Capstone Project 2 | Milestone Report 1
Claire Miles

Project Subject: Land Use/Land Cover Classification

Problem Statement and Dataset Description:
Satellite imagery can help us find solutions to the growing number of environmental problems that we face today. It allows us to not only get a bird's eye view of what's around us, but also uncovers parts of the world that are rarely seen. Tapping into the potential of categorizing land cover and land use around the world means that humans can more efficiently make use of natural resources, hopefully lowering cases of waste and deprivation. But despite its potential to be incredibly useful, satellite data is massive and confusing, and making sense of it requires complex analysis.

This project could help a variety of stakeholders, including conservationists, urban planners, and environmental scientists, survey and identify patterns in land use to see which natural areas are under threat or which areas are best for urban development. But in this case, I will tailor this project to a client that is similar to the agriculture imagery company that I currently work for. Imagery companies can use land use classification models to categorize what's in each image and optimize their efforts towards the parts of land that are important to them. For example, an agriculture company will only want to be concerned with land that's labelled as pasture, annual crop, or permanent crop. If a satellite is constantly taking images, this project would help save hours of time manually sorting through imagery.

In addition to sorting imagery, land use classification is important for identifying the parts of an image to which certain analyses are applied. For example, if the company has different crop stress algorithms for fields that are next to urban areas versus fields that are next to rivers, this project would help them to automate the process of applying these specialized algorithms and more effectively solve environmental problems.

For this project, I'm using open source EuroSAT Sentinel-2 satellite images from the German Research Center for Artificial Intelligence, which can be downloaded locally [link]. The dataset consists of 27,000 labeled images of 10 different land use classes:

1. Annual Crop
2. Forest
3. Herbaceous Vegetation
4. Highway
5. Industrial
6. Pasture
7. Permanent Crop
8. Residential
9. River
10. Sea / Lake

Each multispectral image consists of 13 different color bands that represent different wavelengths of light/color and different resolutions. These different light bands help distinguish parts of the landscape that reflect certain types of light in particular ways. Since most images don't include special bands like Vegetation Red Edge, Coastal aerosol, or SWIR, for this project I chose to only use the red, green, and blue bands in an effort to make my model generalizable to most images.

Sentinel-2 Bands	Central Wavelength (µm)	Resolution (m)
Band 1 - Coastal aerosol	0.443	60
Band 2 - Blue	0.490	10
Band 3 - Green	0.560	10
Band 4 - Red	0.665	10
Band 5 - Vegetation Red Edge	0.705	20
Band 6 - Vegetation Red Edge	0.740	20
Band 7 - Vegetation Red Edge	0.783	20
Band 8 - NIR	0.842	10
Band 8A - Vegetation Red Edge	0.865	20
Band 9 - Water vapour	0.945	60
Band 10 - SWIR - Cirrus	1.375	60
Band 11 - SWIR	1.610	20
Band 12 - SWIR	2.190	20

Sentinel-2 Bands, Wavelength, and Resolution

(Image Source)

Data preparation

Creating Folders

In order to be compatible with the keras function *flow_from_directory*, I created train, validation, and test set folders for the data. Within each folder, the data was separated into more folders by category.

Translating File Type

Compatibility with keras/tensorflow also required me to translate the files from tif to jpeg format. I found that this was easiest using the *gdal_translate* function in the command line, which I accessed within the notebook using the *!* notation.

When translating, I selected bands 2, 3, and 4, which are the red, green, and blue bands. Choosing the RGB bands makes for a traditional image, rather than a specialized satellite image which may have extra near-infrared, red-edge, or short-wave infrared bands. This makes the model more accessible to the average, everyday image of the outdoors.

After translating a file, I moved it to the training data folder. When all files of a certain category were trained and moved, I separated the jpeg files into the validation and test data folders with a train:validation:test separation of 80:10:10.

Exploratory Data Analysis & Findings
Before building the deep learning model, it's worth it to get more familiar with the dataset. Especially as an image dataset, visualizations help us see differences in the different image categories as well as the red, green, and blue bands of a single image.

Photos by Category
The bar chart shows the number of photos in each category in the dataset, which range from about 2500 photos in the Industrial and River categories to about 6000 in the AnnualCrop, SeaLake, HerbaceousVegetation, and Residential Categories (Figure 1). Imbalances in representation could affect the efficacy of the model, but we will leave the dataset as is for now.

