Understanding the Amazon from Space

Derek Ching, Yuetian Li, Guanzhou Song
Khoury College of Computer Sciences
Northeastern University
Boston, MA, USA
{ching.de, li.yuet, song.guan}@husky.neu.edu

Abstract

The purpose of this project is to find out how the rainforest of the Amazon deteriorates by classifying satellite images given by the company, Planet, and correctly identifying a combination of the 17 different tags to an image. These include several land and weather tags. In order to get a good accuracy, we explore various image processing techniques to make the images more clear and extract several crucial statistics from the original and processed image sets. Afterwards, we take these statistics which are then combined into a single data frame from which we are able to use some well-known classifier algorithms to apply on to it, and output a corresponding confusion matrix, and a standard classification report (precision, recall, F-1 score, and support). By doing this, we can analyze the accuracies of the prediction of labels in images, across the three classification algorithms - Decision Tree, Random Forest, and CNN.

Introduction

The Amazon rainforest suffers on a daily basis due to human intervention of its natural ecosystem. The creation of farms, roads, and mining facilities take place in the Amazon rainforest, and because of this, we can see a drastic breakdown in indigenous wildlife and forests over the past decades. This has sparked a huge debate amongst many entities like Save the Amazon Rainforest Org, Rainforest Foundation US, and many other organizations who support the natural order of the Amazon rainforest. Finding the reason for the rainforest's deterioration and rate of decline has never more crucial, and through the emergence of new technologies and machine learning, we can start to understand the impact that humans have on the land and weather.

Currently, in the field of machine learning and artificial intelligence, image classification is an emerging topic. With the use of satellite images, we are able to now gather large amounts of data that can take several high quality photos of our planet. Today, it is essential to be able to identify various labels without having to expend manual human labor to filter through thousands upon thousands of images to determine a suitable context for the image. Governments hire several highly trained analysts to do the job of manually sorting through the data and to assess the images, but it is a difficult task to have them all completely agree on what is present in a given satellite image. In order to process the multitude of these images, the need for machine and deep learning models are needed to simulate how the decision making of the human mind works.

Background

Planet is the designer and builder of the world's largest constellation of Earth-imaging satellites, will soon be collecting daily imagery of the entire land surface of the earth at 3-5 meter resolution. While considerable research has been devoted to tracking changes in forests, it typically depends on coarse-resolution imagery from Landsat (30 meter pixels) or MODIS (250 meter pixels). This limits its effectiveness in areas where small-scale deforestation or forest degradation dominate. Furthermore, these existing methods generally cannot differentiate between human causes of forest loss and natural causes. Higher resolution imagery has already been shown to be exceptionally good at this, but robust methods have not yet been developed for Planet imagery.

Image classification models needs to be powerful, and robust. The use of GPU's and high CPU count systems are now more than necessary to process batches of images, this is called a compute engine. In this project, we use Google Cloud's Compute Engine in order to set up a sturdy-enough working environment that can handle processing the 40,000 high quality satellite images provided by the company, Planet, with as little latency as possible. A typical problem of image classification is that training the model takes hours upon hours to do, and finding an issue or bug often leads to waste of time and resources. By utilizing neural network, a form of unsupervised learning, large computation is required due to its many layers of complexity. Additionally, often times, using a neural network is much more efficient than its supervised learning counterparts and is able to achieve a greater degree of human-like accuracy.

In this paper we will predict the results of classifying and predicting the labels to an image, where there exists any combination of labels per image. We will be using the trained input image set that contains 40,000 JPG and TIF images to build up our complex image pre-processor, which we can then clean and turn into data for weather and land prediction analysis. The prediction of labels is implemented through the use of Decision Tree, Random Forest, and CNN classifiers. Decision Tree and Random Forests are used to test the accuracy of unsupervised methods which is between 82-89% depending on the how large the training data we use, while in a CNN classifier the accuracy is boosted to 90%+. This shows that using a neural network we are able to get slightly higher accuracy however at the cost of longer computational times.

