Franklin Lantigua
4/18/2021
Principles of Data science
Final Report

Introduction

Philadelphia has a staggering litter problem that seems to have no light at the end of the tunnel. It was temporarily in a better state late August last year because of lockdown but the problem arose once again as people began littering masks all over the streets and parking lots.My goal is to help build a classifier that focuses primarily in detecting garbage bins using Google Maps Street View as the primary source of images. While I have also taken photographs of garbage cans across center city I made sure to construct my dataset so that it consists of mostly Google Street View garbage bins.

This project by Sharon Morris [3] sought out to develop an SVM classifier to determine whether an image contained a bumble bee or a honey bee. He used various images of bees parked on top of flowers and fed them into his classifier for training and testing. Google is well known for their captchas to deter bots from attacking web pages but the captchas they use typically revolve around harvesting data for use in their own image classifiers. You'll find that they occasionally ask users to identify images with fire hydrants or school busses. It's a clever way of shooting two birds with one stone when generating data.

Due to time restraints I wanted to set an achievable goal when it came to developing this classifier. Firstly, I wanted to make sure that I can reasonably obtain a quality image set for the project that can output a degree of accuracy worth considering. As there is no repository of images revolving around garbage bins of this nature (likely due to Google possessing ownership of the images), my intent was to focus on collecting data that is suitable for an SVM classifier. Even when collecting data, I also made sure to focus on the garbage bin receptacles shape and not the function. This meant that I was going to generally classify garbage, recycling, and cigarette/dog bins as one singular "garbage bin" entity.

Approach

As described earlier, the intent was to manually acquire images from Google Street View to use for the SVM classifier. Below are some examples of the images I recorded for the dataset:

Source:Google Maps Street View on Rising Sun Avenue

Users of Google Street View would know that this isn't the entire image but a small portion of an image. I cropped out only the object in question which allowed me to determine the feature description of the image using a *Feature Descriptor*. A "feature descriptor" is an algorithm that extracts information from an image and converts the image into a vector of numbers that describe what the image contains within. One kind of feature descriptor is known as a Histogram of Oriented Gradients(HOG) and it describes an object based on their general overall shape by detecting contours and edges of the subjects in the image. By using HOG we would be telling our classifier to identify cylindrical or, in the case of smart bins, boxy figures within a single image. Due to the project scope, the most important feature to look for in a garbage bin was its shape and because we only care about the shape it would be reasonable to simply grayscale the trash bins and extract the HOG features that way. This serves two functions:

1. It conserves processing power

2. It also decreases the amount of information generated since we're now focusing on one single color channel instead of 3.

After converting the image into grayscale we can now run a HOG library that will generate a histogram defining the features of the garbage bin in the image. From here I had my program loop through and repeat this process over for the remainder of the image set. With the feature description recorded, it was trivial to train and test our model.

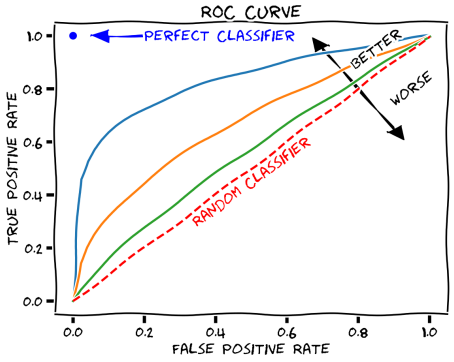
Results

Given how small my dataset was, I was surprised with how accurate my classifier turned out to be. When training my dataset, I set aside 280 images to use for testing and 1,120 for the actual training of the model. After constructing the dataset, and some testing, found that the classifier yielded approximately 76.428% in image detection accuracy.

```
Model accuracy is:  0.7642857142857142
```