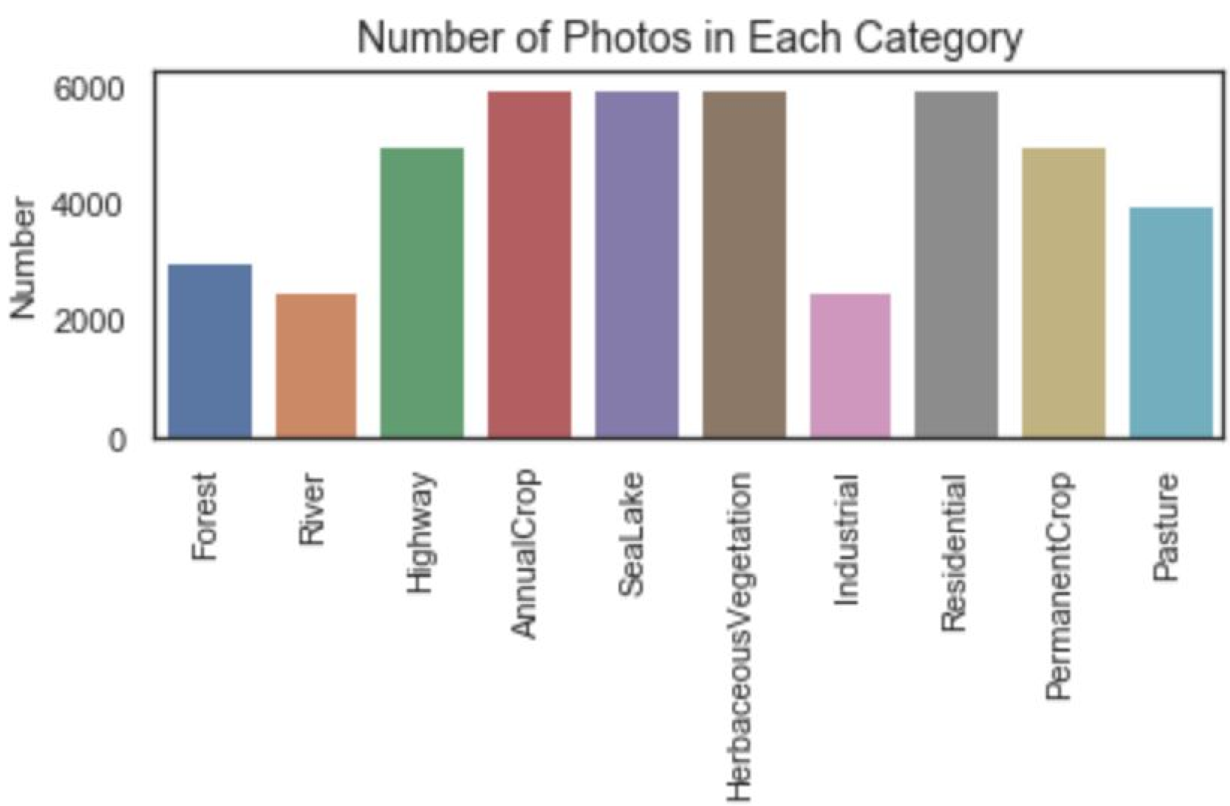

Figure 1. Bar plot of the number of images in each category of the dataset.

The RGB representation of a photo from each category provides a better sense of what the images actually look like (Figure 2). Though they are relatively low quality, it's still pretty easy to see the difference between what is denoted as cropland, residential, highway, etc.