In order to get higher benefits with using the Decision Tree and Random Forest, we will be using Sklearn package to examine the optimal tree parameters to be used on both methods using Sklearn's GridSearchCV toolkit. We have provided for each classification method a standard classification report that calculates various scores such as precision, recall, f1-score, and support.

Related works

Classification task for satellite imagery has a long history. In 1998, It is demonstrated that Landsat imagery can be used to map forest clear cuts in the Pacific Northwest. Some researcher also combined MODIS and Landsat data to discover the change of forest cover in boreal forests and the Congo Basin. In 2009, Hilker et al. developed a new data fusion model specifically to analyze forest disturbance to improve the fusion of MODIS and Landsat data. Also, the Brazilian government has established their own real-time system DETER and PRODES to keep watch over deforestation. Traditional machine learning has also been introduced to satellite image classification. Gradient boost decision trees and SVM are applied on satellite images in order to classify land coverage by Baker et al. and Otukei et al.

In recent years, with the rise of deep learning, many other methods to approach satellite image classification are introduced. A hybrid deep neural network to do small object detection is declared by Chen et al. By altering the convolutional layers and pooling layers, they enhanced the ability of deep neural network on extracting features of different scales. Besides, Quintano et al. applied filtering algorithms as pre-classification to improve satellite image classification.

Project description

Our team discusses the performance of the Amazon Rainforest label prediction analysis in this paper. Our tasks are to compare and contrast different predictive modeling techniques to properly label a picture with its different land and weather tags. In this section, we will therefore discuss our approach to image pre-processing, image cleaning and analysis, and various classifier algorithms.

Data Preprocess

The data source used in this project is Planet's kaggle competition to provide the required training data which together is approximately 32 GB worth of data. This includes the following:

- train.csv - a list of training file names and their labels, the labels are space-delimited.
- {train/test}-tif-v2.tar.7z - tif files for the training and test set.
- {train/test}-jpg-additional.tar.7z - jpg files for the training and test set.

TIF images provide extra information about the infrared features of the satellite image, whereas JPG images show the natural light spectrum of the image.

There are a total of 17 different labels. But in order to simplify the problem, we can identify which labels correspond to weather and which labels correspond to land.

- Weather: Clear, Cloudy, Partly Cloudy, Haze.
- Land: Habitation, Bare Ground, Cultivation, Agriculture, Blow Down, Conventional Mine, Selective Logging, Slash Burn, Artisanal Mine, Blooming, Primary, Water, and None.

To summarize, there are 4 labels in "Weather" and 13 labels in "Land". Because land is fairly complex, and because our problem is to identify which land labels have been affected by human intervention, we can further simplify the problem by categorizing the "Land" tag as the following: Primary, Water, Other, and None (where "Other" refers to any Land labels that have human footprints).

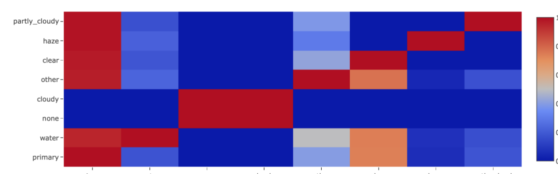

Figure 1: EDA Label Weather Correlation Results

Figure 1 represents the correlation of labels in the trained image set, we can see that any region which is marked by red shows a high correlation between the x-axis label and its y-axis label counterpart. We can see that primary tends to show up the most with combination of other weather labels.

Land does not need a correlation analysis, as we have already simplified our process of land labels, and there can only be 1 out of the 4 land labels (Primary, Water, Other, and None) at a time.

Image Data Pre-processing

Image pre-processing is used only in Decision Tree and Random Forest Models because CNN is able to learn from its many different layers, and extract which features are useful to classify the image. In this section, we will talk about the techniques of image pre-processing used in the Decision Tree and Random Forest models.