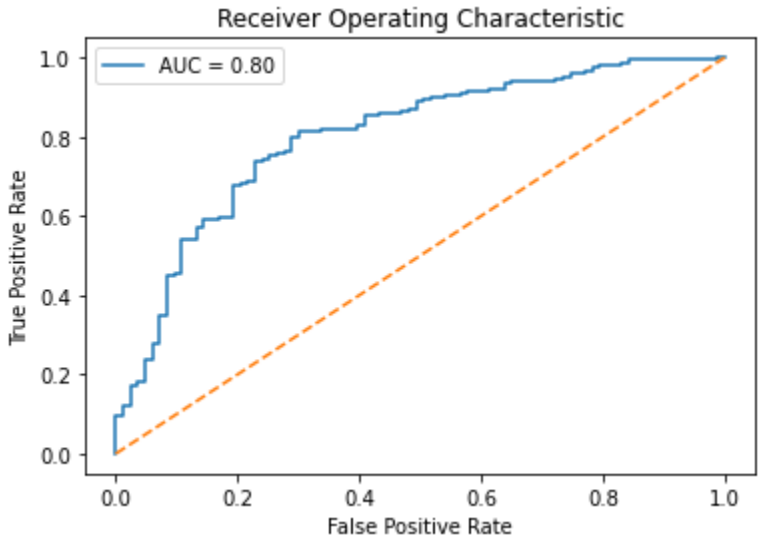
Exact accuracy of the SVM garbage bin classifier

I plotted a receiver operating characteristic curve (ROC curve) to better understand the performance of the classifier. When reading an ROC curve, if the graph curves closer to the top-left most portion of the graph then the classifier is reliable. If the curve is riding along the centerline, or even below the centerline, then it means that the classifier is either inconsistent, overfitted, or in most cases entirely random. See the next page for a sample graph showing the possible outcomes for the reliability of a classifier and what a user would ideally like to see after running their tests validating their classifier.



The closer the curve is to the top left, the better. The center curve would mean a random or inaccurate classifier [5]

As seen below you'll see the ROC Curve for my classifier and it very closely resembles the blue line in the example graph meaning it is quite reliable as a garbage bin classifier.



The curve for my classifier sharply curves towards the top left corner of the graph

Conclusion

In the end, my results returned an acceptable classifier that can be used to detect garbage bins in a city. There are a few things I could have done differently that could have increased the accuracy of the garbage bins; for starters, I could have recorded a much larger data set. Initially, I had planned to gather 5,000 data points and I have confidence that it is possible to do so with all the garbage bins present in Street View however I think having a classifier like the one developed over the past few weeks can exponentially grow the current dataset by simply finding trash bins on its own. Additionally, if I had time and a larger data set, I

could have further divided the classifications of garbage bins into other potential classes: recycling, dog waste, and cigarette bins. All three of those additional classifiers however would require an individual to retrieve them on foot as they're typically found in parks and other recreational settings. Those settings are normally not recorded in Google Street View and if they do appear in Street View the data size would not be sufficiently large enough for use in a classifier. Through research, I found that some garbage bin distributors like Big Belly already have a map describing the location of all of it's garbage bins so the developed classifier would find it's use for cities that don't have well documented garbage bin locations or locally provided garbage bins. Ultimately, I'd like to use this classifier and pair it up with an application that detects garbage bins on street corners to help a user find the nearest garbage bin on a map in the future. By offering a garbage bin finder tool, one may be less inclined to litter if one knows where to properly dispose of garbage and hopefully curb some of the litter problems some cities like Philadelphia have.

Acknowledgements

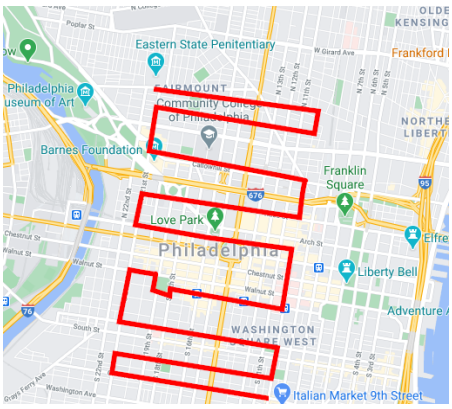
A big help during this project was Sharon Morris' article on SVM classification with different species of flowers [1]. His project was very similar to mine in which we wanted to identify a specific subject and so a lot of his work was relevant in regards to the scope of my project. There are some key differences however, Sharon was testing the data set against itself because there were multiple classes to consider (different flower species). For my classifier there was only one class that we were training for but with the introduction of a binary "0" class for whatever wasn't the class we were training for. Another difference was the organization of our data. Sharon had a csv file with the directory/file location, and classification of all his plants. I manually retrieved and set the classification in the script itself because my setup was unique in where I could programmatically label the classes off hand since the data set was already curated beforehand.

