CAPSTONE PROJECT #2

MILESTONE REPORT 1

Author: Jerome Gonzaga

Problem Statement

1. **What is the problem you want to solve?**

    a. A current issue that frequently arises for my company is that the turn-around time for reviewing construction drawings is very long and the process is tedious. Much of the review time is spent on determining whether or not a particular construction symbol shows up on certain construction plans, whether a building code requirement is met and drawn on the plans, and determining if the construction team addressed all the project requirements. Even just one task of determining where on the drawings all the symbols are associated for a particular building component would save a lot of time, effort and money for my company.

2. **Who is your client and why do they care about this problem? In other words, what will your client do or decide based on your analysis that they wouldn't have done otherwise?**

    a. If my company could create some sort of image recognition model to detect where on a drawing a particular component is, then this model could be utilized to tackle many of the tasks associated with finding certain building components on a construction plan. With this knowledge of knowing if or where particular building components are on a construction drawing, then this knowledge would decrease the amount of review time drastically! This is a very daunting task that not many online have been able to tackle. So, to help make one step closer to achieving this goal, this project will consist of creating a model to detect particular objects within pictures, whether it is a building, a car, a person, or even something like a light pole.

3. **What data are you using? How will you acquire the data?**

    a. The images to be used to train the model and validate the model is a Kaggle dataset in the following link:
        i. https://www.kaggle.com/puneet6060/intel-image-classification

4. **Briefly outline how you'll solve this problem. Your approach may change later, but this is a good first step to get you thinking about a method and solution.**

    a. The first step in this object detection project is to ensure that the images can be classified appropriately based on a particular category. Once the deep learning model can successfully classify the image into a category, then the next step will be to train the model to detect a particular object, for example to detect a person or a traffic signal within the pictures. Lastly as a trial, I will try to use transfer learning or additional training to try and detect a particular building component on the construction plans.

5. **What are your deliverables? Typically, this includes code, a paper, or a slide deck.**
    a. The deliverables for this project will eventually be a jupyter notebook outlining the steps that I take to effectively classify images and to detect objects and another deliverable will be a slide deck to present my results in a concise way.

Data Wrangling

The original dataset from Kaggle includes a set of training images, testing images and images that are not pre-labeled. The training images and testing images are pre-labeled and sorted into separate folders according to the following classifications:

1. Buildings
2. Forest
3. Glacier
4. Mountain
5. Sea
6. Street

The total number of images within the training, testing and “unlabeled” subsets are as follows:

Original Dataset

Training Images	Testing Images	Unlabeled Images
14034	3000	7301

The majority of the images within each subset are colored images with the image size of 150px by 150px. All of the images were reduced down to 75px by 75px, if possible, for the purposes of passing the images through the deep learning model. However, there were some images from each of the subsets that were not originally 150px by 150px - these images were removed from the dataset. So, the final number of images within each subset that were used for this project were as follows:

