Object Detection Using Convolutional Neural Networks

Progress

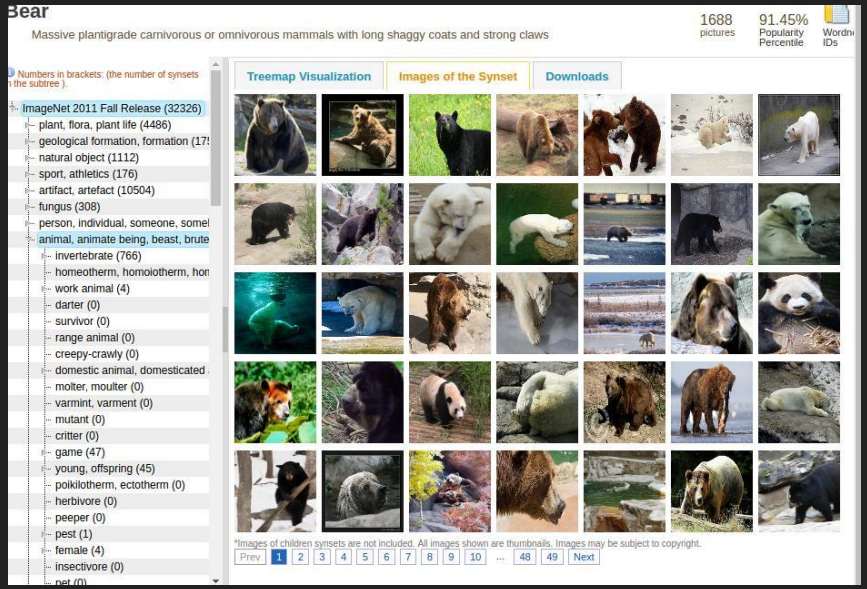
Guided by:
Dr. Sushil Kumar

Amlaan Bhoi
A2305213303
7CSE4-Y

