Automating the Identification of Illegal Human Activities in the Amazon Rainforest

Anisha Goel
Stanford University
anisha14@stanford.edu

Jacky Lin
Stanford University
jackylin@stanford.edu

Charles Zhang
Stanford University
masarain@stanford.edu

1 Introduction

Forest depletion is among the leading factors exacerbating climate change. The United States Geological Survey provide daily satellite images of the entire planet, including the Amazon rainforest, to the public. This project aims to gather those images, and (1) identify sensitive areas and (2) areas with illicit activities, automating the process of identifying destructive human activities to allow governments and NGOs to respond proactively. For this paper, we chose to focus on the Amazon rainforest, and specifically focus on activities such as farming, logging, and mining. We will perform a series of feature extraction on each image and assess their efficacy with a logistic and softmax regression classifier. Then, we will compare those results to those obtained by a K-means algorithm and neural networks, discussing the accuracy results and steps needed to improve the performance of these models for the final paper. Moreover, we will stitch multiple images to create a bigger neighborhood area map using the classified images. This will serve as the input to neural network to provide us high risk areas given the content in the image.

2 Related Work

One of the leading papers using AI in satellite image identification was published by Neal Jean et al. at Stanford University, where they used high-quality satellite images to predict poverty levels in developing countries in Africa [2], correlating certain features such as vegetation levels, water presence, and presence of straight lines (indicative of roads, bridges, and buildings) to economic wealth. We have also seen various researchers using satellite images with CNNs to detect the presence of golf courses in various precincts [3]. One of the big limitations, however, as noted by Basu et al., is the high variability in satellite images [4]. Thus, the model that Jean et al. used to obtain their poverty levels do not scale due to the fact that granular satellite images are expensive and not readily accessible, unlike the images provided by the United States Geological Survey, which are readily accessible, but not granular. The granularity of the dataset we will used is much more representative of the dataset available to governments than the one used in Neal Jean et al. Our project, therefore, will train models on datasets that are representative of the datasets local governments and NGOS using AI in satellite imagery would use. Further, one of the authors is an expert who performs image analysis and preprocessing of images for a Machine Learning company, and we will use the features extracted from the preprocessing step and evaluate its efficacy and compare the accuracy to that obtained by a neural network.

3 Datasets and Features

3.1 Data

Our data consists of satellite images (each of size 256 x 256) of the Amazon Rainforest provided by Planet Labs, a San-Francisco private Earth imaging company and our labels are tags of what that image corresponds to (for instance, “habitation”, “agriculture”, “conventional mine”, etc.). We divided the image chips into 36,000 train, 2,000 validation, and 2,000 test. The labels for the images were crowd-sourced and hand-labeled, and thus there is noise within the 40,000 image chips. Each chip, with a resolution of around 4 meters (i.e. each pixel represents 4 meters of land) encompasses around 220 acres of land area, with all 40,000 images chips encompassing more than 74 million acres of land in or near the Amazon Rainforest.

3.2 Feature Extraction

We manually performed a series of features extraction we thought would be helpful features with our Softmax regression and k-means algorithm We extracted the following features below.

White Balance Ratios White balancing is a process wherein we find the scaling ratios to find how far the grey color values, including white and black, are off from the actual white point of the image. It is define by two ratios $mean(G)/mean(R)$ and $mean(G)/mean(B)$. This feature allowed us to represent mean RGB in ratios to identify the relation between R, G, and B channels. The White Balance Ratios will be helpful in determining cloud cover.

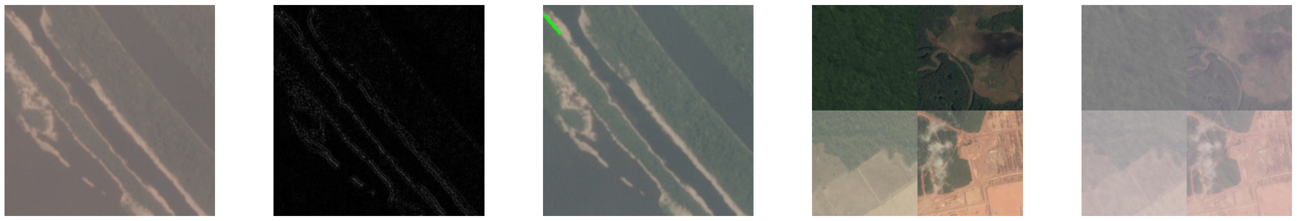

(a) (b) (c) (d) (e)

Figure 1: (a) Original Image from training dataset. (b) Laplacian response of (a) [zooming into the photo will reveal light streaks of lines]. (c) Hough Transform response of (a). (d) Concatenate 4 samples. (e) Blended with hazy image.