Another problem that needed to be addressed is the blurriness of some images. This would help to improve the extraction of land features of pictures with haze. We used Kaiming He, Jian Sun, and Xiaoou Tang's Single image haze removal using dark channel prior to recover a haze-free image. It uses OpenCV package in Python to detect the thickness of the

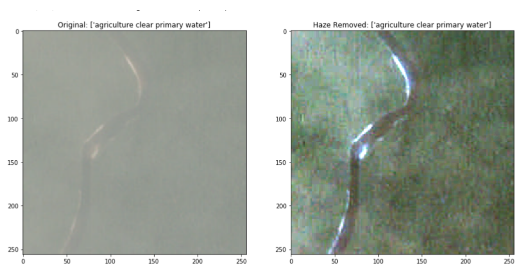

Figure 2: EDA Label Weather Correlation Results

haze, and remove it from the original image. We can see an example of this being used on one of our satellite images in Figure 2.

Basic assumptions were made about the spectral analysis of images. The JPG images provided a lot of important information about RGB data, which we can correlate from logic that a lot of blue means water, green means vegetation and forest, white means clouds. Each RGB has a "pixel value" and if the value is higher, then it meant a stronger color intensity of that region. Extraction of this color variance from each satellite image is then made possible. As a result, it is added it as a feature to our data frame. However, using the TIF image data set we could identify features that couldn't be split through natural light, and use that to further extract more information.

In using infrared images taken from the TIF data set, we could give our models further "intelligence" by extracting water features and vegetation features. "Water" features were typically define by removing blue frequency in the IR values, whereas "Forest" features were defined by removing the red frequency from the IR values. Below are the formulas used to calculate those indexes:

- Water index = (Blue - Infrared Light) / (Blue + Infrared Light)
- Forest index = (Infrared Light - Red) / (Red + Infrared Light)

Edge detection in the images can be analyzed using Laplacian and Sobel. Laplacian outputs the color gradients in the images.

$$\Delta src = \frac{\partial^2 src}{\partial x^2} + \frac{\partial^2 src}{\partial y^2}$$

Sobel outputs the edges of the color gradients in the x and the y directions.

$$G = \sqrt{{G_x}^2 + {G_y}^2}$$

where G is the approximation of the gradient, and G_x and G_y are the first order derivatives of the x and y edges.

Since the Laplacian is a second order derivative and increase in orders of differentiation cause spatial changes, there is more sensitivity to noise in the images than the Sobel images. The majority of the pictures may not have large color gradients. As a result, many of the images are dominated by their respective label components and relatively few satellite images contain edges.

After image cleaning and image data pre-processing steps are completed, we can now obtain image statistics that we can form as features that will be the inputs to our predictive models. We obtain a variety of statistic information to add to a data frame, including the mean, variance, standard deviation, kurtosis, and skew of the image colors. Next, we will gather the mean and variance of the image edges.

Model evaluation

In this section, we discuss the implementation and performance of our predictive modeling techniques. In order to show the performance of a model, precision, recall and F-1 score are calculated.

Precision is the ratio of $\frac{tp}{tp+fp}$ where *tp* is the number of true positives and *fp* the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative.

Recall is the ratio $\frac{tp}{tp+fn}$ where *tp* is the number of true positives and *fn* the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples.

F-1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0. Below is the formula for calculating a F-beta score.

$$(1+\beta^2)\frac{pr}{\beta^2 p + r} \text{ where } p = \frac{tp}{tp+fp},\ r = \frac{tp}{tp+fn}, 0 \leq \beta < +\infty$$

F-1 score is when β = 1. For the overall evaluation of the model, the F-1 score is the optimal option as it reflects the overall accuracy of the model.

Decision Tree

The idea of the decision tree algorithm is to obtain a highly probable result by constructing a tree-like decision graph based on the known probability of occurrence of various situations. It can be used for project risk evaluation, judgment of behavioral feasibility, etc. It is a machine learning algorithm that intuitively uses probability analysis. This decision branch draws a graph much like a tree's branches, so it is called a decision tree. It is one of the predictive modeling methods often used in statistics, data mining and machine learning. A target variable can be called a classification tree using a set of discrete value tree models; in these tree structures, leaves represent class labels, and branches represent the connections that lead to the function of these class labels.

