the models are very confident about the example being classified into class 1 (presence of solar panel). Similarly, the example of True Negative shows that both models have a significantly low probability score assigned to the image. However, the EfficientNet-B4 model assigns a more confident class 0 to this example as it has a probability score of 0.000005 while the LightGBM model predicts class 0 with a probability of 0.00093. This suggests, again, that the EfficientNet-B4 has a much better understanding of the underlying structure of the data than the gradient boosting machine.

When we look at the LightGBM model, it shows a recall of 0.72 based on a 0.5 threshold. This implies that 72% of the actual positives were correctly identified as positive cases (True Positive Rate). It also has a precision score of 0.76 which indicates that of all the predicted positive classes, 76% of the images actually belonged to class 1.

Some of the examples belonging to class 1 which were misclassified as class 0 by LightGBM are shown in Figure 10. These misclassifications possibly arise due to the gradient, color shade and size of the solar panels


Examples of ground truth (class 1) misclassified as Class 0 by LightGBM

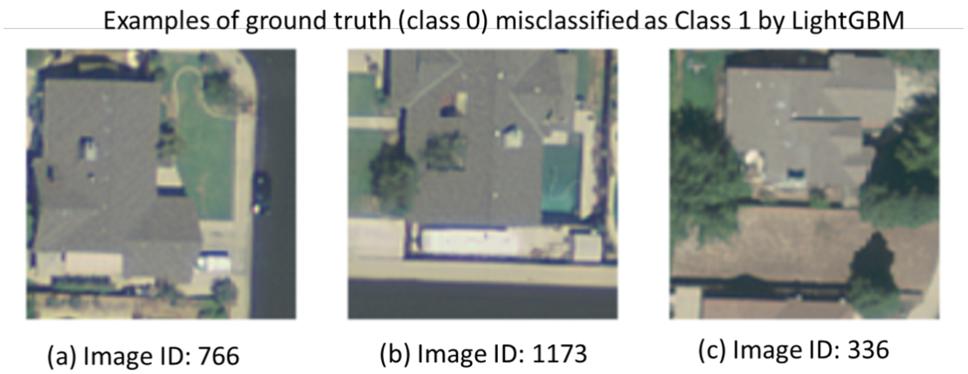
(a) Image ID: 24

(b) Image ID: 37

(c) Image ID: 148

Figure 10: Three examples from Class 1 misclassified as Class 0 by LightGBM

in these images. These images however were correctly identified as class 1 (presence of solar panel) by the EfficientNet-B4 model. This is because the EfficientNet-B4 model is able to capture these features more accurately.


Examples of ground truth (class 0) misclassified as Class 1 by LightGBM

(a) Image ID: 766

(b) Image ID: 1173

(c) Image ID: 336

Figure 11: Three examples from Class 0 misclassified as Class 1 by LightGBM

The above image (Figure 11) shows 3 examples of images that had no solar panel (ground truth: class 0) but were misclassified into class 1 by the LightGBM model. We hypothesize that objects like the swimming pool, car/road and shadows were erroneously misidentified as solar panels by the LightGBM, while the EfficientNet-B4 was able to differentiate them from the solar panels leading to more accurate classifications.

Conclusion

In this paper, we evaluate through comprehensive model comparisons the efficacy of our gradient boosting machine and convolutional neural network model on solar panel identification in aerial image data. We find that EfficientNet-B4 consistently performs better than LightGBM both in terms of AUC and confidence scores for the class assignments. We note that EfficientNet-B4 has superior performance in comparison to LightGBM as a result of its ability to capture the spatial relationships in the images, which LightGBM struggles with. To improve the performance of LightGBM, we implement the HOG feature descriptor, which yields notable improvements due to dimensionality reduction and feature importance concentration. However, the performance of LightGBM was still inferior in comparison to EfficientNet-B4. This might be attributed to the fact that HOG is sensitive to the scale and orientation of images. As EfficientNet-B4 is trained with rotated images, it becomes rotation invariant and yields more robust predictions.

With EfficientNet's proven performance, in future research, we can further incorporate EfficientDet, which utilizes EfficientNet as its backbone network, with more annotations to locate where the small-scale solar panels are, instead of merely classifying the images. When considering applying EfficientNet to other applications where the orientation of the images is essential, we may want to use Fourier transformation on those images to capture their frequency components and modify the convolution operation in EfficientNet to be a pointwise product. For those images or videos that possess super low signal-to-noise ratio, it would be recommended to perform the truncated Singular Value Decomposition (SVD), Higher-Order SVD or Tucker's tensor decomposition to denoise or extract useful features before deploying them into EfficientNet.

Roles

Altamash Rafiq: Altamash produced a CNN model that did not make it into the final report as it was outperformed by this report's EfficientNet CNN implementation. Regarding this report, Altamash wrote the Introduction and Background sections of this report and contributed content to all the other sections.

Christy (Die) Hu: Christy worked with Aaron on the development of the CNN model with EfficientNet and AdvProp presented in this report. Regarding this report, Christy collaborated with Aaron and Srishti in writing major portions of the Methods and Result sections. Christy and Aaron wrote the abstract and conclusion as well.

Ronald: Ronald is responsible for visualizing and providing comparative insights about the performances and results of both models as well as the data section of this report. Worked on the non-neural net models which included a Gradient boosted classifier and SVM, but were not chosen as candidate model for this report.

Srishti Saha: Srishti was the primary developer of the gradient boosting machine model presented in this report. She also created an SVM model which did not make it into this report as the LightGBM model outperformed the same. Regarding this report, Srishti collaborated with Christy and Aaron in writing major portions of the Methods and Result sections.

Tien Yu (Aaron) Liu: Aaron was the primary developer of the CNN model with EfficientNet and AdvProp presented in this report. Regarding this report, Aaron collaborated with Christy and Srishti in writing major portions of the Methods and Result sections. Aaron and Christy wrote the abstract and conclusion as well.

References

[1] C Xie et al (2019) Adversarial Examples Improve Image Recognition
[2] Christopher Kanan et al (2012) Color-to-Grayscale: Does the Method Matter in Image Recognition?
[3] Guolin Ke et al (2017) LightGBM: A Highly Efficient Gradient Boosting Decision Tree
[4] IJ Goodfellow et al (2013) Maxout Networks
[5] J Yosinski et al (2014) How transferable are features in deep neural networks?
[6] J Yu et al (2018) DeepSolar: A Machine Learning Framework to Efficiently Construct a Solar Deployment Database in the United States
[7] J. M. Malof et al (2017) A Deep Convolutional Neural Network with Pre-Training for Solar Photovoltaic Array in Aerial Imagery.
[8] JM Malof et al (2016) Automatic detection of solar photovoltaic arrays in high resolution aerial imagery.
[9] K He et al (2015) Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification
[10] Kieran G. Larkin (2016) Reflections on Shannon Information: In search of a natural information-entropy for images
[11] M Tan et al (2019) EfficientDet: Scalable and Efficient Object Detection
[12] M Tan et al (2019) EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks
[13] Mathew George et al (2013) Object Detection using the Canny Edge Detector [14] Mustafa Ustuner and Fusun Balik Sanli (2019) Polarimetric Target Decompositions and Light Gradient Boosting Machine for Crop

Classification: A Comparative Evaluation [15] N. Dalal et al (2005) Histograms of oriented gradients for human detection [16] Niall O' Mahony et al (2019) Deep Learning vs. Traditional Computer Vision [17] NS Keskar et al (2017) Improving Generalization Performance by Switching from Adam to SGD [18] S Ioffe et al (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift