

Quinn Calder

B Period

Ms. Despres

8 February 2016

Counting Saguaro Cacti With Google's Tensorflow

Abstract

This engineering project's goal was to create a program using Google's Tensorflow to count saguaro cacti from satellite images of Saguaro National Park. This is meant to alleviate manual counting of the large cacti when trying to get population estimates. This involved getting samples of satellite images of areas of the park (called scraping). This was done with a python program. These samples were then run through a second program where I manually labeled the cacti. I only did this to 350 of the 100,000 samples that could be scraped from Saguaro National Park. Then the sample was split in 19 by 19 overlapping images that were all saved separately with names depending whether they had a cactus (True) or not (False). Those images were then set up and inputted into Tensorflow. With the images now set in Tensorflow it can now start making models, which is the computer program that can detect cactuses in an image. This model is improved by running the images through it multiple times, it can only get better the more you run through it. Then the model is put to the test and given samples to label. In the end it was proven that you can write a program with Google's Tensorflow that can be better at estimating the saguaro cactus population then a human. A human that looked at 250 samples and counted the cacti in the images and then used those results to estimate a population had an error rate of around 14%, the machine could match this by taking 1000 samples which it can do easily. Someone who looked at 500 images would have an error rate of around 10%, the computer can match it at 2000 samples. And if someone looked at 1000 samples they would have an error on the total population of about 6%, the computer could match that at 4000 samples. The computer can match any population estimate error rate by looking at four times the number of samples.

Introduction

Saguaro cacti (*carnegiea gigantea*) are a large species of cactus that grow in the Sonoran Desert in the Mid West (National Park). They can reach up to 40 feet tall and live up to 200 years. They are important in the ecosystem because they provide not only shelter to birds and small mammals ,but they also provide a good food source with their fruit (fs.fed.us). The fruit

and seeds are an important food source for many birds and mammals (fs.fed.us), and bats drink their nectar and pollinate the large cacti (National Park). Birds nest on the cacti and woodpeckers dig inside the cactus and nest there. If the saguaro cacti were to die off it could have a large negative impact on the ecosystem (fs.fed.us). So it is important to count the population of the cacti so people can fix any problems with the population. This will also help identify problems in the area by being able to track the saguaro population and compare it to different events in the region.

The images taken to identify cacti in are taken off of Google Maps. Google Maps is a free map service provided by Google that can look at landscapes from satellites. Therefore Google Maps can be used to get images of the Sonoran Desert with ease. To make it even easier Google Maps offers a selection of API's (application program interface) that can do different things on the maps, such as showing elevation or distance. The API used in this project was a web API that took static (still) images and allows those images to be saved. It is important to be able to save the images because Tensorflow (See below) must have the images read into it from the computer, it cannot read in from Google Maps.

Saguaro cacti are the perfect subject to be counted with satellite images for many reasons. They are large plants that leave a fairly identifiable shadow (a long stick like shape that sometimes has arms). This allows the program to have a specific thing to identify as a saguaro cactus. If there was no defining feature to differ it from the surrounding landscape then it would be nearly impossible to identify from a bird's eye view perspective. They inhabit deserts which do not have many things to block the images taken, except for ground foliage which can look

like the shadows, but not enough to cause a serious problem. And deserts have a low cloud cover so it is very easy to see the cacti from satellites.

The code that was used to count the cacti is called Tensorflow. Tensorflow is an open source library that uses mathematical operations to get to do numeric computation (Tensorflow). This will be used to tell a computer what to look for to identify a saguaro cacti. The code will “look” at satellite images taken with Google Maps Static API and count the number of cacti in said image. It will be able to count the shadows the cacti cast and in turn count the cacti. The more samples taken the more accurately the machine can identify the cacti because it will gather more info on what the shadow looks like. All the images will be taken from Saguaro National Park in Arizona.

The program will only be able to identify saguaro cacti because no other flora has such an identifying feature. Trees come close but the only way to truly identify the tree is to look at the leaves or trunk, but the satellite images cannot focus well enough to get a good enough image to do such identification methods.