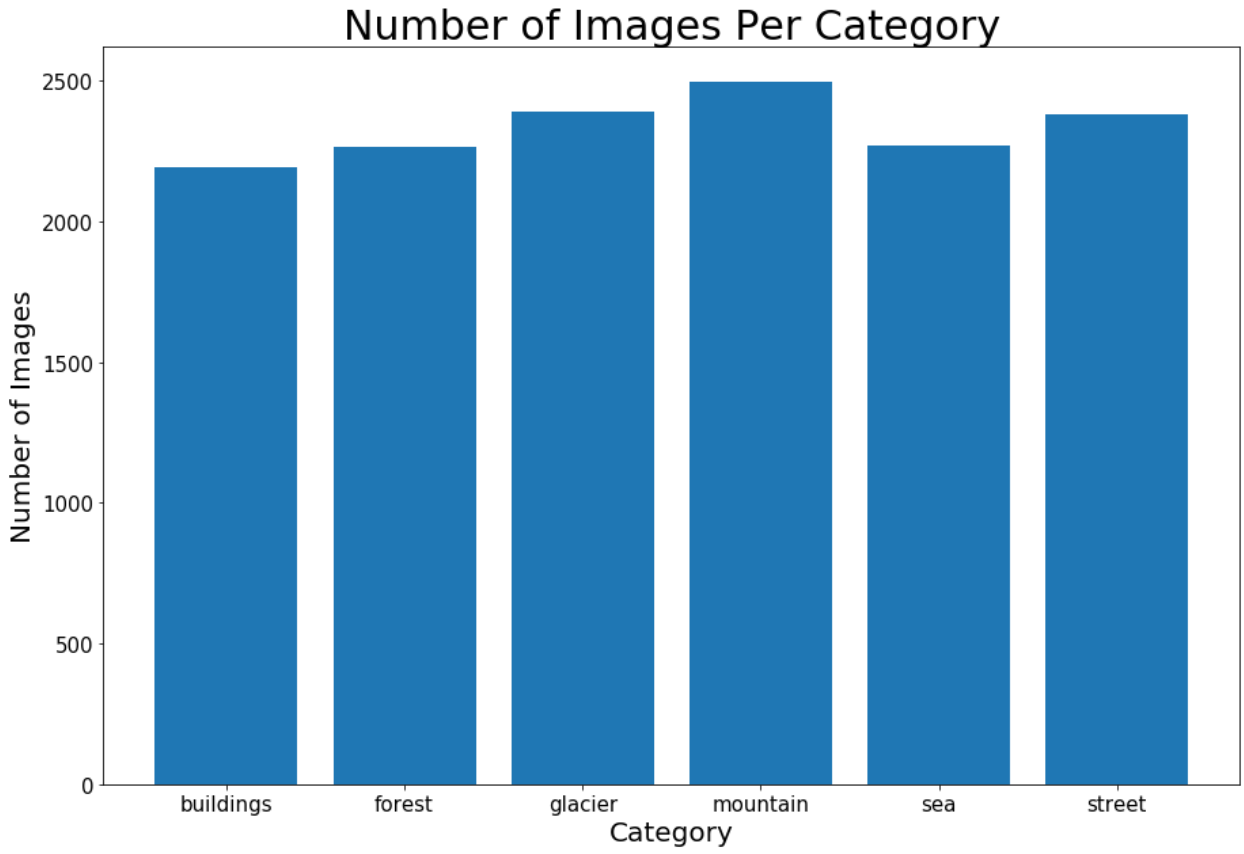
Final Dataset

Training Images	Testing Images	Unlabeled Images
13986	2993	7288

Analysis of the Dataset

One consideration that was looked at for the final dataset was to determine if the deep learning model had an equal number of images for each category to train off of. If there is an uneven number of images for a specific category, then upsampling or downsampling may need to be considered in order to “level the playing field” for each of the categories.

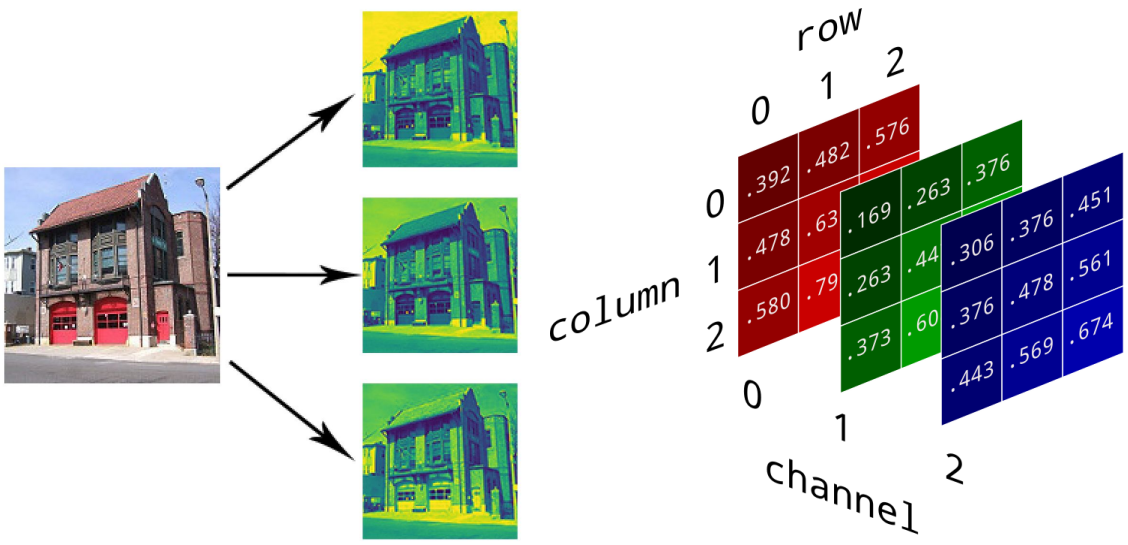
An initial look at the final dataset for each of the categories shows that there is at least over 2000 images for each category.



Since the number of images for each category are between 2000 and 2500, upsizing or downsizing was not performed for this project. However, if the accuracy scores produced by the deep learning model are not as high as expected or desired, then some technique of upsizing or downsizing will be performed to hopefully increase the value of the deep learning models accuracy score.

Splitting Images Into Separate Channels

An additional step taken to clean up this final dataset used for this project, each image was split into three images based on the RGB values within the image.



Splitting each of the images into three separate channels (red, green, blue) was done in order to potentially increase the accuracy of the deep learning model.

There are some techniques performed on image datasets to normalize each of the arrays that are created for each channel but this project does not employ that technique at this moment in time. Normalizing the arrays for each channel may be explored later if classification accuracy scores for the deep learning model are below expected values.