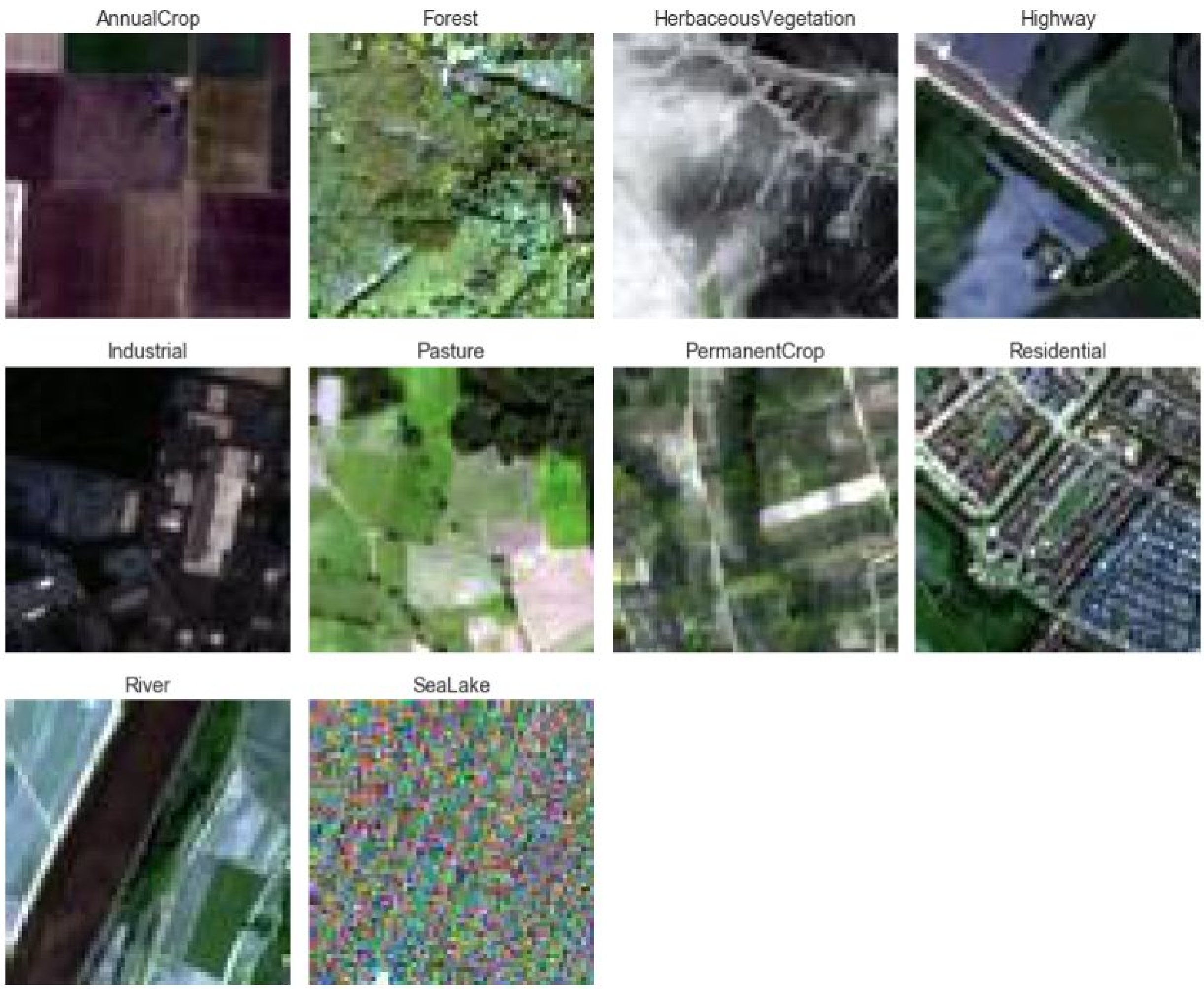

Figure 2. One image from each category in the dataset.

Photos by Color Band

The three color bands within each photo can also provide more information about the satellite data. The package rasterio, has helpful tools to extract information about the images that will be useful when constructing the convolutional neural network.

For example, the height and width of the images is 64 pixels, and there are 3 bands in each image. This makes sense, since we extracted three bands in our translation of the images from tif to jpeg. The RGB bands can be corroborated by the realistic look of the sample images from each category.

Plotting each band doesn't tell us too much about the makeup of the data, but it is proof that there are three different bands in each image (Figure 3). However, stacking the three bands results in a RGB composite photo with realistic looking colors (Figure 4). In order to create this stack image, the bands have to be normalized first.