In data mining, decision trees are one of the common algorithms. Each leaf represents the value of the target variable, given the value of the input variable represented by the path from root to leaf. It can simply represent a classification situation. Suppose all input features have a finite discrete domain and there is a single target feature called a "classification." Each element of the classification domain is called a class. A decision tree or classification tree is a tree in which each internal non-leaf node is marked with an input feature. The arc from the node marked with the input feature is labeled with each possible value of the target or output feature, or the arc leads to a dependent decision node on a different input feature. Finally, on the leaf nodes of the tree, there is a class or probability distribution. The composition of the tree can be created by dividing the original data set into subsets that are tested based on attribute values. This process is repeated recursively on each derived subset and is therefore referred to as a recursive partition. Recursion is done when the subset on the node has exactly the same value as the target variable, or when the segmentation is no longer adding value to the prediction. The decision tree top-down induction process is an example of a greedy algorithm, one of the most common strategies for learning decision trees from data. In data mining, decision trees can also be described as a combination of mathematical and computational techniques to help describe, classify, and generalize a given data set.

The classification tree is the decision tree that is applied to the classification problem. Classification is the process of recognizing, distinguishing, and understanding concepts and objects. Classification diagrams are designed to help create and ultimately visualize results. According to Brinton, in the classification table, the facts, the data, etc. are arranged so that each person's position relative to other people is easy to see, and there is no need to give a quantity, although quantitative analysis increases the value of the classification map. Karsten explained that in the creation of all graphs, the material to be displayed must be accurately compiled before drawing. In order to understand the classification table, we must study the connotation of classification and indexing. The classification art plays the role of "whole" and its "parts". "The ability to visualize together.

Compared to other data mining methods, decision trees have various advantages:

- Easy to explain. Decision trees are well understood and trees can be graphically displayed in a way that is not easily understood by experts.
- Can handle various types of digital and classified data. Many other machine learning algorithms can only be used to analyze data sets with only one type of variable. For example, a relational rule can only be used for nominal variables, while a neural network can only be used to convert numeric variables or classifications to 0-1 values.
- Little data preparation is required. Other algorithms usually require data to be standardized. Since the tree can handle qualitative predictors, there is no need to create dummy variables. 4. White box model. If a given situation is observable in the model, it can be explained by Boolean logic. In contrast, in the black box model, the interpretation of the results is often difficult to understand, such as the now popular artificial neural network.
- You can use statistical tests to verify the model. This can illustrate the reliability of the model.
- Do not make hypothetical training data or non-statistical methods for predicting residuals.
- It works well for large data sets. Even processing large amounts of data does not consume too much resources.
- More like the human decision-making process. This can be useful when modeling people's decisions or behaviors.
- Strong anti-collinearity. Other irrelevant features will be less common in order to be removed in subsequent runs.

The project's Decision Tree takes in 13 parameters including criterion of splitting measurement, the strategy used to choose the split at each node, maximum depth of the tree, the minimum number of samples required to split an internal node, and minimum number of samples required to be at a leaf node.

Random Forest

Random Forest is essentially the upgraded version of the Decision Tree model by constructing multiple decision trees from one dataset by splitting the dataset into different subsets, and in each subset, randomly pick different *data* and *features*. In practice, it is proved that this operation can help to improve the predictive accuracy and control over-fitting.

Random forest refers to a classifier that uses multiple trees to train and predict samples. It is a collection learning method that can be used for classification as well as for regression and other tasks by constructing multiple decision trees at training time and outputting classification categories in classification problems, or outputting average predictions in regression problems. value. Random decision forests correct the problem of over-fitting a decision tree on its training set.

The first random decision forest algorithm was created by Tin Kam Ho using a random subspace method. In Ho's formula, this is a classification method that implements Eugene Kleinberg's random discrimination. The algorithm was later developed by Leo Breiman and Adele Cutler. This combines the idea of bagging algorithm proposed by Breiman with the random selection feature, first introduced by Ho, and later introduced independently by Amit and Geman to construct a set of decision trees with controlled variables.

