Unified, real-time object detection

Final Project Report, Group 02, 8 Nov 2016

Akshat Agarwal (13068), Siddharth Tanwar (13699)
CS698N: Recent Advances in Computer Vision, Jul–Nov 2016
Instructor: Gaurav Sharma, CSE, IIT Kanpur, India

1 Introduction

Object Detection and Recognition is one of the most important topics in visual perception. Correctly identifying the objects in its environment and hence inferring their expected behavior from memory is a skill that even young kids have, and is much needed by intelligent agents to navigate our world. The human visual system is highly advanced and can learn to identify an unknown object after only a few (in some cases, even a single) encounters, immediately learning about the object's properties for future use. Autonomous cars need to identify other cars, road markers, pedestrians, cyclists, obstacles, traffic signs, traffic lights, lane markings etc. Domestic robots need to identify with much greater accuracy the exact location of small objects like cups, toys, socks etc. Search engines need to learn object detection in order to better perform image retrieval and classification. The manufacturing industry needs object detection to allow machines to identify the correct tool they need or to identify anomalies in machines. Currently, object detection is at a stage where recently Dieter Fox proposed the 100/100 tracking challenge, which aims to identify and track 100% of the objects in a scene with 100% accuracy.

In our project, we have studied a new object detection technique, YOLO [13] and have done a parameter study on the network in order to identify where it can be improved. We have also trained a YOLO network, pre-trained on ImageNet[10], to classify objects in the KITTI [6] autonomous driving vision benchmark suite.

2 Related Works

Some seminal works on object detection are reviewed below. Most methods comprise of a long disjoint pipeline each of which needs to be tuned independently.

1. **Deformable Parts Model (DPM)[5]**: Extracts static HOG features maps from the image at 2 resolutions, correlates them with linear filters and trains a latent SVM. mAP of 33.4%.

2. **Selective Search [15]**: Uses bottom up segmentation to merge similar regions. mAP of 35%.

3. **R-CNN [8]**: Uses region proposals from object detectors, then extracts features from these region proposals using a CNN pre-trained on a large dataset like ImageNet [2] and fine tuned for object detection on the Pascal VOC dataset [4]. mAP of 53.3%. An improvement on this algorithm, **Fast R-CNN** [7] introduced a RoI pooling layer, leading to a speed up of 213$\times$ over R-CNN, but still taking 0.22 s/image.

4. **Faster R-CNN [14]**: Proposed a Region Proposal Network (RPN) that uses shared convolutional features for generating proposals. This algorithm achieved a maximum mAP of 59.9% at 17FPS.

5. **Single-Shot Multi Box Detector (SSD) [11]**: SSD discretizes the output space of bounding boxes into a set of default boxes with different aspect ratios and scales, per feature map location. mAP of 72.1% at 58FPS.

6. **PVANET [9]**: Modifies the feature extraction part to be deeper but have fewer channels. Adopts concatenated ReLU, Inception modules and HyperNet blocks, getting an mAP of 82.5% at 20FPS.

7. **R-FCN [1]**: Region-based, fully convolutional network with all computation shared on the entire image, using position-sensitive score maps. mAP of 83.6% at around 6 FPS.

3 Base Paper

YOLO (You Only Look Once) [13] is a real time object detection framework, processing images at 45 frames per second. Fast YOLO, processes 155 FPS on a Titan X GPU. The paper presents a unified architecture that replaces the slow, complex and hard to optimize pipelines in the earlier methods. A single CNN predicts all bounding boxes simultaneously along with the class probabilities of each box, trains features in-line optimizing them for detection, and performs non-maximal suppression directly from full images. Only a single network evaluation is required to predict which objects are present and where they are located. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance cf. training individual components separately as done in earlier approaches.

4 Approach

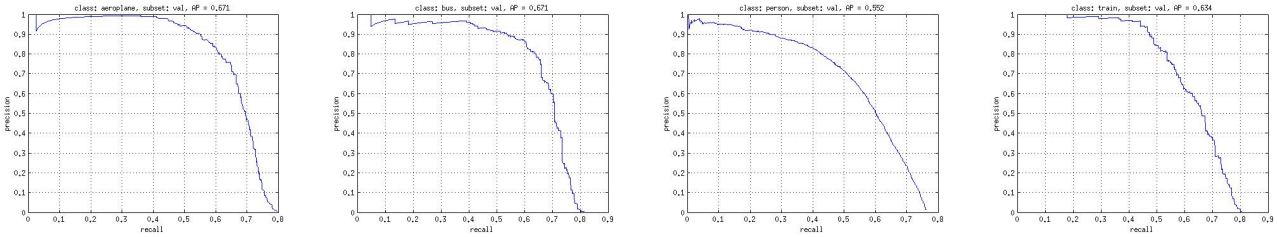
We trained a scaled down version of YOLO, with 8 conv layers followed by 1 FC layer. This network was pre-trained on ImageNet and finetuned on VOC 2007 training+validation and 2012 training set, and testing done on VOC 2012 validation set because the VOC 2012 testing data annotations are undisclosed. We also evaluated the performance of YOLO on dense urban scenes in the KITTI [6] object detection benchmarking suite. We conducted parameter studies by varying the most important parameters in the algorithm (size of grid and number of bounding boxes per grid cell) in order to analyse how they affected the localization ability of the network. Further, we trained another instance of our pre-trained (on ImageNet) network on KITTI Object Detection Evaluation 2012 and produced qualitative results on the same due to a lack of accessibility to the testing benchmark itself.