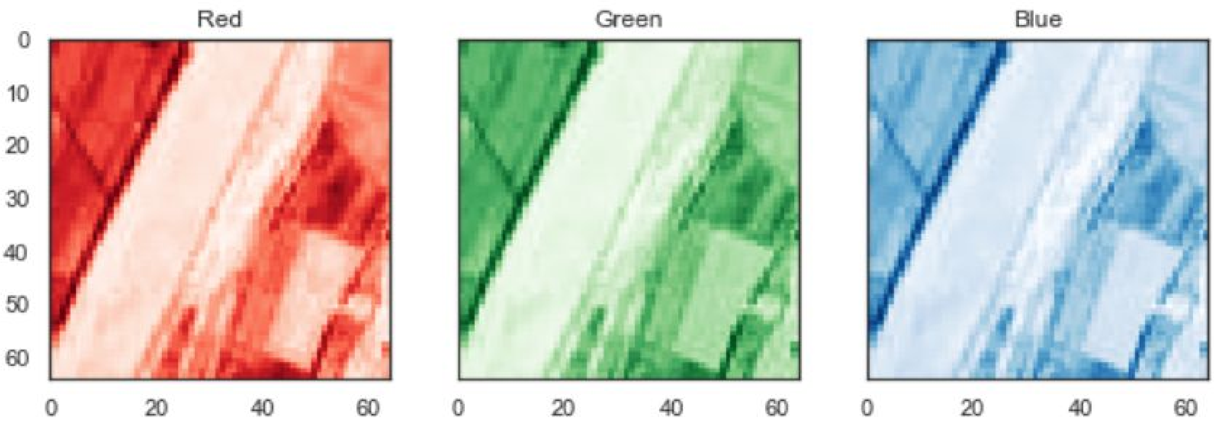

Figure 3. The red, green, and blue bands of a single image in the dataset.

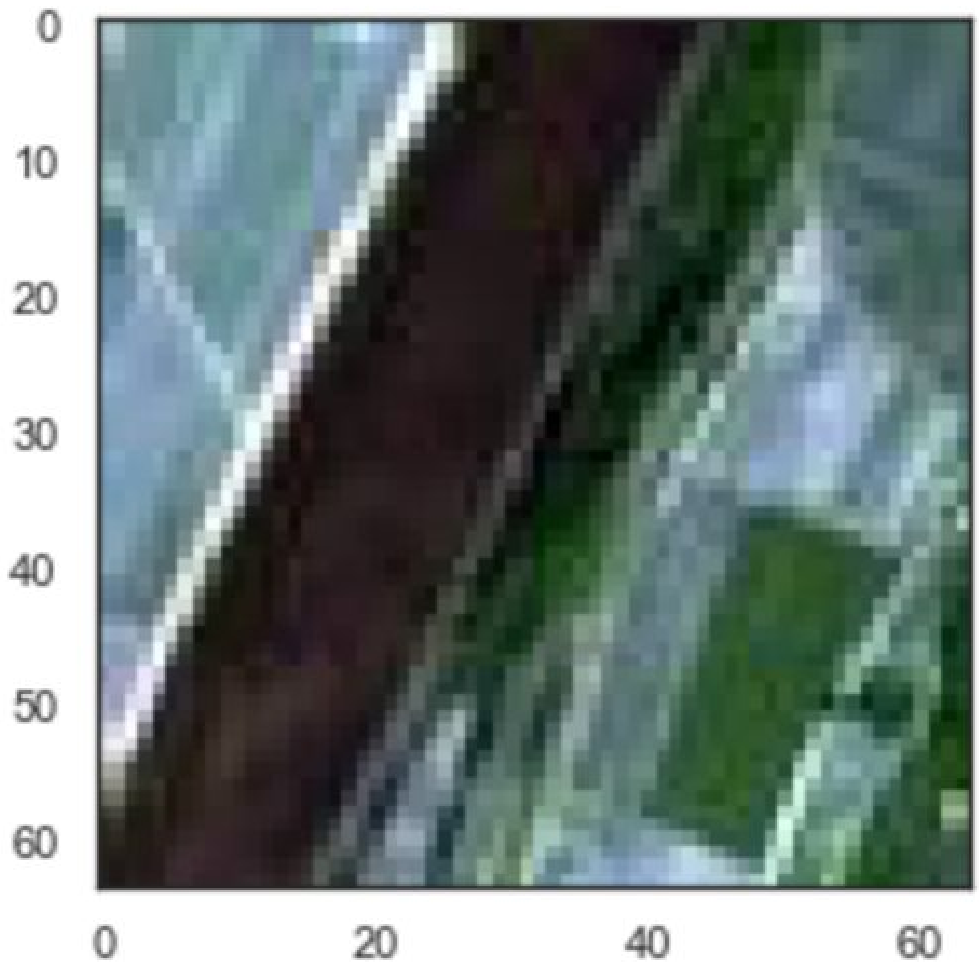

Figure 4. The same image when the red, green, and blue bands are stacked, creating a realistic look.