Mean luma and mean G Luma of a pixel represents the brightness value i.e. linear Y channel of YUV color space. We calculated Y using [4]: $Y = 0.299R + 0.587G + 0.114B$ Y channel is used to distinguish the images that contain higher intensity (candela / meter square) values such as clouds and haze. Mean of G channel is use to distinguish green features like trees, pasture, and grass land.

Mean of Laplacian filter Laplacian filter in simple terms allows edge detection by computing the rate of change in the values of color values. This feature is used to distinguish images that contain details such as trees, rivers, pasture, logging, mining, habitation etc as compare to cloudy images that do not contribute to our final goal.

Hough transform Hough transform allowed us to find lines and curves. For our use case, the lines detected are the man made structures such as roads, agricultural land and buildings. The curves detects the natural features such as rivers and rough roads.

Stitching images as Maps We created local maps to increase our dataset for the final goal to predict the high risk areas. We tried three ways of stitching: 1. Stitching based on features, but the images in the dataset did not have enough overlap of features. 2. Stitching based on geotag information in tiff images, but images did not contain any geographical information. 3. Concatenating images. This is the simplest and easiest way of creating local maps. Moreover, applied blending if there is an Image with clouds or haze is present. We created labels corresponding to these images based on the labels of the original images that are used to construct a given stitching image.

4 Methods

4.1 Logistic Regression

Logistic regression is a way of deciding whether a datapoint belongs to category A or category B. To do so, we use a hypothesis function $h_\theta(x)$. When a datapoint x is given to the hypothesis function, if $h_\theta(x) > 0.5$, we classify it as category A, and if $h_\theta(x) < 0.5$, we classify it as category B. More formally, $h_\theta(x)$ is defined as follows: $h_\theta(x) = \frac{1}{1+\exp(\theta^T x)}$

where θ is our parameter vectors that we wish to find to make $h_\theta(x)$ as accurate as possible. In order to find θ, we use the gradient descent algorithm on the loss function. For logistic regression, the loss function is defined to be the log likelihood function l, $l(\theta) = \sum_{i=1}^{n} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)})$ where $x^{(i)}$ is the i^{th} datapoint, and $y^{(i)}$ is the i^{th} label.

4.2 Softmax regression

Softmax regression is extending logistic regression for classifying amongst multiple categories. This means that our hypothesis function now outputs a vector, where each vector element corresponds to the probability that the datapoint belongs to that specific category.

$$h_\theta(x) = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_k \end{bmatrix} = \begin{bmatrix} \frac{\exp(\theta_1^T x)}{\sum_{j=1}^{k} \exp(\theta_j^T x)} \\ \frac{\exp(\theta_2^T x)}{\sum_{j=1}^{k} \exp(\theta_j^T x)} \\ \vdots \\ \frac{\exp(\theta_k^T x)}{\sum_{j=1}^{k} \exp(\theta_j^T x)} \end{bmatrix}$$

As before, we find the θ_j via gradient descent on the log likelihood function $l(\theta) = \sum_{i=1}^{n} \prod_{l=1}^{k} \left(\frac{\exp(\theta_1^T x)}{\sum_{j=1}^{k} \exp(\theta_j^T x)} \right)^{1\{y^{(i)}=l\}}$

4.3 K-Means

We will run K-Means on the features extracted (described above) for each image to produce two centroids, classifying if the image is cloudy (whose label is fully cloudy, partially cloudy, or hazy). Formally, we want to find

$$\arg_u \min \sum_{i=1}^{k} \sum_{\mathbf{x} \in u_i} \|\mathbf{x} - \mu_i\|^2 ,$$

where u is a cluster and x is a datapoint associated with that cluster. To perform K-Means, we first randomly initialize two clusters u_1, u_2. Next, we for every example i, we set $c^{(i)} := \arg\min_j \left\| x^{(i)} - \mu_j \right\|^2$ and for each cluster j, we set $\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)}=j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)}=j\}}$, repeating until convergence. We perform K-Means for 20 iteration and take the smallest loss.

