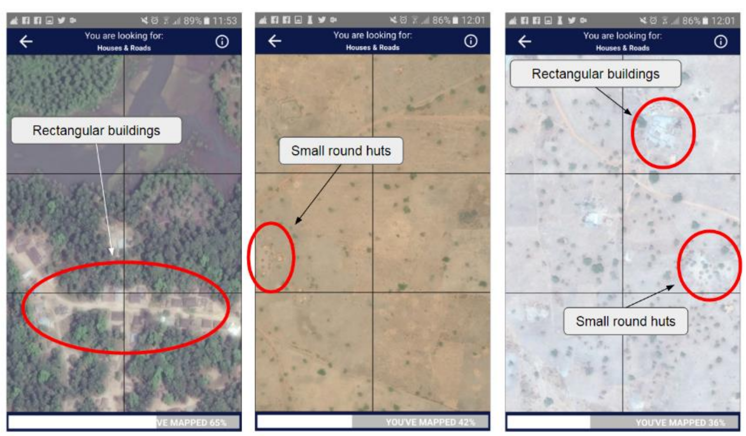
Machine Learning Engineer Nanodegree
Capstone Proposal

David Ghan
January 8, 2018

Deep Learning Satellite Imagery to Map Rural Communities

Domain Background

Deep Learning has become widely used in many industries to tackle a diverse number of problems, ranging from image classification to natural language processing. The power of the convolutional neural networks (CNNs) has allowed us to take advantage of the wealth of data that many organizations accrue. I propose to use CNNs in a geospatial manner, to identify structures and characteristics within a satellite image. My overall goal is to explore the power of using deep learning with publicly available imagery of rural areas in developing countries. There are many resources available to do this, including OpenStreetMaps, but I have narrowed my goal to classify images in areas where there have been humanitarian emergencies requiring improved satellite imagery to map communities. For this, I will take advantage of the publicly available API for MapSwipe, a nonprofit application that uses crowdsourcing to classify satellite images.



MapSwipe User Interface (link)

The Map Swipe application allows users to sort through and label unclassified satellite images of rural areas. These labeled satellite images are then used by international nonprofit organizations for better response to humanitarian crises. Areas that are labeled are typically found in under-developed regions. The projects MapSwipe conducts have varying target labels, ranging from classifying images with structures to images with roads or paths. I will limit my project to classifying MapSwipe images where the target was labelling satellite images when they have buildings, homes, or other related structures.

Problem Statement

This deep learning project will be addressed as a classification problem. The goal is to build a model that can be generalized in classifying satellite images from rural areas around the world. The performance of the model will be measured according to its accuracy in classifying the MapSwipe images.

I will extract the images with the help of from Github user philiptromans, which, I will mention, is extremely helpful Python code. With their repository I can extract classified MapSwipe images from specific projects to train, test and predict for my capstone project. Using this code, the images are extracted and separated according to three labels: 'built', 'empty', and 'bad imagery'.

Datasets and Inputs

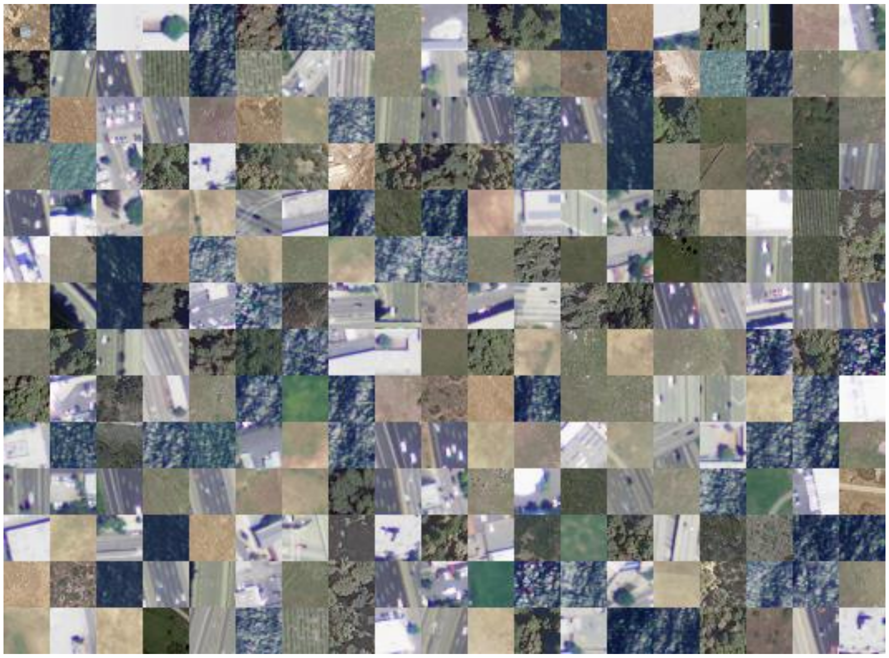
The dataset used will be extracted from the MapSwipe API and Bing Maps using Github user philiptromans' Python code. I will extract 13500 images of rural Malawi. This will consist of 9600 training images, 1500 validation images, and 2400 test images. Each image is 256 x 256px, or an area of 0.024km^2. The images are in 3-band color (RGB), but I will also convert them to 2-band grayscale during the feature engineering phase of the project. As mentioned above, the image labels are:

1. 'built' – images that are classified by a MapSwipe user as having the target variable. In our case, this will be buildings.
2. 'bad imagery' – images classified by users as being obstructed (unable to determine if the image has the target variable or not, often due to satellite imagery with cloud cover).
3. 'empty' – images that are not classified with having the target variable.