The general method of random decision forests was first proposed by Ho in 1995. She determined that if trees were randomly restricted to sensitive features, then it would be accurate to split the trees with obvious tendencies because they did not suffer from excessive training. Subsequent work draws conclusions along the same line of thought, and other methods of division, as long as, they are randomly forced to be insensitive to certain feature dimensions. They will be similar in behavior. Note that observing a more complex classifier becomes almost monotonously more accurate, which is in stark contrast to the fact that the complexity of the classifier can only be increased to a certain level before being overfitting. The resistance of the forest method to overfitting is explained in Kleinberg's stochastic discriminant theory.

The early development of Breiman's concept of random forests was influenced by Amit and Geman, who introduced a search for a single tree based on a random selection of a subset of available decisions when splitting a node. The idea of Ho's random subspace selection also has an impact on the design of random forests.

In this approach, a piece of forest consisting of multiple decision trees is generated and introduced into each tree or node by introducing changes between the trees before the training data is projected onto the randomly selected subspace. Finally, the idea that random forests also include randomized node optimization was first introduced by Dietterich. He proposed that each node's decision is chosen by a random process rather than deterministic optimization.

The earliest explanation for the rationality of random forests came from Leo Breiman. In his paper, he explained a method of constructing unrelated tree forests using CART, combined with random node optimization methods and bagging algorithms. In addition, this paper combines several components, some previously known and some novel ingredients that form the basis of modern practices in random forests, especially in estimating generalized errors and by arranging measured variables. Both of these methods use the out-of-packet error rate in the bagging algorithm. The report also gives the first theoretical result of a random forest, the boundary of its generalization error. It depends on the robustness and relevance of the decision tree in the forest.

The project's random forest classifier takes in 17 parameters which includes the number of trees in the forest, the function to measure the quality of a split, the maximum depth of the tree, the minimum number of samples required to split an internal node, and the minimum number of samples required to be at a leaf node.

Convolutional Neural Networks

Convolutional Neural Network (CNN) is a class of deep neural networks, most commonly applied to pattern recognition for image data. The CNNs are regularized version of multilayer perceptrons, and can automatically extract the correlations between the image hidden features and learn the representation of data. As the computing resource greatly increase in the recent years, CNN with deep layers shows its enormous potential on image classification, and is now broadly regarded as one of the most powerful tool for computer vision problems.

Convolutional networks are inspired by biological processes, because the connections between neurons are similar to the tissues of the visual cortex of animals. Individual cortical neurons respond to stimuli only in restricted areas of the field of vision known as the receptive field. The sensory fields of different neurons overlap partially, making them cover the whole field of vision. Compared with other image classification algorithms, CNN uses relatively less preprocessing. This means that the network learns the filter designed by hand in the traditional algorithm. This prior knowledge and human effort independent of feature design is a major advantage.

The CNNs that the model used includes VGG-16, VGG19, ResNet, NASNet, Inception ResNet V2, Inception V3, and Xception, which are CNNs that have been proved to be excellent models, and tuning is required for the optimal result because of the difference between datasets. Due to limited resources, parameters from paper were chosen and proven to provide the optimal results for image classification.

VGG-Net The main idea of VGG-Net is a thorough evaluation of networks of increasing depth using an architecture with very small convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers. These findings were the basis of University of Oxford's Visual Geography Group's ImageNet Challenge 2014 submission, where they secured the first and the second places in the localization and classification tracks respectively. They also show that their representations generalize well to other datasets, where they achieve state-of-the-art results.

ResNet ResNet is a residual learning framework to ease the training of networks that are substantially deeper than those used previously. It explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. It is proved with comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. ResNet was evaluated on the ImageNet dataset with a depth of up to 152 layers: 8x deeper than VGG nets but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. ResNet is also proved to be efficient on CIFAR-10 with 100 and 1000 layers. The depth of representations is of central importance for many visual recognition tasks. Solely due to our extremely deep representations, ResNet obtain a 28% relative improvement on the COCO object detection dataset.