4.4 Neural Networks: Regular Feed-Forward and Convolutional Neural Network

We will flatten each image into a single array (of size $256 \cdot 256 \cdot 3 = 196,608$ and feed it to the feed-forward neural network. We trained a standard feed-forward network where each neuron in the hidden layer and the output were a sigmoid function. After hyper-tuning the model parameters, we find that a neural network with 5 hidden layers of 20, 15, 10, 5, 1 parameters, respectively, led to the smallest soft margin loss when classifying whether or not each chip was cloudy (which includes fully cloudy, partially cloudy, and hazy). The soft margin loss for the output layer will be calculated as $\text{loss}(x, y) = \sum_i \frac{\log(1+\exp(-y[i]*x[i]))}{|x|}$ and the activation of neuron will be $\text{ Sigmoid } (x) = \frac{1}{1+\exp(-x)}$.

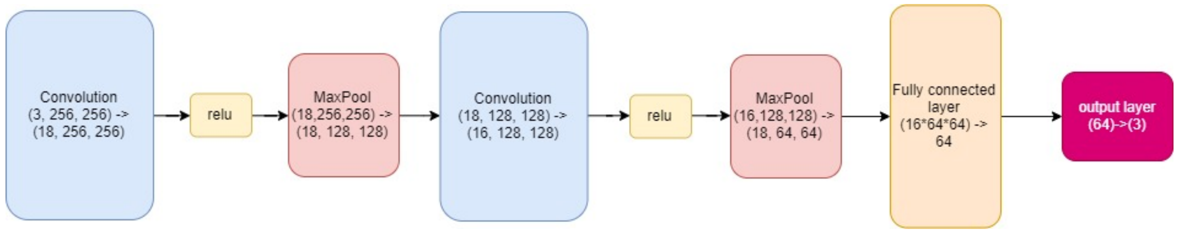


Figure 2: CNN architecture

The CNN is composed of two convolutional layers with one hidden fully connected layer and uses Cross Entropy Loss: $-\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_k^{(i)} \log \hat{y}_k^{(i)}$ for n examples and K classes. The convolutional layer are composed of 24 (or 18) and 16 filters respectively, with relu as the activation function after and then max pool layer to downsample the outputs by 4x. When we had only one convolutional layer with 18 or 24 filters, the model was overfitting to the training data. Increasing the complexity of the model by adding one more convolutional layer helped. Moreover, we added L2 regularization on model parameters to help avoid overfitting.

We have two fully connected layers, one as hidden and the other is output layer. We have tested the fully connected layer with two different activation function, relu and sigmoid function, and output layer with linear and softmax activation function. As shown in the results the difference in using different activation functions is negligible. We, also, tested the fully connected layer with 64, 128, and 256 neurons. When we made the system complex by increasing the number of neurons, the model was overfitting to the training data. We used cross entropy loss to calculate the loss for the model. We have developed CNN to take advantage of GPU whenever the hardware is available to train the model on. This helped us with running multiple experiments in a short period of time.

5 Experiments/Results/Discussion

5.1 Regular Logistic Regression and Softmax Regression

We used logistic regression to find features that would best differentiate images with clouds in them from images without clouds. These features performed decently in differentiating cloudy, or partly cloudy or hazy images from the rest of the dataset. The overall accuracy of the logistic regression classifier was 74% for detecting cloudy vs non-cloudy images.

