Car Counting

Estimating vehicle counts from satellite imagery.

CEGEG075 Spring 2018
Image Understanding

1873 Words

Introduction

In 2010, a novel method employed by American mega-retailer WalMart to project quarterly sales made headlines (Javers 2010). While cutting edge hedge fund techniques rarely escape the Wall Street Journal or Planet Money podcasts, the implications of this bold approach captivated (and alarmed) consumers and private citizens. Using satellite imagery purchased from commercial providers, financial analysts counted the number of cars in one hundred WalMart parking lots over several months and used advanced modeling to predict sales trends.

As a proof of concept, this method relied on manual counting of cars in a relatively small number of parking lots, a feasible if arduous approach. Yet in the intervening eight years, the amount of imagery available to private buyers increased exponentially. In February 2017, Planet completed the most audacious imaging infrastructure development to date, launching a full constellation of 88 Dove satellites capable of imaging every part of the earth every day (Schingler 2017).

Beyond estimating quarterly sales, counting objects in satellite imagery provides vital information for military, humanitarian, and disaster relief missions. Publicly available satellite imagery was used to identify a build up of ground forces near the Ukraine border in the Russian cities of Belgorod, Kuzminka, and Novocherkassk in April 2014, prior to the escalated conflict that occured that summer (Wolfinbarger et al 2014; Cendrowicz et al 2014). Imagery was used to count refugee dwellings at the Rukban border crossing between Jordan and Syria and identify evidence of burial sites (Taylor 2016), and vehicle counts could be used to understand the extent of displacement caused by natural disasters (Carlos Davila, FEMA Office of Response and Recovery, personal communication, July 2014).

With over 300 million km^2 of imagery being collected every day by the Dove constellation alone, manually counting the number of vehicles - even in small patches - is a Sisyphean task. Traditional image processing techniques struggle with the wide variety of shapes, sizes, colors, and poses that vehicles may take. Machine learning techniques such as SegNet have provided encouraging early stage results for other problems and could feasibly be adapted for this problem set.

This project endeavors to segment vehicles from satellite imagery. Two techniques are tested. First, region growing, briefly explained in class, is adapted to identify car-shaped blobs from 15cm EO RGB imagery. Finally, an ambitious (and unsuccessful) attempt is made to adapt SegNet for vehicle segmentation.

Methodology

Dataset

This project uses the Cars Overhead With Context (COWC) dataset as ground truth. The dataset contains 15 cm RGB EO imagery from six international cities with both positive and negative examples of cars, as shown below. More information on this dataset is available in the original paper (Mundhenk et al, 2016). Unfortunately the ground truth contains only one pixel in the center of the example, and does not contain fully segmented examples.

Figure 1: Close up of parking area in Potsdam, Germany, showing positive examples of cars (red pixels) and negative examples (blue pixels) from the COWC dataset.

Accuracy Assessment

For the purpose of this assignment accuracy is defined as:

*100 * # true positives / (# true positives + # false positives + # false negatives)*

This definition of accuracy, termed "quality percentage" in Jan-Peter Muller's Lecture 3, ensures that accuracy statistics will not be artificially inflated by methods that provide significant false positives.

Since the dataset used contains only single pixel locations, not fully segmented vehicles, intersection over union and other similar accuracy assessments are not appropriate for this dataset.

Region Growing

Region growing segments an image by selecting all pixels adjacent to a "seed point" that are within a threshold distance of color. This iteratively progresses until adjacent points are too different from those previously selected; the selected pixels are all categorized as belonging to the same region.

While vehicles are a variety of shapes, sizes, and colors, they are similar in area. By restricting results to regions of a specific size, it is possible to segment regions that are likely to be car. In this case, "blobs" are restricted to reasonably rectangular size not exceeding 37 pixels in width or height.

This process is repeated for each 360x480px image tile using a large number of random seeds. A blob is determined to be a car if the COWC car ("truth") pixel (shown in red in Figure 1) is within the blob. The accuracy assessment does not account for cases where a blob may contain non-car pixels in addition to car pixels and truth pixels.

A number of variables within this method need to be optimized: number of seed points, neighborhood definition (queen vs rook), vehicle patch size, segmentation threshold, region ratio (how square blobs must be), and maximum region dimension (how long/tall blobs can be). Due to computational constraints, a brute force approach was taken to optimize only number of seed points and segmentation threshold.