Methodology

This project consisted of three python programs. They were called `saguaro.py`, `saguaro_inputs.py`, and `saguaro_tf.py`. The main program that had a large chunk of the code and did many jobs was `saguaro.py`. This included scraping the images, setting up images to be labeled, saving the labeled sub-images, building models, displaying examples of computer labeled images, and doing math to get the results. It included 10 python libraries (which are groups of tools to be used in programming) to complete its tasks. these libraries were `argparse`,

cv2, requests, random, os, numpy, Tensorflow, math, and it included saguaro_tf and saguaro_input so it could use functions (tools) from those two programs. The program saguaro_inputs.py takes in the sub-images and turns them into one-dimensional arrays that are then stored into a huge matrix of those arrays. This makes it so the computer does not have to read the images from the computer's disc every time it wants to build a model making the process faster.

The final program is saguaro_tf.py. This is the program that holds the Tensorflow code and is in charge of building models. There are two ways it can build the model, through the simple or complex model programs. The simple model program is the first model Tensorflow's tutorials teach you how to build and therefore not the better one. It can still make models but they tend to be less accurate than the complex model or any other Tensorflow teaches. There is an upside to using the simple model, it is much faster than the complex model. The complex model which is what they use at Google for machine learning is much more accurate than the simple model, but also takes much longer to build a model for. Other than those two model building programs Tensorflow is a black box. Google made Tensorflow to allow everyday people like me to use machine learning for whatever they wanted. They made it so you don't need to know the math behind the code to make your computer learn with their amazing system. Which is why it is so great and my choice for the machine learning code.

