Regional Image Recognition using R-CNN

Brendan Case* Guy Coop† Vasileios Mizaridis‡
Priyanka Mohata§ Liangye Yu¶

December 7, 2017

Abstract

Image Recognition has, in recent history been relatively "solved" in the sense that algorithms have now been recorded outperforming humans in simple image recogntion tasks. However regional image recognition where the algorithm is tasked with recognizing multiple images inside a whole chaotic scene is still an emerging field. Our team analysed the most promising methods of regional image recognition, and implemented our own solution to a simplified regional recognition task. Our solution *was able to locate and classify objects inside an image with a 92% accuracy of locations and a 100% accuracy of classification.*

*bkc721 - 1801421
†gtc434 - 1447634
‡vxm716 - 1844216
§pxm374 - 1341274
¶lxy736 - 1810736

Contents

1 Introduction

For this project, our team was assigned the task of implementing a regional image recognition system.

1.1 Data sets

The data set provided was given in the following format:

- Data: 400x400 RGB image files in .jpg format
- Labels: each image has a corresponding text file that describes the location of each of the predefined objects in the image. This location was given as pairs of integers describing horizontal runs of pixels that form a rectangular bounding box around the object. If the object was not present in the image it was given as [object 1 0] meaning that it had a run length of 0 pixels.

Training Set The training set provided contains 14,625 (image, label) pairs that includes image files and label files that describe the objects and bouding boxes. for conducting experiments, this data set should be subdivided into a Training set and a Validation set to measure how parameter changes affect the accuracy of the results

Test Set The test set contained 2500 images that are given to simulate unseen data coming into the system. These images do not have the associated label files, and our task is to produce label files that describe the location and class of objects in each of the test image files.

1.2 Our Aims

Our aims for this project are as follows:

- Discover and Analyse currently exisitng method of performing regional image recognition
- Produce our own implementation of one of these methods, using any necessary packages or source code segments as necessary.
- Conduct experiments to optimize the recognition system in terms of bounding box locations, and object classification.
- Produce a set of conclusions about the effectiveness of the various recognition methods, and the optimal parameters of our implementation for the data set provided.

2 Design

For this task our Initial instict was to attempt to solve it with a simple feedforward neural network. And whilst this would have been adequate for a simple classification task, we determined it would not provide a good solution to this region bounded classification task. From there we exaimned other possible methods of solving the task. The first sub-method we investigated was using Selective Search to identify regions of interest inside the image. We then expanded this to determine other pre-established algorithms that make use of selective search and how they compare to other similar algorithms.

2.1 Selective Search

Selective Search is an algorithm used for regional image searching....

2.2 Regional Convolutional Neural Networks (RCNNs)

Given the regional nature of this task, The first option that should be analysed is "Regional Convultional Neural Networks" and their successors. There are three implementations of this algorithm that will be examined:

- R-CNN [GDDM14]
- Fast R-CNN [Gir15]
- Faster R-CNN [RHGS15]

R-CNN R-CNN [GDDM14] makes use of selective search to generate the region proposals. it typically produces around 2000 region proposals per image, these regions are then sent forward to the CNN in order to determine if they contain an object in the dataset, and what that object is. Once the objects have been detected in the bounding boxes, regression algorithms are used to tighten the bounding boxes more accurately around the objects.

Fast R-CNN Fast R-CNN [Gir15] is an update on the original R-CNN technique that was developed in 2015, it acheives approximately a 9x speed-up on the original at train time, and over 200x speed-up at test time. It does this by unifying the training phase of the boudinding boxes, and the object classification algorithm into a single round of training, rather than having to train the two algorithms seperately.
It also makes use of Region-of-Interest (RoI) pooling layers. "RoI max pooling works by reducing the $h*w$ RoI window into an $H*W$ grid of sub-windows of approximate size $h/H*w/W$ and then max-pooling the values in each sub-window into the corresponding output grid cell." [Gir15]

Faster R-CNN Faster R-CNN [RHGS15] is another significat update on the Fast R-CNN technique that acheives another dramatic speedup. This algorithm was designed as part of an attempt at real time regional image recognition, and as such is able to operate in almost real time.
Similar to Fast R-CNN it makes use of the RoI pooling layer to create a significant imporvement in performance over traditional R-CNN. Faster R-CNN also introduces a Region Proposal Network (RPN). The RPN shares convolutional features with the detection network. This allows it to provide almost "cost-free" region proposals to the system.
Fast R-CNN is recorded to be able to operate at approximately 5 frames-per-second (fps) when running on a GPU, meaning that it could be used for real time applications.

2.3 "You Only Look Once" (YOLO)

"You Only Look Once" (YOLO) was named as such because the algorithm centers around only performing a single pass across the image, rather than having to analyse the same data multiple times.

3 Implementation

–Using RCNN
–TensorFlow Layers [Goo17]
–SciKit Image package [sidt17]
–YOLO_v1 [RDGF15]
–using YOLO [RF16]
–Darknet

4 Experiments

5 Conclusion

6 Description of Collaboration

An overview of each member's contribution to the project is given below:

Brendan Case: did something

Guy Coop:

- Produced data_handler python class used by multiple systems to reformat the training data into the required input format for the neural network.

And reformat the output from the network back into a format that matches the input.

- Lead the writing of the report, and collated it into a LaTeXdocument.

Vasileios Mizaridis: did something

Priyanka Mohata: did something

Liangye Yu: did something

References

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. R-cnn for object detection. 2014.

[Gir15] Ross Girshick. Fast r-cnn. *arXiv:1504.08083*, 2015.

[Goo17] Google. Tensorflow 1.0, 2017.

[RDGF15] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *arXiv:1506.02640*, 2015.

[RF16] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv:1506.01497*, 2015.

[sidt17] scikit-image development team. skimage 0.13.1, 2017.