NASNet NASNet is a method to learn the model architectures directly on the dataset of interest. As this approach is expensive when the dataset is large, we propose to search for an architectural building block on a small dataset and then transfer the block to a larger dataset. The key contribu- tion of this work is the design of a new search space (which is called the "NASNet search space") which enables transferability.

Inception Net Inception Net is a series of deep neural networks that not only increase the depth but also increase the complexity of the neural network. The later version of Inception Net also combines many popular concepts from other deep neural network to make improvement, for example, the Inception ResNet, which combines ResNet. The main hallmark of this architecture is the improved utilization of the computing resources inside the network. By a carefully crafted design, we increased the depth and width of the network while keeping the compu- tational budget constant. To optimize quality, the architec- tural decisions were based on the Hebbian principle and the intuition of multi-scale processing.

Xception Xception is an interpretation of Inception modules in convolutional neural networks as being an intermediate step in between regular convolution and the depthwise separable convolution operation. In this light, a depthwise separable convolution can be understood as an Inception module with a maximally large number of towers. This observation leads us to propose a novel deep convolutional neural network architecture inspired by Inception, where Inception modules have been replaced with depthwise separable convolutions.

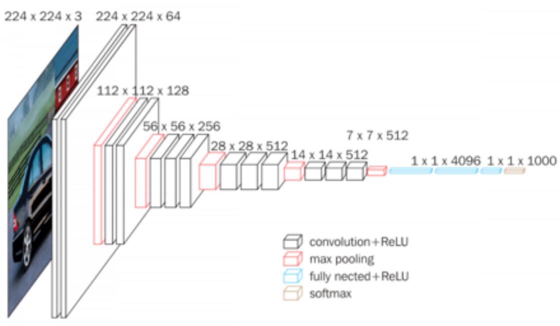

Figure 3: VGG16 Architecture

Figure 4: GoogLeNet network with all the bells and whistles.

Experiment

In our experiment, we will first find the best CNN model among 7 successful CNNs, and then make further comparison amaong Decision Tree, Random Forest and the representing CNN.

Implementation

This project code are deployed and conducted on Google Cloud Platform, as well as all experiment are fulfilled there. The online computing platform is used for improved computing ability. Implementation of data preprocessing, Decision Tree, Random Forest, and are based on Sklearn and CNN is based on the publicly available Tensorflow. Python 3 and various scientific computing packages such as NumPy, SciPy, Pandas are applied in this project to load and process the data, and visualization tools such as matplotlib and plotly are applied to visualize the dataset and the prediction results. For better visualization of the computing process, we use Jupyter Notebook so we are able to better visualize and analyze the result for each code block.

Thanks to GPU acceleration, which allowing us to perform training and evaluation on multiple GPUs which is reportedly 8-17 times better than CPU-based calculations. Multi-GPU training on GCP exploits data parallelism, and is carried out by splitting each batch of training images into several GPU batches, processed in parallel on each GPU. After the GPU batch gradients are computed, they are averaged to obtain the gradient of the full batch. Gradient computation is synchronous across the GPUs, so the result is exactly the same as when training on a single GPU.

For Decision Tree and Random Forest, grid search is applied for tuning the algorithm's parameters. Two parameters are tuned for Decision Tree, which are the maximum depth of the tree, and the minimum number of samples required to be at a leaf node. Three parameters are tuned for Random Forest, which are the number of trees in the forest, the maximum depth of the tree, and the minimum number of samples required to be at a leaf node.

For Convolutional Neural Networks, as stated above, we conduct 2 sections of experiments.

The first experiment is intended to show the performance of different training models on the same data set to determine the best CNN for this image classification task. The model used, VGG-16,VGG19, ResNet, NASNet Mobile, Inception ResNet V2, Inception ResNet V3, and Xception. So in this experiment, same training process (including data set), early stop and all other parameters, were used for different models, and comparison of their results are drawn evaluating their precision, recall and F-1 score.

The second experiment is intended to show the relationship between numbers of labels (complexity of data) and the result achieved. For image classification, we ran our training/testing process on three kinds of labels with the same image data source. The conducted experiment aims to receive the best result while at the same time minimizing the number of epochs to achieve an optimal result.