We used the Darknet [12] open source neural network framework which has been written in C and CUDA by Joseph Redmon. For evaluation we used the VOC development kit provided by the challenge organizers [4]. We developed the code in Python for porting KITTI data to a VOC and Darknet compatible format to train our network on it.

5 Empirical Results

Class	**S=5, B=2**	**S=7, B=2**	**S=9, B=2**	**S=9, B=4**	**S=9, B=6**	**Reported**
Aeroplane	58.69	61.7	67.1	60.92	56.53	-
Bicycle	52.28	47.77	56.86	51.64	36.37	-
Bird	26.25	23.17	33.28	26.28	15.41	-
Boat	17.7	15.42	20.18	13.49	7.6	-
Bottle	8.98	5.84	12.56	6.04	0.6	-
Bus	64.97	61.74	67.11	65.09	55.01	-
Car	38.53	34.49	41.94	38.64	30.00	-
Cat	61.57	59.1	70.09	59.6	50.07	-
Chair	16.58	12.01	20.92	16.06	8.44	-
Cow	31.14	27.96	35.63	28.07	7.9	-
Dining Table	28.89	35.81	45.62	35.46	28.58	-
Dog	52.84	48.51	51.94	47.14	35.1	-
Horse	49.97	45.95	56.05	49.8	37.29	-
Motorbike	53.99	56.51	61.25	56.69	38.8	-
Person	47.88	48.84	55.19	50.81	45.77	-
Potted Plant	10.06	8.05	12.58	8.99	1.78	-
Sheep	31.16	27.03	31.89	17.41	12.01	-
Sofa	33.94	50.19	47.86	37.29	11.37	-
Train	59.95	58.44	63.44	61.14	52.35	-
TV Monitor	31.65	33.52	37.54	32.49	16.65	-
Overall	38.86	36.92	44.65	38.1	27.39	52.7

mAPs for different grid size (S) and bounding boxes per grid cell (B) for all the classes of VOC dataset[1]



(a) Aeroplane (b) Bus (c) Person (d) Train

Figure 1: Precision-Recall curves

(a) Sheep from VOC 2012 Validation Set

(b) Mona Lisa with Cat

(c) Cars on the road, from KITTI

Figure 2: Object Detection examples on VOC, Artwork and KITTI datasets

6 Discussion

- The author's reported mAPs on the scaled down YOLO are slightly better than ours because he had 16k training images while we had 10k.
- We observe that increasing the grid size to 9x9 has a significantly positive impact on the overall mAP, taking it from 36.9 to 44.65. This is as expected because the network is now able to detect smaller objects too, which is especially favorable in cluttered scenarios.
- Surprisingly, we get an improvement with S = 5 as well, for the classes with larger objects like aeroplane, train etc. which usually take up most of the image.
- B = 2 outperforms B = 4 and 6 by a significant margin, which is unexpected. We postulate that this is because a greater B means that there is a greater chance of the wrong box being detected as positive from that grid cell.
- We see that YOLO generalizes well to artwork as well as images in the wild, with Fig. 2(b) showing an example.
- YOLO performs admirably well on the dense urban KITTI dataset (Fig. 2(c)), running on videos at 50-70 FPS on a NVIDIA GeForce GTX 760.

[1]Paper reported only the overall mAP that too for VOC 2012 test set whereas our results are on VOC 2012 validation set

7 Comparison Table

S.No.	Work proposed	Work upto mid-semester	New work for final evaluation
1	Reproducing reported results on PASCAL VOC dataset	Initial set up of a smaller GPU	Got comparable results on the Tiny-YOLO model (Fast YOLO) on VOC dataset
2	Parameter study on grid size and number of bounding boxes per grid cell	Limited training of a scaled down model (Tiny-YOLO). Partial training for parameter studies, results not reported	Experimented for various grid sizes (5,7,9) and bounding boxes combinations (2,4,6) and got improved mAP from S=9
3	Fine-tune on datasets such as KITTI and ETHZ Multi-person tracking dataset [3]	-	Fine-tuned YOLO on KITTI and obtained good qualitative results
4	Demoing a real-time detection	-	Presented real time detection on videos obtained from Youtube, at 58 FPS, using both VOC as well as KITTI trained networks
5	Decreasing localization error through literature review	-	-

References

[1] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. *arXiv preprint arXiv:1605.06409*, 2016.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[3] A. Ess, B. Leibe, K. Schindler, , and L. van Gool. A mobile vision system for robust multi-person tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*. IEEE Press, June 2008.

[4] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[6] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[7] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[9] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park. Pvanet: Deep but lightweight neural networks for real-time object detection. *arXiv preprint arXiv:1608.08021*, 2016.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. Ssd: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*, 2015.

[12] J. Redmon. Darknet: Open source neural networks in c. `http://pjreddie.com/darknet/`, 2013–2016.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[14] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[15] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.