Table 1: Logistic regression performed for clouds, accuracy of 0.74

	Cloudy label	Non-cloudy label
Predicted cloudy	252	63
Predicted Non-cloudy	197	488

We also attempted to extract features for classifying between human activities and no human activities. We used features such as the Laplacian filter and the Hough transform as a heuristic for analysing the number of edges/straight lines. This was based on the idea that human dwellings or human activities tend to have rectangular shapes, or artificially straight lines. However, in this regard, we were much less successful in classifying the images. Here are the results for finding roads (a destructive human activity) in the Amazon:

Table 2: Logistic regression performed for road, accuracy of 0.569

	Road label	Non-road label
Predicted road	174	403
Predicted Non-road	28	395

We applied the classifier on the rest of the labels, and found that it was especially inaccurate for labels that are rare in occurrence, such as blooming. Based on these experiments, we decided to narrow the scope of classification. Instead of classifying each chip (each image) as one of the seventeen different labels, we decided to simplify it into classifying each chip into one of three categories: cloudy, no human activities, some human activities. The cloudy categories will now encompass cloudy or partly cloudy or hazy labels. No human activities include the primary, river, blooming and bare ground labels. The some human activities include agriculture, habitation, road, cultivation, slash and burn, conventional min, artisanal mine, and selective logging. Reducing the number of categories will reduce the complexity of our classifying algorithm and give us more time to explore each category further.

Using the smaller scope, we trained a softmax regression model which gave us the following result:

	Cloudy	Human	Non-human
Predicted cloudy	379	134	74
Predicted human activities	69	205	62
Predicted no human activities	131	227	719

Overall, softmax regression had an accuracy of 0.6515. It has particular difficulty in classifying images with human activities. It also has a tendency to categorize it into the "no human activities" bucket. In fact, the softmax algorithm is over classifying into the "no human activities". This is a good sign that we were not able to find features that are particularly good at distinguishing between the "human activities" and "no human activities" bucket.

5.2 K-Means

Table 3: K-Means Classification, accuracy of 0.72

	Cloudy label	Non-Cloudy label
Predicted Cloudy	39	540
Predicted Non-Cloudy	0	1401

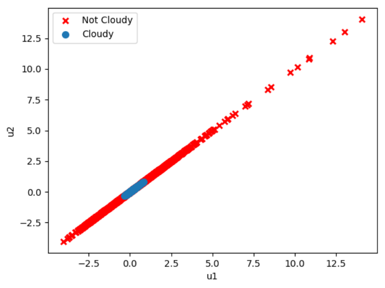


Figure 3: PCA Dimension Reduction to 2 Dimensions

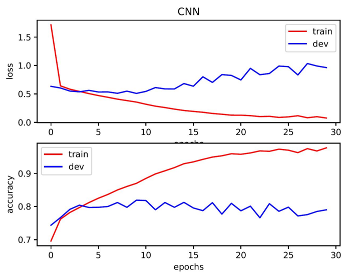
The results of the K-Means are shown in table 3. We see that conditioned that the image chip is cloudy, the k-means classification will classify the algorithm to be cloudy with high accuracy, but given that the image is non-cloudy, the classification will only classify the image chip accuracy around 72% of the time. To understand why, we performed principal component analysis on the features extracted to gain a better understand of the classification. Specifically, we maximized the variance we get by projecting all our features to two dimensions, parameterized by u_1 and u_2, maximizing $u^T\left(\frac{1}{n}\sum_{i=1}^n x^{(i)}x^{(i)T}\right)u$ for two unit vectors and renormalize the dataset. We then calculate the variance that can be explained by u_1 and u_2 and we find that they explain approximately 98% and 1% of the variance in the dataset. We plot our principal components and we see why the K-Means classification accurately predicts cloudy labels but does not as accurately predict non-cloudy labels.

5.3 Regular Feed-Forward Neural Net and Convolutional Neural Net

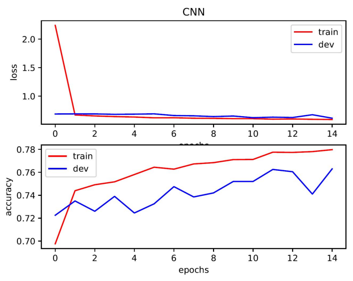
Running the regular neural net, minizíming the soft-margin loss, we find that the accuracy of the test set was exactly equal to the percentage of non-cloudy images. That is to say, our model strictly predicts that all the images are non-cloudy, where we have that non-cloudy images make up the vast majority of the training set. Obtaining such poor performance on identifying even the presence of clouds, we decided to proceed forward with a CNN and significantly increase the complexity of the architecture, taking advantage of a GPU.

Table 4: CNN classification, accuracy of 0.81 on test images, 0.76 on stitched images.