For the project image prediction, we run the CNN models with three different labels:

- Full label without any modification(18)
- Weather label (4 tags)
- Processed land label (4 tags)

For each different combination of tags within the label, the experimental model is trained with 32,000 images and 8,000 images for cross validation.

While training, an early stop method is used to achieve the optimal result and at the same time minimize the amount of resources used. From an earlier observation, 20 to 25 epoches with bath size of 128 usually converges.

Result

Table 1 shows that for the same data input, different models are all perform relatively well, and the best result is achieved by VGG-16 model with result in 0.9023 F-1 score. So in the following sections, VGG-16 will be our representation of CNN algorithm.

The table 2 shows precision, recall and F-1 score of the CNN(VGG-16) model perform on same data image with different label. It clearly shows that CNN model can perform great on the Amazon rainforest image classification problem. Precision may decrease with the increase of number of labels but it still around 90% which is close to the reported performance for the VGG-16 model. So it can show that the model is well-fit for our problem as image classification, even without any data pre-process.

The experiment shows that, for our image classification problem, CNN, together with the best model we can have as

Table 1: Comparison for different CNN models

	VGG-16	VGG-19	ResNet	NASNet Mobile	Inception ResNet	Inception V3	Xception
Parameters	138,357,544	143,667,240	25,636,712	5,326,716	55,873,736	23,851,784	22,910,480
Precision	0.8770	0.8728	0.8768	0.8795	0.8552	0.8606	0.8700
Recall	0.9520	0.9523	0.9412	0.9336	0.9360	0.9389	0.9405
F-1 score	0.9023	0.8994	0.8967	0.8951	0.8819	0.8862	0.8930

Table 2: Comparison for different labels

	Weather	Land (Processed)	Raw Label
Number of labels	4	4	18
Precision	0.9279	0.9341	0.876986235
Recall	0.9732	0.9759	0.952007164
F-1 score	0.9429	0.9440	0.902292296

VGG-16, can achieve around 0.9 on F-1 score which is better than other models. For the same data input, models are all perform well and VGG-16 is the most outstanding one. For the same model, with the increase of number of labels, F-1 score can slightly decrease but the result is still relatively great. So the experiment proves that CNN is a great model with high performance on image classification, and can be further explored and enhanced by tuning different parameters.

Figure 5 shows the decrease of the model's prediction loss after several epoch.

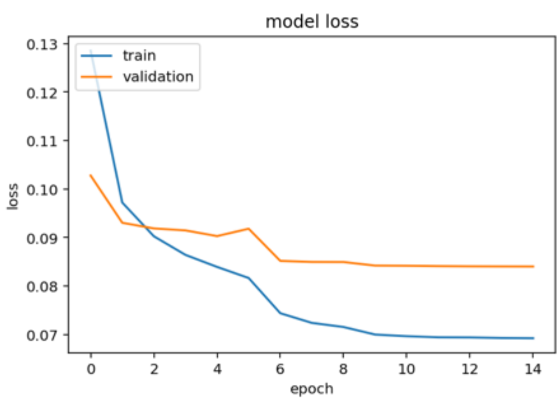

Figure 5: model loss after several epochs with batch size 128

The tables 3, 4 and 5 shows the 3 machine learning algorithm's results for weather prediction, land prediction and weather-land combined prediction with preprocessed data. As we can infer from the tables, the CNN is indeed the best algorithm among the three on this image classification task.

Conclusion

Overall, the conclusion that can be drawn from the table results show that CNN comes in first place, then Random Forest in second place, and Decision Tree in third place.

Our results proves that CNN is much better than the previous two model in separate weather and land label classification as well as in combined weather and label classification. This is aligned with current appreciation that CNNs have the ground breaking performance on pattern recognition and are the best algorithm for image classification tasks.