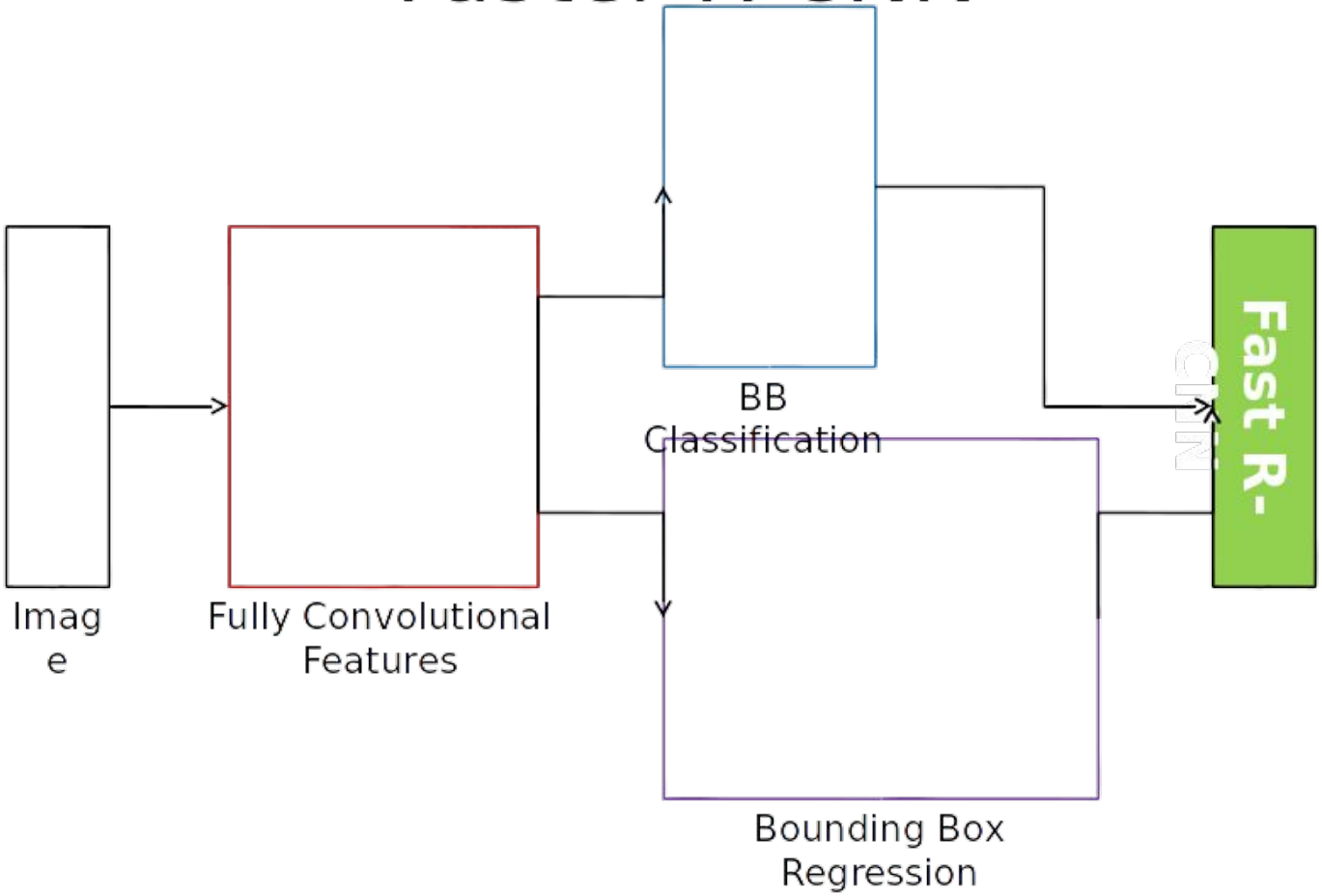
Dataset

Large Scale Visual Recognition Challenge 2016

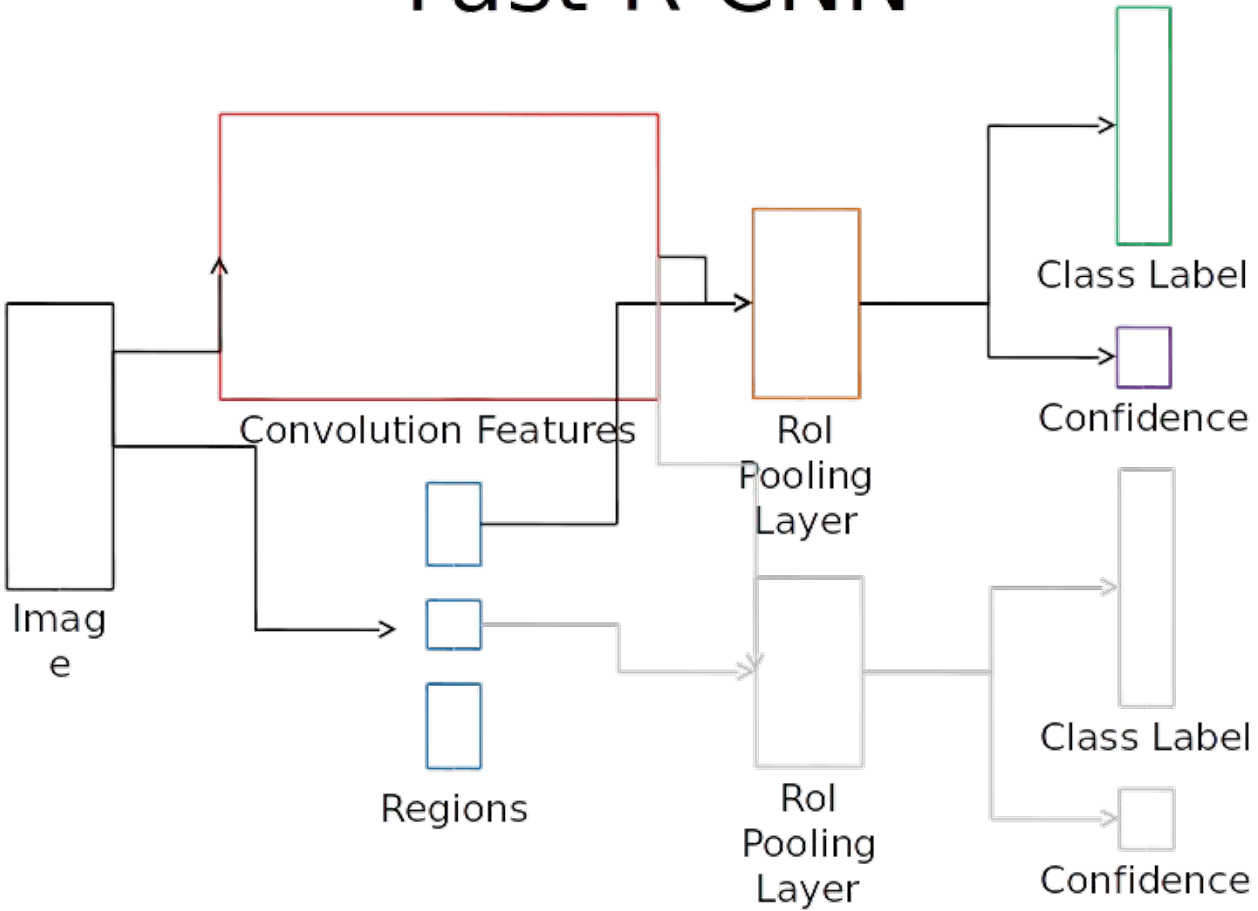
http://image-net.org/challenges/LSVRC/2016/browse-det-synsets



