Alex Brockman

CSC 391: Introduction to Computer Vision

4/27/19

Project #4: Classification and Testing

Part 1: Classification

Project #4 is an extension of Project #3. In Project #3, I created a small data set of images with rivers and images without rivers. I then applied a combination of the HOG feature extraction algorithm and the ORB/SIFT keypoint detector algorithm to distinguish images with rivers from those without. I then displayed my results of the combination of the algorithms.

For Project #4, I created a much larger data set of images containing rivers and images without rivers. My dataset of images contained over 4000 images without rivers and 350 with rivers. I created the dataset by running a small script called image_split.py that split up the large images I used in Project #3 into smaller 128 x128 pixel and 256x256 pixel images. For classification, I used different feature descriptors called Hu Moments, Haralick Texture and Color Histogram. I determined that these algorithms would be more effective than my method from Project #3. Haralick Texture accounts for texture similarities in an image, Color Histogram accounts for color characteristics in images and Hu Moments accounts for shapes within images.

Part 2: Testing and Results

My initial goal of this project was to not only be able to identify whether rivers existed in a chosen image but also to be able to identify where a river existed in an image. I have since decided that simply identifying whether an image contained a river or not was sufficient for my project. Most of the smaller test images feature rivers that spans across the entire image.

Therefore, I utilized machine learning techniques to train a model to identify whether a test image contained a river. My model first compared 7 different machine learning models. They included Logistic Regression, Linear Discriminant Analysis, KNeighbors Classifier, Decision

Tree Classifier, Random Forest Classifier, GaussianNB and SVM. I then compared each model against my training data to determine the most effective technique to use against my data.

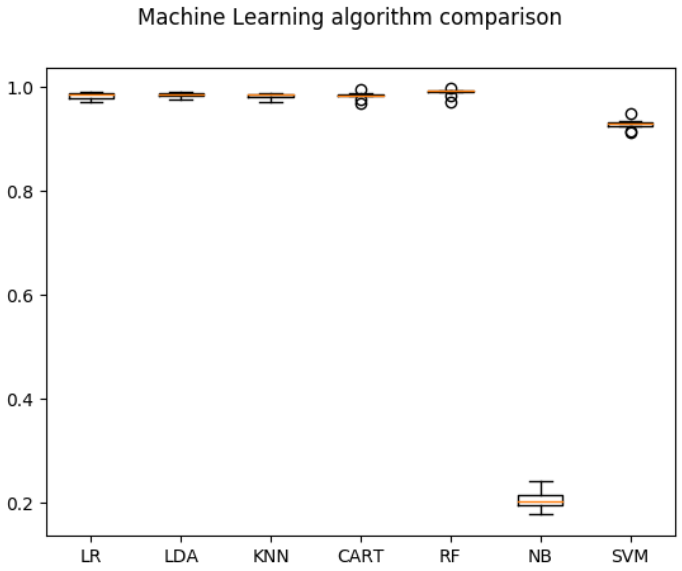
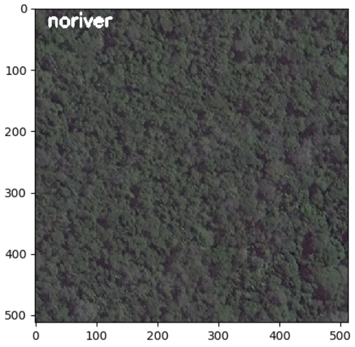
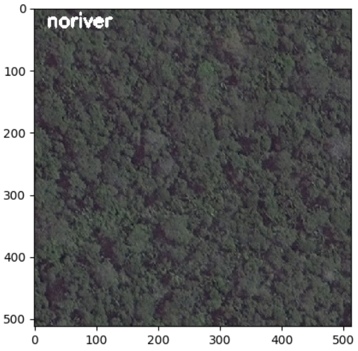


Figure 1: A comparison of the machine learning algorithms

Every algorithm scored above a 90% accuracy except for the GaussianNB algorithm that scored around 20% accuracy. My model then displayed the test images with a "river" or "noriver" label in the top left corner. Here are some of the results I received from images not containing rivers:





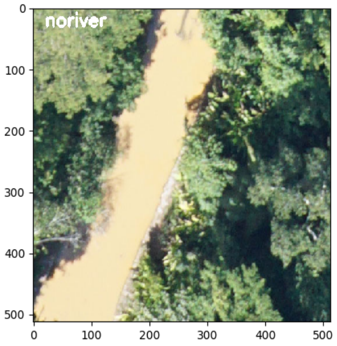


Figure 2: Images predicted without rivers

It is clear that 4 out of 5 of these images did not contain any rivers and my model was able to identify correctly. However, there is clearly an image that contains a river that was labeled incorrectly. A machine learning algorithm is only as accurate as the amount of data in your dataset. Luckily, my image recognition problem is not as complex as it could be because images with rivers are generally easy for a computer to identify. Here are my results for images that contained rivers:

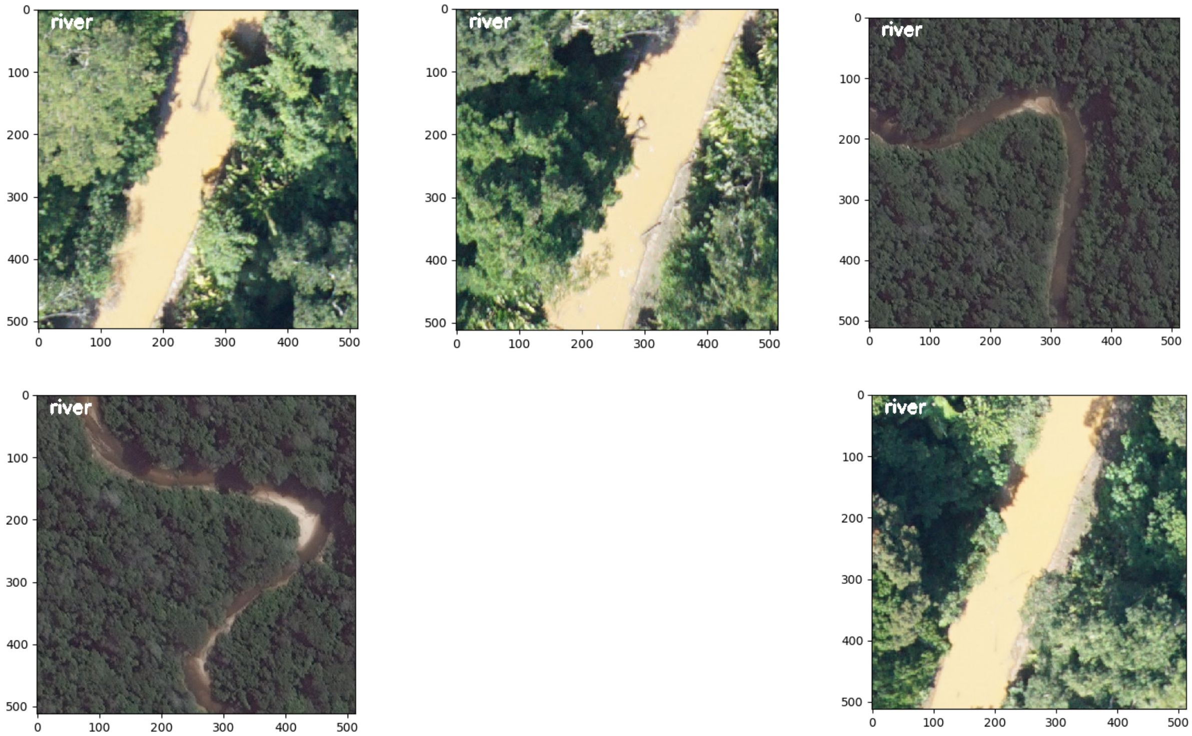


Figure 3: Images predicted with rivers

Each one of these images was predicted accurately. It is possible that it is easier for the program to identify that a river exists versus when it does not. I learned a lot about machine learning and classification from this project and it would be possible to add more features to my program that could improve its accuracy and speed. I believe this type of analysis has the potential to be very useful for farmers or other specialists that might want to analyze drone images for a specific purpose.