Table 3: Comparison for different algorithms

Weather	Decision Tree	Random Forest	CNN(VGG-16)
Precision	0.87	0.90	0.93
Recall	0.87	0.90	0.97
F-1 score	0.87	0.90	0.94
Land	**Decision Tree**	**Random Forest**	**CNN(VGG-16)**
Precision	0.92	0.94	0.93
Recall	0.90	0.91	0.98
F-1 score	0.89	0.91	0.94
Combined	**Decision Tree**	**Random Forest**	**CNN(VGG-16)**
Precision	0.90	0.92	0.92
Recall	0.88	0.90	0.97
F-1 score	0.88	0.90	0.94

Also, we may see that VGG maintains a great record on F-1 score among all the CNNs. Under F-1 score, the punishment of false negative is less than the other F-β evaluation functions such as F-2. The good balance between the precision and recall make it the most successful CNN under our standard.

Future work

In this paper, we designed and implemented three modeling techniques for weather and land labels, both weather and land labels separately and with combined labels. We created three different image classifiers, Decision Tree, Random Forest, and a Convolutional Neural Network. In this project we have implemented separate land and weather label classification analysis for all three models.

In the future, we would like to use object-oriented classification. In our current design, we use "Other" label in land to represent any human-intervention related labels such as Habitation, Bare Ground, Cultivation, Agriculture, Blow Down, Conventional Mine, Selective Logging, Slash Burn, Artisanal Mine, and Blooming. But we don't make a clear distinction between these various labels. The use of object-oriented classification would allow spatial recognition of groups of pixels to be correlated with objects which could provide better accuracies in these tags. This would also require more improvements and calculations to be performed on our image-preprocessing to include shadow density, and spectral feature calculations of various roads, mines, and other related scenery.

Reference

Safavian, S. R.; Landgrebe, D. 1991. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics* 21: 660 - 674.

Breiman, L. 2001. Random forests. *Machine learning* 45: 5–32.

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

He, K.; Zhang, X.; Ren, S.; Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR),* 770–778. IEEE.

He, K.; Sun, J.; Tang, X. 2011. Single image haze removal using dark channel prior. In *IEEE transactions on pattern analysis and machine intelligence,* 33(12):2341–2353.

Simonyan, K.; Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR,* abs/1409.1556.

Chollet, F. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* IEEE.

Zoph, B.; Vasudevan, V.; Shlens, J.; and Le, Q. V. 2018. Learning Transferable Architectures for Scalable Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 8697–8710. IEEE.

Baker, C.; Lawrence, R.; Montagne, C.; Patten, D. 2006. Mapping wetlands and riparian areas using land- sat etm+ imagery and decision-tree-based models. *Wetlands,* 26(2):465–474.

Otukei, J. R.; Blaschke, T. 2010. Land cover change assessment using decision trees, support vector machines and maximum likelihood classification algorithms. *International Journal of Applied Earth Observation and Geoinformation,* 12:S27–S31.

Cohen, W. B.; Fiorella, M.; Gray, J.; Helmer, E.; Anderson, K. 1998. An efficient and accurate method for mapping forest clearcuts in the Pacific Northwest using Landsat imagery. *Photogrammetric engineering and remote sensing,* 64(4):293–299.

Chen, X.; Xiang, S.; Liu, C.; Pan, C. 2014. Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geoscience and remote sensing letters,* 11(10):1797–1801.

Quintano, C.; Cuesta, E. 2010. Improving satellite image classi- fication by using fractional type convolution filtering. *International Journal of Applied Earth Observation and Geoinformation,* 12(4):298–301.

Hilker, T.; Wulder, M. A.; Coops, N. C.; Linke, J.; McDermid, G.; Masek, J. G.; Gao, F.; White, J. C. 2009. A new data fusion model for high spatial- and temporal-resolution mapping of forest disturbance based on landsat and modis. *Remote Sensing of Environment,* 113(8):1613–1627.

Dietterich, T. G. 2000. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Machine Learning,* 40(2):139-157.

Acknowledgment

1. All data are downloaded from the Kaggle competition homepage: https://www.kaggle.com/c/planet-understanding-the-amazon-from-space

2. For Professor and TAs: Please see our group's github repository for full code and all commit history: https://github.ccs.neu.edu/dching/CS5100_Final_Project