Scene Tagging Summer Project

Holistically determine subjective labels for street view images of New York.

Goal of Carmera Internship Project

Classify street view images under subjective, descriptive labels.

Summary

Successfully classified images on the following tags: safe, residential, outside venue, tourist, old, rich, construction, commercial, green, family friendly, beautiful, desolate, vibrant. Created a simple web app to streamline manually labeling images for training. Finetuned the ImageNet from caffe to train two different convolutional neural networks. Compared network results to the MIT safety streetscores under a binary classification framework, the networks had a 91.5% and 97.3% accuracy rate.

Possible Directions to Continue

- Continue finetuning the networks, changing the learning parameters, or network structure to find networks that most accurately classify different images under a range of tags.
- Increase the labeled dataset for training. More images to train the network on will improve the results of the network and prevent overtraining.
- Add any potential tags that would be useful data considerations.
- Perform more accuracy testing, especially on the remaining tags.
- Perform statistical analysis to determine what network output probabilities function as the best cutoff value for a binary classification system. Determine how to rank the network output probabilities into a more continuous rating system.
- Perform scene tagging with the inclusion of objection recognition. Certain objects may correlate heavily to certain tags (e.g. fire trucks and safety, trees and greenness)
- Continue adding different algorithms and networks as a way of cross-referencing to boost accuracy.
- Move to semi-supervised training to deal with such a large image dataset.
- Explore construction, tourist, residential, vibrant, and green tags. These are the ones that I believe are the best to continue exploring beyond ‘safe’.
- Continue safety testing, most of the verification was performed on images considered ‘safe’ and only on roughly 1/30 of the coordinates. Relate the q-score to a continuous gauge of the network output, move beyond binary classification.

August 5, 2016 Patrick Chen

Questions to Consider

- How to obtain quality, consistent data that reflects specific tags and is specific to Carmera image database?
- How many different networks or algorithms should be trained and tested?
- How should the work between differentiating scenes be distributed among networks?
- For multiple scene descriptors, how is it possible to build them on top of each other to maximize efficiency?
- What preprocessing techniques should be employed to improve image quality?
- What role will object detection play in junction with scene labeling?
- Automating the process of scene detection?
- Differentiating images that were taken at different angles?
- To what extent can networks already trained on a broader dataset be employed for scene recognition?
- Determining what the cut off value should be for multiple outputs of a network?

Possible Scenes

- Safe – Compared to MIT's streetscore
- Residential - Urban (apartment, hotel), Suburban (house)
- Outside venue (courtyard, park)
- Commercial – Small Shops, Financial Districts, Chain Stores
- Tourist
- Area of the city (Midtown, Chinatown, etc) – To what extent do the areas have differentiating looks?
- Old or Modern
- Poor or Rich
- Construction
- Considerations on above tags: Many of the tags after training seem related, repeated, or overlapped. Which of these tags are the most important? Also it is difficult to distinguish things such as age and price, even during manually labeling the images. A lot of the images can be classified into seemingly contrasting tags (e.g. apartments on top of small commercial shops).
- Added following tags under Ro's suggestions, used for the second network trained: green, family friendly, beautiful, desolate, and vibrant.

Results

- First network trained on: safe, residential, outside venue, tourist, old, rich, construction, and commercial.
- Second network trained on: safe, green, family friendly, beautiful, desolate, and vibrant.
- First network categorized the safety score of 9144 out of 9993 coordinates correctly, 91.5% accuracy.
- Second network categorized the safety score of 9725 out of 9995 coordinates correctly, 97.3% accuracy.
- The second network found 15 or more images within 50 meters of the radius for 6780 of the coordinates tested.
- The second network examined a total of 149895 images while testing the safety tags. 16.23% of the images where below the cut off probability value of the network output (.1). The remaining distributions are shown in the following array, broken up into .1 segments: {0.1623, 0.0420, 0.0498, 0.0876, 0.1643, 0.1850, 0.0877, 0.0723, 0.0635, 0.0854}. The second network seems to be fairly decent at distinguishing images with certainty, at a cut off point around .1-.3.
- Any errors during the testing phase can be examined in the error output text files.

Notes

- Pay attention between python2 and python3, pip2.7 and pip3. Caffe doesn't contain much support for python3, while Carmera has agreed to do everything in python3. Right now the Granola production API can be used with python2, but the staging API cannot (staging API was used for earlier stages of project).
- Figure out how MIT Streetscore converted q-score to a linear, continuous score.
- During training, the learning rate of the last layer (the one with randomly initialized rates) is higher than the rest of the network. This is to reflect the need to train the last layer faster since it has no starting point.
- During training the learning rate will also drop as the network begins to learn. As the network approaches a local or global minimum there is less need to continue changing the weights as much.
- The network loss will approach some minimum. It should never approach zero since you are comparing binary training data to a network output probability. Also, you want to avoid overtraining, where the network performs extremely effectively on the training data but loses its flexibility to accurately examine new images.
- I setup a different user on the Amazon gpu Ubuntu instance, called scene. It is setup so you don't need to physically type the .pem key. Use the following command to ssh into the instance: ssh scene@current.ip.address