Timeline Summarized

Week 1

I have amassed 1,000 garbage can images in the first week of the project. In my initial proposal I noted that I would like to collect 5,000 images but given the time constraints I may have to settle with 1,000-2,000 images and test to see if what I currently have is sufficient to generate an adequate classifier. Approximately ~180 images are real life photos of garbage bins I manually took myself walking around the city with the remaining images being sources from google maps street view.

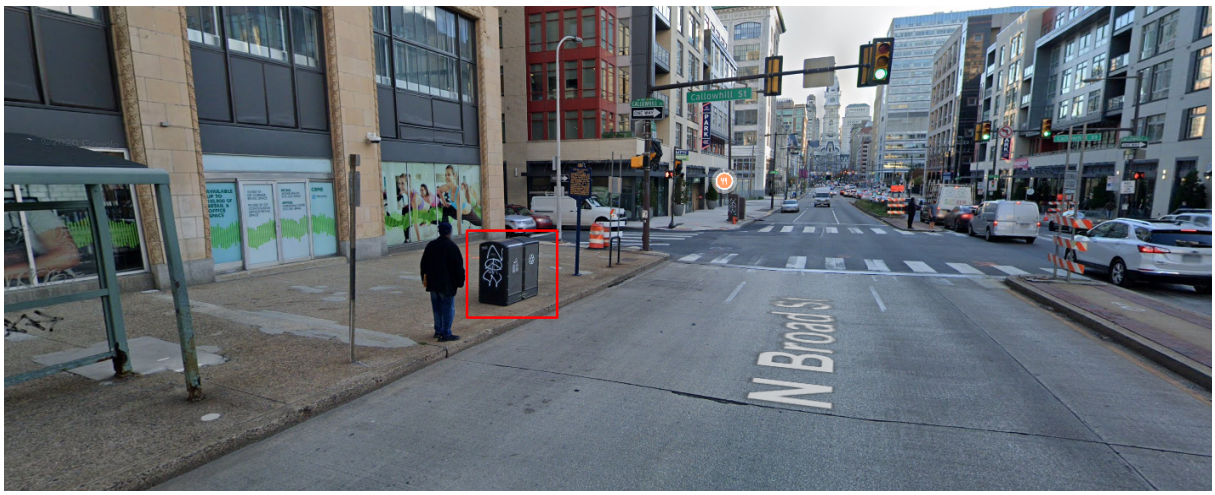
When collecting images in person, I tried to map out a route that would most efficiently net me garbage cans without backtracking unnecessarily. Below is the route I took while retrieving images.



To save time I systematically traversed center city to record as many unique garbage bins as possible

Note that while going through this path there were parks along the way, despite not shown on the map I took detours through those parks because the trash bins there were unique enough to justify adding to the dataset of garbage bins.

While collecting images through Google maps I used an application called Lightshot to help me crop out the portion of the image that would just contain the garbage bin. SVM requires only the subject and nothing else when building a classifier. To show what the process was like, refer to the image below and note the red rectangle outlined.



When screenshotting images I binded screenshot to Ctrl+W and save to Ctrl + S to make recording data as fast as possible

The red rectangle would be what I'd screenshot using Lightshot. Lightshot also allows me to set keybinds for screenshotting to make saving images nearly as fast as I can click and drag my mouse over the trash bin. I also took advantage of the time capsule function in Google maps that allowed me to go back in time and take pictures of different types of trash cans that were eventually replaced with either Big Belly or simply a new receptacle. Below are some of the types of trash bins that I recorded.




The cities I visited in Google maps were: Philadelphia, New York, San Francisco, Seattle, and Los Angeles.

Week 2

Below are the functions describing how I'm processing the images and retrieving the hog, these same notes can be found in my github repository for the project (links at the end as always). I haven't looped over the entire image set, this is mostly setup.

retrieveAllImages()
Summary:

- **Returns a list of all the files in a directory, our usage will simply be to retrieve the images**

Parameters:

- **directory (str): Name of the directory being loaded into a list, the default is './garbage bins'**

Returns:

- **fileNames (str[]): list of all the filenames in the directory passed or defaulted value**

returnImageArray()
Summary:

- **Returns a list of all the files in a directory, our usage will simply be to retrieve the images**

Parameters:

- **filename (str): Name of the file being located**
- **directory (str): Name of the directory being loaded into a list, the default is './garbage bins'**

Returns:

- **fileNames (str[]): list of all the filenames in the directory passed or defaulted value**

processImageFeatures()

Summary:

- **Returns a list of the image's features, it tries to flatten the information into a single array instead of multiple**