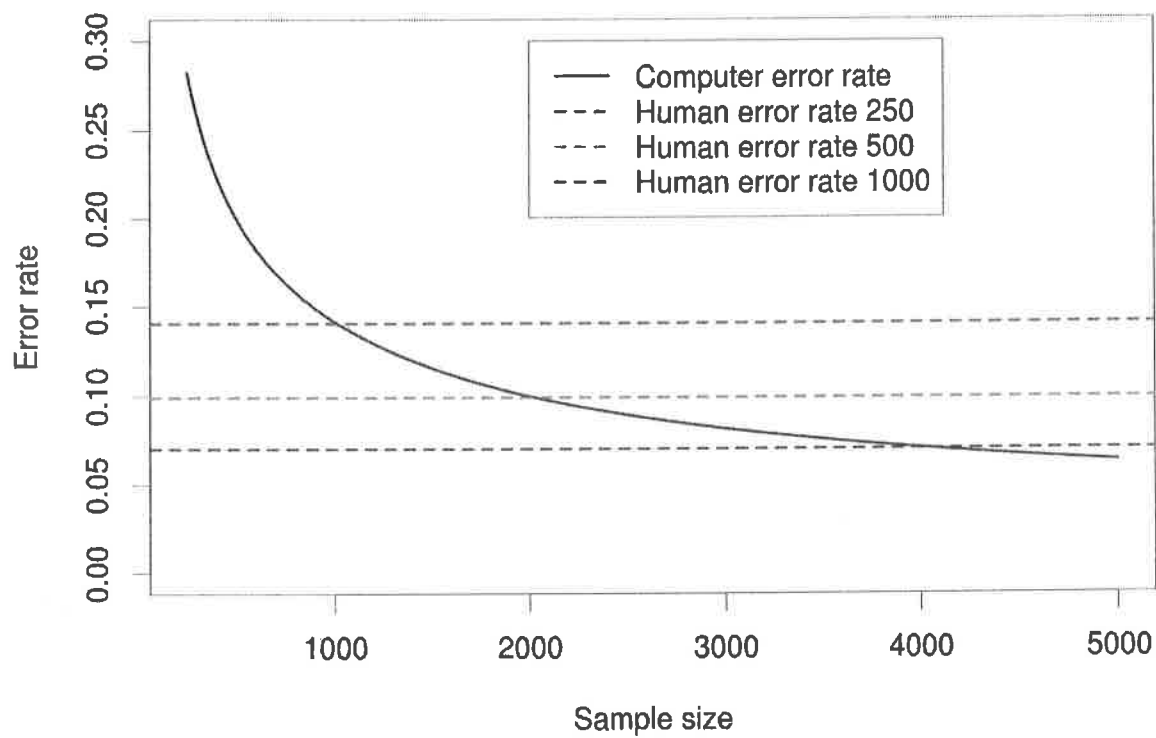


Figure 1

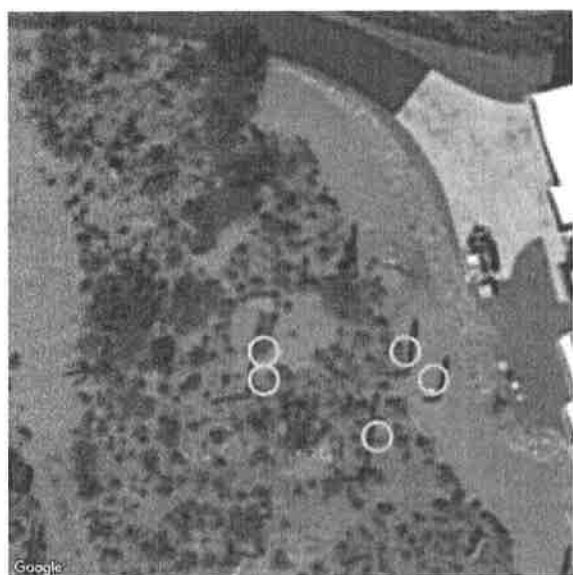
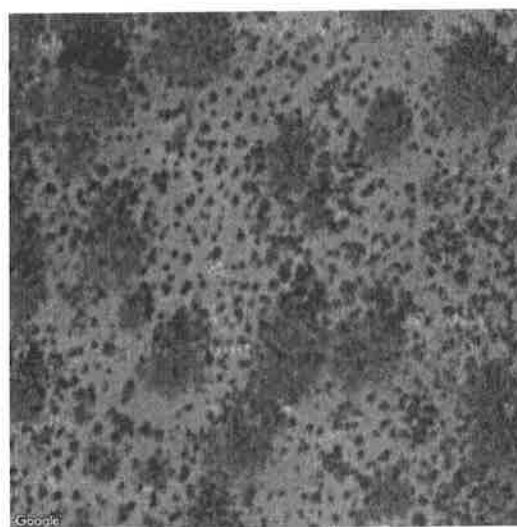


Figure 2

Figure 3



Results

Figure 1 is a graph showing the results of the Tensorflow program's model that was built. The straight colored lines represent a human's error rate, each line represents a different number of samples they looked at. The curved black line represents the error rate of the computer depending on how many samples the computer got. The y-axis represents the error rate percentage and the x-axis is the sample size. The computer manages to catch up to the human no matter how many samples the human looks at. Figure 2 and 3 are computer labeled samples. Figure 2 shows a sample with cacti that the computer found and labelled. The computer is not perfect though so it doubled labeled a cactus and missed one completely. Figure 2 is an image that had no cacti and the computer recognized this and did not label any cacti.

Conclusion

The computer built model was able to beat the humans error no matter how many samples the human looked at. The computer would match the error rate by taking four times the amount of samples. The error rate was determined by seeing how close the estimated population was to the actual population. This was measured in percent of the estimated population. I took 250 samples. I then found the average number of cacti in each sample (around 3.5 cacti each). I then multiplied that finding by the number of samples in the area of the park I chose (there are around 107,000 samples) and got my total. My error rate was around 14%. And as I said at the beginning, the computer as able to match my error rate by taking around 1000 samples. As my number of samples increase and my error rate decreases, the computer is still able to match my error rate and get better. The project was a success.

The codes written are very simple and therefore not the best out there. People with much more experience with python or with Tensorflow could probably make an even more accurate computer model. But, the code does have a very nice program that allows the human to look at a sample and click on the cacti in the image so that they are saved separately and input into Tensorflow astrue images so the computer knows what to look for.

The project as a whole would be very useful to conservation efforts that wish to track the population of the saguaro cactus. Currently people must go out to the desert and count the cacti by hand. But, with this code it may take the trouble out of the estimates and gets rid of the need for manpower. If this code were to be edited by people who code and work with machine learning more than I do, it may be able to identify more than just saguaro cacti. The possibilities are endless.

Citations

Rosebrock, Adrian. "Capturing Mouse Click Events with Python and OpenCV - PyImageSearch."

PyImageSearch. N.p., 09 Mar. 2015. Web. 08 Feb. 2016.

<http://www.pyimagesearch.com/2015/03/09/capturing-mouse-click-events-with-python-and-opencv/>

Calder, Matt. Personal Interview. 08 February. 2016

"Image Thresholding." *OpenCV*. OpenCV, n.d. Web. 8 Feb. 2016.

http://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html#gsc.tab=0

"TensorFlow Is an Open Source Software Library for Machine Intelligence." *TensorFlow*.

Google,

n.d. Web. 08 Feb. 2016.

<https://www.tensorflow.org/>

"Stack Overflow." *Stack Overflow*. N.p., n.d. Web. 8 Feb. 2016.