SegNet

It is apparent even prior to seeing results that the region growing method will catch plenty of objects that are not vehicles but are similar in size and shape, such as shipping containers, AC units, or hedgerows. One method of circumventing this concern is using an adaptation of SegNet, a machine learning technique used to segment every pixel in an image of a street scene into a number of classes such as street, vehicle, sky, pedestrian, etc (Badrinarayanan, 2016). This method has been adapted for many cases, such as identifying building rooftops (Ørstavik, 2018).

While this method could likely be adapted well for car counting, obtaining sufficient data and processing power to sufficiently train it remains a significant challenge.

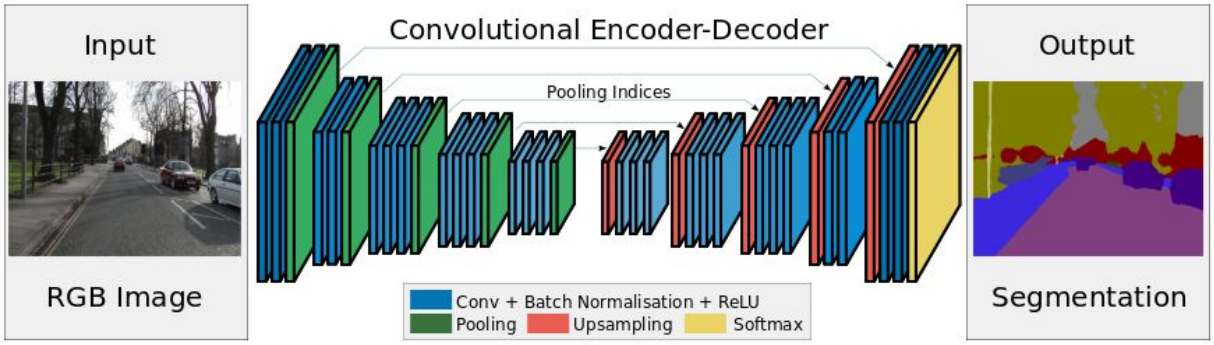

Figure 2: Visual description of SegNet process (Badrinarayanan, 2016).

Results

To determine the best parameters, tests were run where the number of seed points was varied between 100 and 2500 and the threshold between 0.001 and 0.5. All tests were conducted on tile 15 of the image 'top_potsdam_2_10_RGB_Annotated_cars.tif'. Due to the random nature of the image seeding, each test was performed five times and the average accuracy (as described in methodology, above) recorded. The combination of these parameters that led to the overall lowest value was selected.

		Rook's Case					
		Threshold Value					
		0.001	0.01	0.05	0.1	0.25	0.5
Seed Points	100	3.65	1.85	2.51	2.07	2.85	1.29
	500	5.66	4.56	5.87	3.86	3.97	5.39
	1000	8.52	9.71	8.22	10.02	9.09	8.59
	1800	17.60	**22.78**	16.83	19.04	19.87	20.77

Table 1: Results of the brute force optimization for variable estimation showing the percent accuracy for selected values tested. The lowest combination detected is shown in bold.

Once the values for the threshold and seed point had been selected, they were tested on different data from the COWC dataset from a different area of Potsdam, Germany. The results were computed tile by tile, and are shown aggregated below. Due to computation constraints, the accuracy was only computed once. Ideally, an average for each tile using different random seeds would be preferred.

Dataset	**Min Accuracy**	**Mean Accuracy**	**Max Accuracy**
Potsdam	0	4.92	25.00
Potsdam (zeros removed)	3.03	14.75	25.00

Table 2: Results of tests on the Potsdam dataset.

Clearly, these results are far from sufficient. The best possible case was 25% accuracy, and of the 12 tiles tested only 4 had nonzero accuracy values. While it is possible that those images simply don't have any cars (many of the tiles in the test dataset have no cars at all), it is also possible that the estimate provided by the algorithm is deeply flawed.

As Figures 3 and 4 below show, plenty of things are segmented that are not vehicles.