The most important aspect about this dataset retrieved from MapSwipe is that the three different classes will be balanced. This will allow us to use accuracy as a metric for performance.

I also plan to utilize the DeepSat-6 satellite image dataset to extract the images' features and apply transfer learning by using the model weights when training and testing a new model with the MapSwipe images.

The DeepSat-6 dataset is 405,000 images, each 28 x 28px. The image labels are 'barren land', 'trees', 'grassland', 'roads', 'building', and 'water bodies'.


DeepSat-6 Satellite Image Samples (link)

Solution Statement

As mentioned above, I will use Keras with the TensorFlow backend to create a CNN model that can be applied on the MapSwipe images. A major part of this will be feature extraction and image augmentation to better generalize the model. I will explore the possibility of using transfer learning and the DeepSat-6 dataset to create a more robust model.
Predictions will be made on the test dataset of one of MapSwipe's Malawi projects.

Benchmark Model

Most of this project is exploratory to see how I can improve the CNN model for satellite image classification. For this reason, I will use a model that classifies each image randomly, producing a score of roughly 33% accuracy. This naïve score will be my benchmark to measure how my CNN model performs in comparison.

Evaluation Metrics

Since the dataset images for each class are balance, we can use accuracy as a metric for model performance. I will also use precision, recall and F1 scores to compare the different models' performances.

I will also use a loss function to validate each model in Keras. For this I will use categorical cross entropy loss to optimize the model on a validation dataset.

Project Design

I will separate the project into four primary components:
1. Data importation, cleaning and benchmarking.
2. Transfer learning and bottleneck feature extraction on DeepSat-6 dataset.
3. Feature extraction and data augmentation.
4. Training and testing on MapSwipe data.

The initial step of importing and preparing the data for exploratory data analysis and converting to tensors for Keras will take a bit of processing power as I will be manipulating more than ten thousand 256 x 256px images. I will need to convert the data into tensors.

Dataset will have this shape:

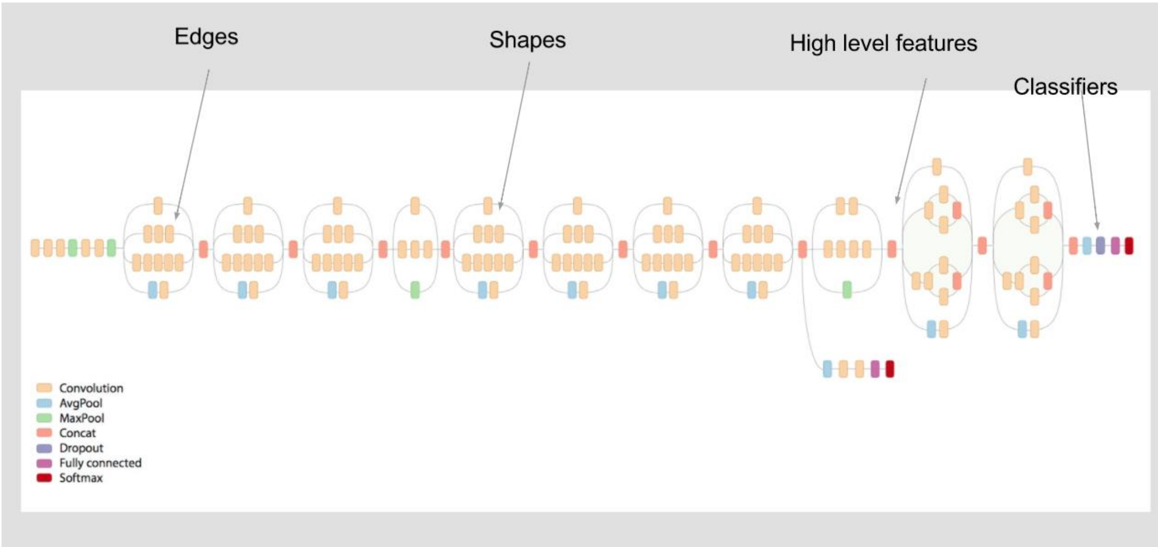
Number of Observations	Image Size (pixels)	Number of Layers
(9600,	256, 256,	3)

Observations within the dataset will have this shape:

Image Size (pixels)	Number of Layers
(256, 256,	3)

Once my data is cleaned and prepared, I will run a simple random forest model to produce a benchmark score for me to build upon. My metric will how accurate the model predicts the image labels.

From there, I will import and manipulate the DeepSat-6 data to pre-train a CNN with the data features. This will involve a great deal of additional research as the Udacity course discussed transfer learning with pre-train models on the ImageNet dataset, but did not go deep into pre-training a model on a new dataset to extract its features.



What the CNN layers are learning (link)

Independent of whether the DeepSat-6 model is improves our prediction score on the MapSwipe dataset, I will conduct several feature extraction techniques and compare the impact of these techniques on the accuracy score. The two primary methods I will explore are:
1. Edge detection.
2. Image augmentation.

I will build a few different models to train and test their performances. The key metric for this will, again, be the accuracy in which the model predicts image labels and the categorical cross entropy loss. This will take several iterations and will likely be the most time-consuming step of the project. I will be performing most of the work on an EC2 instance with one GPU and four CPUs, which will help to move training along.