<http://stackoverflow.com/>

"Carnegiea Gigantea." *Carnegiea Gigantea*. N.p., n.d. Web. 30 Nov. 2015.

<http://www.fs.fed.us/database/feis/plants/cactus/cargig/all.html>

"TensorFlow Is an Open Source Software Library for Machine Intelligence." *Tensor Flow*.

Google, n.d. Web. 30 Nov. 2015.

<https://www.tensorflow.org/>

United States. National Park Service. "Saguaro Cactus." *National Parks Service*.

U.S. Department of the Interior, 28 Nov. 2015. Web. 30 Nov. 2015.

<http://www.nps.gov/orpi/learn/nature/saguaro-cactus.htm>

Introduction

This project was meant to test whether you could make a computer program that would be better at estimating the population of saguaro cacti from a selected area in Saguaro National Park in Arizona. The program I used is called Tensorflow and is Google's machine learning program that they use for their various apps. The cactus that I counted is called a saguaro cactus, *carnegiea gigantea*. They are a large breed of cactus that grow in Sonoran Desert in the Mid-West. They are an important part of the ecosystem, providing both habitat and food for animals. The saguaro cacti are the perfect choice for counting because it grows in the desert where there are not many trees to obstruct the view for the satellite. This is important because the images that the cacti are counted from are gotten with a Google Maps API that allowed me to scrape or gather satellite images of the area I selected in Saguaro National Park.

Conclusion

The computer built model was able to beat the humans error no matter how many samples the human looked at. The computer would match the error rate by taking four times the amount of samples. The error rate was determined by seeing how close the estimated population was to the actual population. This was measured in percent of the estimated population. I took 250 samples. I then found the average number of cacti in each sample (around 3.5 cacti each). I then multiplied that finding by the number of samples in the area of the park I chose (there are around 107,000 samples) and got my total. My error rate was around 14%. And as I said at the beginning, the computer as able to match my error rate by taking around 1000 samples. As my number of samples increase and my error rate decreases, the computer is still able to match my error rate and get better. The project was a success.

Procedure

The first step to setting up the computer to counting cacti was to get samples of the area selected in Saguaro National Park. This involved using a Google Maps API that allowed me to download a selected point on the maps. I set up a program to do this manually and randomly. Then I looked at the images and with a second program I wrote I labeled the cacti in the sample. The sample was then broken up into 361 sub-images, the images that I had selected as cacti were saved as cacti and the ones that weren't cacti were saved not as cacti. Those sub-images were then turned to grayscale then black and white. And then they were shrunk down to half their original size to make the program faster. Then to make it so Tensorflow will accept the images, they were turned into one-dimensional arrays. They were then given to the computer so it can construct a model. The model is the part of the computer that counts the cacti, it is the brain. Then with my counts and the computers counts I was able to calculate the error rates. My error rate was 14% and the computer can easily pass that by taking 1000 samples. The computer is now better at estimating the population of saguaro cacti in Saguaro National Park than a human.

Citations

Rosebrock, Adrian. "Capturing Mouse Click Events with Python and OpenCV - PyImageSearch."

PyImageSearch. N.p., 09 Mar. 2015. Web. 08 Feb. 2016.

<http://www.pyimagesearch.com/2015/03/09/capturing-mouse-click-events-with-python-and-opencv/>

Calder, Matt. Personal Interview. 08 February. 2016

"Image Thresholding." *OpenCV*. OpenCV, n.d. Web. 8 Feb. 2016.

http://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html#gsc.tab=0

"TensorFlow Is an Open Source Software Library for Machine Intelligence." *TensorFlow*. Google,

n.d. Web. 08 Feb. 2016.

<https://www.tensorflow.org/>

"Stack Overflow." *Stack Overflow*. N.p., n.d. Web. 8 Feb. 2016.

<http://stackoverflow.com/>

"Carnegiea Gigantea." *Carnegiea Gigantea*. N.p., n.d. Web. 30 Nov. 2015.

<http://www.fs.fed.us/database/feis/plants/cactus/cargig/all.html>

"TensorFlow Is an Open Source Software Library for Machine Intelligence." *Tensor Flow*. Google, n.d. Web. 30 Nov. 2015.

<https://www.tensorflow.org/>

United States. National Park Service. "Saguaro Cactus." *National Parks Service*.

U.S. Department of the Interior, 28 Nov. 2015. Web. 30 Nov. 2015.