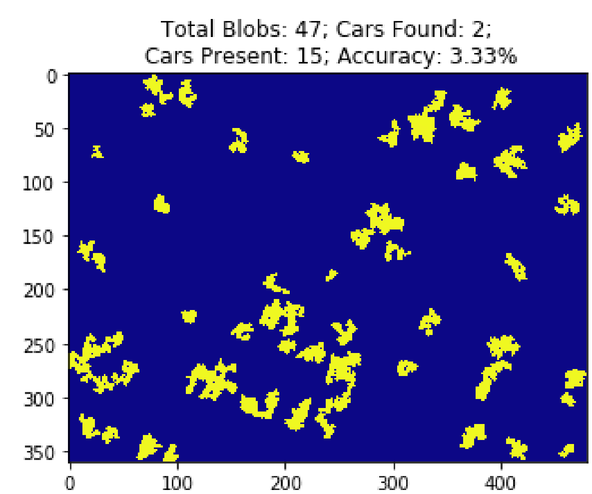

Figure 3: Blobs identified within one image tile.

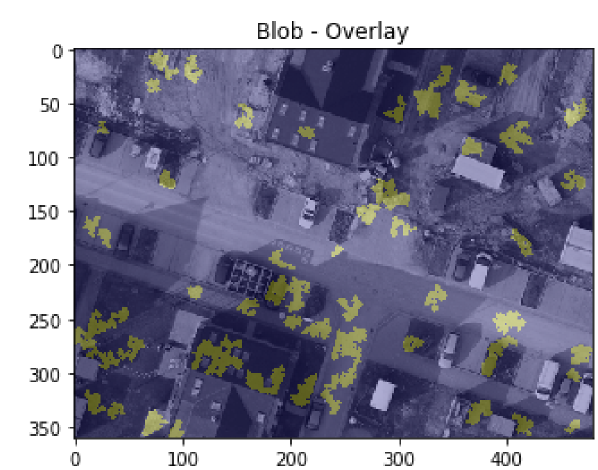

Figure 4: Blobs overlaid on grayscale image.

Despite significant effort, I was unable to get Caffe (a prerequisite for SegNet) functioning on my machine due to dependency conflicts. One particular quote from among the myriad of (occasional contradictory) internet resources seems to sum up my experience: "Do not install [dependencies] through brew. Only misery lies there (Smolyansky, 2016)."

Discussion

Clearly, region growing using random seeds is not a sufficient method for segmenting vehicles from satellite imagery. There are a few reasons specific areas where this algorithm could be improved.

First, the decision as to whether a region-grown blob is a vehicle is not particularly robust. As shown below in figures 5 and 6, blobs can be of various shapes so long as their overall dimensions are approximately rectangular. Ensuring that blobs are necessarily compact would mitigate this concern.

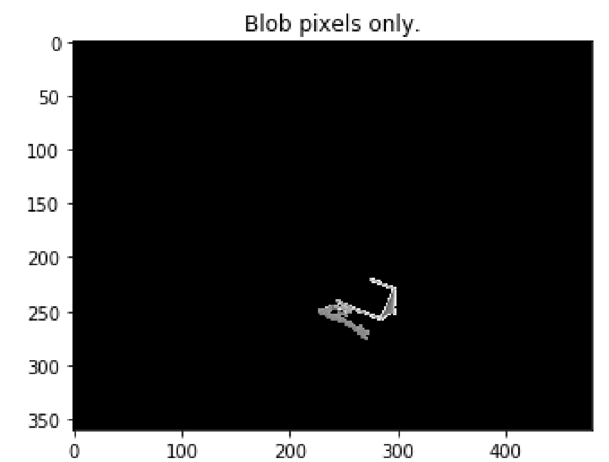

Figure 5: A snakey blob.

Figure 6: Snakey blob overlaid on grayscale image.

Second, this approach did not optimize all possible variables, and the only optimization performed was incredibly coarse. A more sophisticated method, such as gradient descent, would allow for more specificity and optimization over all variables.

Third, the processing time required to address all possible parameter combinations far exceeded the ability of the author's nobel but aging 2013 MacBook. Computation of the optimal parameter values, even for only three of the many possible trained parameters, was stopped after over 6 hours of continuous runtime. Future uses of this should make use of gradient descent or other optimization algorithms.