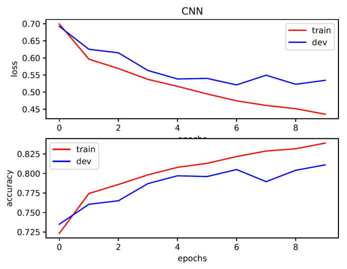
	Cloudy label	Human intrusion label	No-human intrusion label
Test data	0.76	0.82	0.82
Stitched data	0.46	0.83	0.51



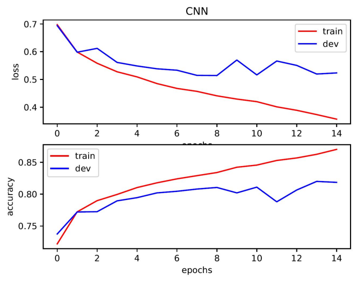
(a)



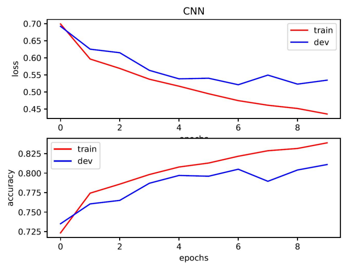
(b)



(c)



(d)



(e)

Figure 4: (a) CNN with single convolution layer of 24 filters and 256 neurons in fully connected layer. (b) CNN with single convolution layer of 24 filters and 128 neurons in fully connected layer. (c) CNN with two convolution layers of 24 and 16 filters with 64 neurons in fully connected layer. (d) CNN with two convolutional layers, 64 neurons in fully connected layer with partly cloudy and cloudy are both considered cloudy (e) CNN with two convolutional layers, 64 neurons in fully connected layer with partly cloudy and cloudy are considered as two different labels as partly cloudy images still have data that can be classified as human and non-human intrusive.

For CNN model results, we used 36000 samples for training, 2000 for dev, 2000 for test, and 2000 stitched data. The CNN model we chose is 2 layer convolutional layer, 64 neurons in fully connected layer with sigmoid activation function. The accuracy of CNN model on dev data fluctuates a lot as the model improves for the training data as we progress through epochs. The one of the reasons for such results is that the data is non-uniformly distributed and very noisy. Moreover, we ran the trained model on the calculated stitched images to check the robustness of the model and it performed good on average. As it can be seen from the results, as we increased the complexity of the models the accuracy improved.

6 Next Steps

As seen from the results of CNN, the model can be made more complex and add more layers of convolution filter to achieve better accuracy. We, also, would like to train the CNN on stitched images with original images to make the model more robust. The model can be improved by learning linear combination of more than one label, to improve on the results and make use of images with haze and clouds. We would also generalize the model to other parts of the world. To achieve this we would need to increase the robustness of the model by adding more noisy data like image of a white chip and train the model to not categorize white objects as clouds only.

7 Reference, Contribution, and Code

[1] Forests Issues & Threats. *Greenpeace* [online]. Available form: https://www.greenpeace.org/usa/forests/issues/

[2] Jean, N., Burke, M., Xie, M., Davis, W. M., Lobell, D. B., & Ermon, S. (2016). Combining satellite imagery and machine learning to predict poverty. Science, 353(6301), 790-794.

[3] Miyamoto, H., Uehara, K., Murakawa, M., Sakanashi, H., Nasato, H., Kouyama, T., & Nakamura, R. (2018, July). Object Detection in Satellite Imagery Using 2-Step Convolutional Neural Networks. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium* (pp. 1268-1271). IEEE.

[4] Basu, S., Ganguly, S., Mukhopadhyay, S., DiBiano, R., Karki, M., & Nemani, R. (2015, November).Deepsat: a learning framework for satellite imagery. In Proceedings of the 23rd SIGSPATIAL interna-tional conference on advances in geographic information systems (p. 37). ACM.

[5] YUV article on wikipedia. Available form: https://en.wikipedia.org/wiki/YUV

Anisha: Feature Extraction, pre-processing and stitching of images, CNN
Charles: Infrastructure support and Logistic Regression Classifier
Jacky: K-Means and Neural Network
All authors contributed to the milestone, poster, and final paper.

Link to code: https://github.com/masarain/cs229TreeHugger/