Plotting the wave frequencies of each band shows that there are differences in the pixel representation for each one (Figure 5). These differences could come in handy as the neural network learns the differences between image features in each category.

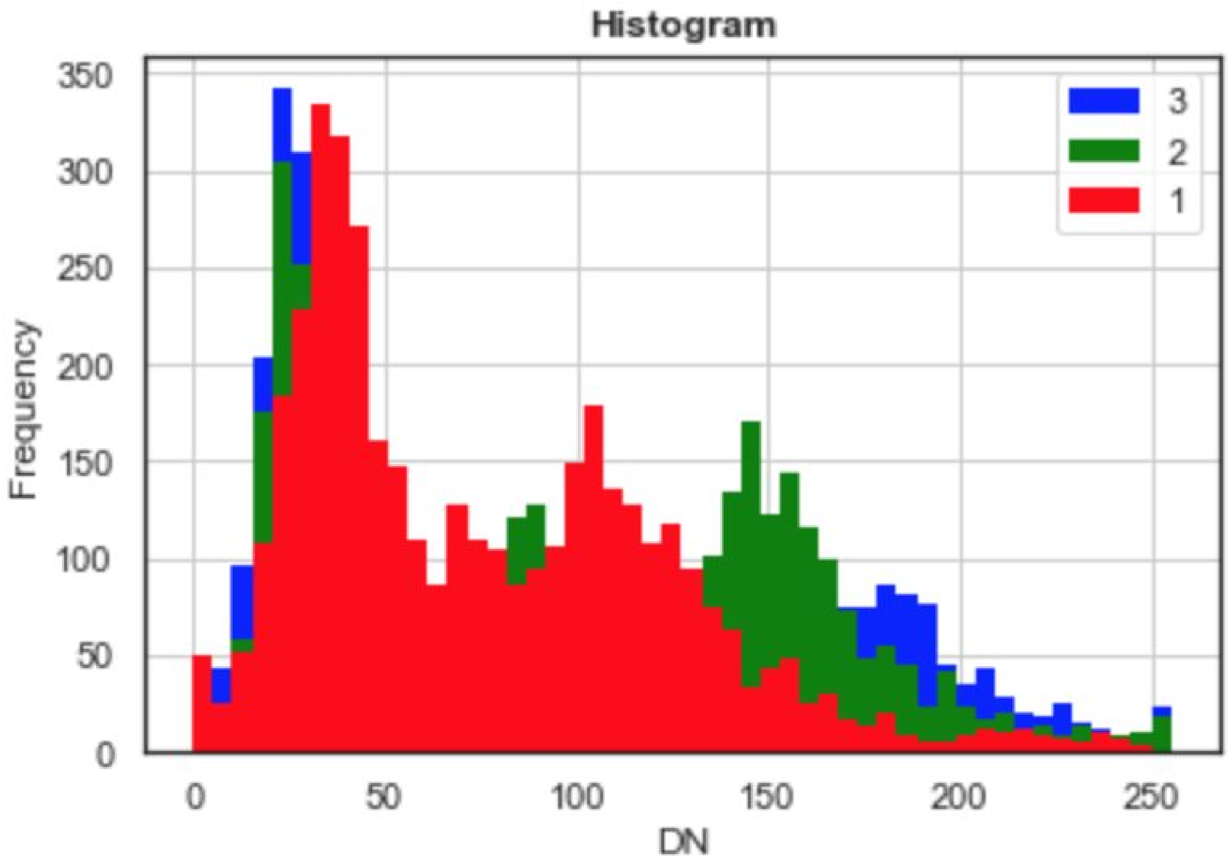

Figure 5. Histograms of the wave frequencies of the red, green and blue color bands of all one image in the dataset.