Faster R-CNN


Imag
e
Fully Convolutional
Features
BB
Classification
Bounding Box
Regression
Fast R-
CNN

Fast R-CNN


Class Label
Confidence
Convolution Features
RoI Pooling Layer
Imag e
Regions
RoI Pooling Layer
Class Label
Confidence

Models

Python Implementation

https://github.com/rbgirshick/py-faster-rcnn

Matlab Implementation

https://github.com/ShaoqingRen/faster_rcnn

Issues?

GPU Implementation

GPU: Titan, Titan Black, Titan X, K20, K40, K80.

1. Region Proposal Network (RPN)
    - 2GB GPU memory for ZF net
    - 5GB GPU memory for VGG-16 net
2. Object Detection Network (Fast R-CNN)
    - 3GB GPU memory for ZF net
    - 8GB GPU memory for VGG-16 net

CPU (56 seconds for 10k images) vs GPU (26 seconds for 10k images)

Trial* using Caffenet (AlexNet variant ~2012)

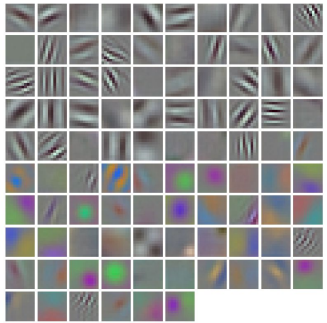
Training file: testy.py
Output: testy_run_log.txt

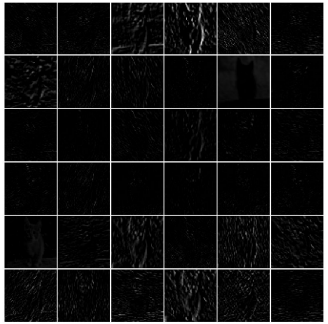
Results:

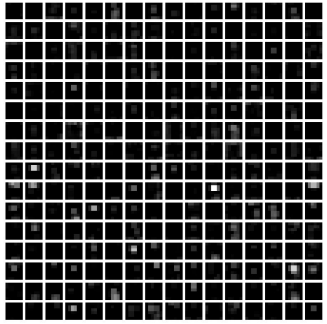
Mean-subtracted Values:
[('B', 104.0069879317889), ('G', 116.66876761696767), ('R', 122.6789143406786)]
Predicted Class:
281
Output Label:
n02123045 tabby, tabby cat
CPU Forward Pass Time:
6.66855120659

* Image classification

Original Image


filter1


conv1


pool5

Next Steps

- Faster-RCNN implementation
- H/W requirements (AWS possible solution)
- Pre-trained Model Zoo (https://github.com/BVLC/caffe/wiki/Model-Zoo)