Finally, optimization (or "training") of these variables should be performed over a larger set of tiles. In this example, they were only trained over one tile due to computational constraints.

As discussed briefly in the Methods section above, methods like SegNet are state of the art for pixel segmentation. As shown in Figure 2, SegNet consists of a number of different layers. Generally, this consists of a series of non-linear pixel wise processing layers ("encoders") and decoders. In addition, pixel wise processing layers are used. This method allows for more information to be included about the area surrounding the car in a nonlinear fashion. Such context may be crucial in differentiating blobs that are cars from those that are not; for example, cars are not usually found in fields, while small sheds may be.

Ideally, an implementation of SegNet to address this challenge would use many different pixel classifications, such as car, road, water, other, to provide the context needed. However, identification of specific car features may also be needed to identify cars in parking lots, for example, from nearby cement barriers (see the far right negative example in Figure 1).

Conclusion

As this assignment demonstrates, region growing using random seeds is *not* an appropriate method for segmenting vehicles from satellite imagery. The method herein proposed achieved at best 25% accuracy within a single tile, and at worst as low as 3% (or even 0%). This is, depending on the density of vehicles in an image, even worse than change.

While region growing could be integrated as part of a larger algorithm, this example shows it does not offer the robustness required for car counting, nor can it differentiate between objects that are car-shaped (such as square hedges, patches of grass, or garage roofs). A true implementation of this question would necessarily make use of advanced machine learning techniques, such as SegNet, or at least more sophisticated multivariate optimization techniques, such as gradient descent.

References

Badrinarayanan, V., A. Kendall, and R. Cipolla, 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *PAMI*. Available at: https://arxiv.org/pdf/1511.00561.pdf [Accessed March 25, 2018].

Cendrowicz, L, L. Harding, and A. Luhn, 2014. Satellite images reveal Russian military buildup on Ukraine's border. *The Guardian*. Available at: https://www.theguardian.com/world/2014/apr/10/satellite-images-russian-military-ukraine-border. [Accessed March 25, 2018].

Javers, E., 2010. New Big Brother: Market-Moving Satellite Images. *CNBC*. Available at: https://www.cnbc.com/id/38722872. [Accessed March 25, 2018].

Mundhenk, T.N., Konjevod, G., Sakla, W.A. and Boakye, K., 2016. A large contextual dataset for classification, detection and counting of cars with deep learning. In *European Conference on Computer Vision* (pp. 785-800). Springer, Cham. Available at: https://gdo152.llnl.gov/cowc/mundhenk_et_al_eccv_2016.pdf. [Accessed March 26, 2018].

Ørstavik, M., 2018. Extracting Intelligent Information from Aerial Images using Machine Learning. At *FOSS4GUK Conference 2018*.

Planet, 2018. SkySat Ortho Scene. *Planet*. Available at: https://www.planet.com/products/hi-res-monitoring/. [Accessed March 1, 2018].

Smolyansky, E., 2016. How To: Set up Caffe Environment and pycaffe on OS X 10.12 Sierra. *Eddie Smolyansky* (blog). Available at: https://eddiesmo.wordpress.com/2016/12/20/how-to-set-up-caffe-environment-and-pycaffe-on-os-x-10-12-sierra/. [Accessed March 25, 2018].

Schingler, R., 2017. Planet Launches Satellite Constellation to Image the Whole Planet Daily. *Planet Pulse*. Available at: https://www.planet.com/pulse/planet-launches-satellite-constellation-to-image-the-whole-planet-daily/. [Accessed March 25, 2018].

Taylor, A., 2016. 75,000 Syrians are trapped near Jordan's border. Satellite images show some are dying. *Washington Post*. Available at: https://www.washingtonpost.com/news/worldviews/wp/2016/09/14/75000-syrians-are-trapped-near-jordans-border-satellite-images-show-some-are-dying/. [Accessed March 14, 2018].

Wolfinbarger, S., J. Drake, and E. Ashcroft, 2014. Satellite Imagery Assessment of the Crisis in Ukraine Part Two: Border Deployments. *AAAS*. Available at: https://www.aaas.org/sites/default/files/Ukraine_Border_Deployments.pdf. [Accessed March 25, 2018].