Parameters:

- **image (np.array): We feed in the array that we generate using returnImageArray()**

Returns:

- **flatify (np.hstack): gives us the single dimension list describing the image in grayscale**

getHOG()

Summary:

- **Returns a histogram of gradient directions that can be previewed in jupyter**

Parameters:

- **image (np.array): We feed in the array that we generate using returnImageArray()**

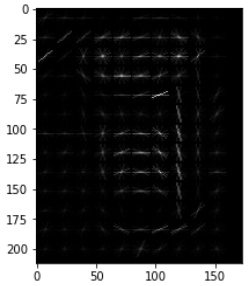
Returns:

- **grayHOG (skimage.feature.hog): Returns the HOG describing the image, it specifically returns a multidimensional array**

Example of A Processed Image and HOG

My test image seems to give me an output better than I was expecting, according to what we see below a lot of the information from the garbage can itself is retained. It's yet to be seen how the HOG will be affected by information that isn't the garbage can across the entire image set. It seems to detect sidewalk lines but the information retained from them does not seem to be significant.

Pre Grayscale and HOG



Post Grayscale and HOG

My main concern will definitely come from not capturing all of the information I'd like from the garbage can. As you can see, the leftmost edges of this garbage can aren't as clearly retained as the rest of the image. It'll be interesting to see how the resolution of the image affects the information gathered. The one above is actually quite high in resolution all things considered and if I were to generate a hypothesis, I'd assume that a lower resolution may be beneficial in making sure information we don't want isn't retained.

Week 3

I looped over the entire image set and attempted to train/test the processed image set. I say "attempted" because I ran into a small problem. The resolution or core feature of the image was not the same across the image set. For me to run a HOG feature description there are a few assumptions that must be true about the image set:

1. The core features of the image must be the same, this includes: the RGBA channel, the file format of the image (as this can affect the color channels causing errors in conversion), and the resolution of the image.

2. You MUST have another class involved, you cannot train or test a data set with a singular set even if you only want to train for one class (like in this instance where I only want garbage bins). The reason for this is because you need a set to test and train against so that the classifier can further solidify what is or isn't a garbage bin.

Below are some additional functions I introduced later in the project and their purpose:

returnClassification()

Summary:

- Returns the values of the image set we're training a model for.

Parameters:

- N/A

Returns:

- arr (int[]): Returns a list consisting of 400 "0"s and 1000 "1"s, the "0" represent everything that ISN'T a garbage bin while the "1" represents everything that IS a garbage bin. I could have simply had a csv that listed all of the filenames and their associated classification but I didn't realize until later that I needed 2 classes even if I were just testing for one. There are ways to test for only one class without needing another image set but this solution was a pretty decent bandaid.

massRenamer()

Summary:

- A small utility function for renaming all of the images from the Food-5k set that I'm using. The Food-5k set specifically isn't importing the food portion of the set but everything that ISN'T food.

Parameters:

- directory (str): Name of the directory being loaded into a list, the default is './not garbage bins/'

Returns:

- Void: It doesn't "return" anything but helps rename everything in a directory.

Relevant Links

1. *Garbage Bins Image Folder*
   https://drive.google.com/drive/folders/1bOWipnnd61DI5Npr6YSKNQc6QlNKmySQ?usp=sharing

2. *Garbage Bin SVM Repository*
   https://github.com/flantig/Garbage-Bin-SVM-Model

References

3. Morris, Sharon. “RPubs - Image Classification Using SVM.” RPubs, 27 2018, https://rpubs.com/Sharon_1684/454441.

4. Singh, Avinash. “Vehicle Detection Using Support Vector Machine(SVM) | by Avinash Singh | Towards Data Science.” Medium, Towards Data Science, 22 Apr. 2018, https://towardsdatascience.com/vehicle-detection-using-support-vector-machine-svm-19e073b61d16.

5. Stemplicity, Dr. Ry. Histogram of Oriented Gradients (HOG) | By Dr. Ry @Stemplicity. YouTube, 5 Aug. 2019, https://www.youtube.com/watch?v=XmO0CSsKg88.

6. BinhMinhs10. Food-5K | By Binh Minhs
   Kaggle, 28 May. 2019 https://www.kaggle.com/binhminhs10/food5k

7. Receiver operating characteristic
   https://en.wikipedia.org/wiki/Receiver_operating_characteristic