Object Detection – Amazon Robotics

Bhakti Sharma, Sophia Abraham, Ying Qiu

In this project, our three team members will work on three different methods to detect the objects in the plastic tote. By implementing three different approaches, we can either perform a comparison and examine the optimal solution among these three methods combined with fine tuning for improved performance or create an ensemble based on the strengths of these methods to produce our own approach.

Method 1 (Sophia Abraham): Mask-RCNN and YOLOv2 [2]

This method is not part of a published paper but was a method that was utilized for a hackathon which required the development of an object detection algorithm to detect objects, including occluded ones with less visibility in the frame. The method combines both YOLOv2 (object detection) and Mask-RCNN for segmenting which mitigates the limitations of both methodologies. The code is available on GitHub and can be retrained utilizing our class data. I would like to play with this concept and attempt different variations of training data that could be used with YOLO, (v1, v2, or v3), in order to obtain rich features that can aid in the development of a strong approach.

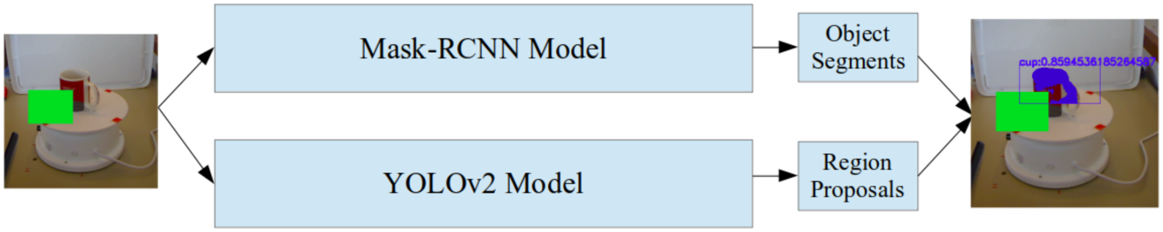

Fig.1 Pipeline of methodology 1.1 [2].

Experiments with data:

The variations for experiments with the data include:

- Directly using Mask-RCNN
- Segmenting images with FLoRIN [3]

- Synthesis for Instance Detection [4] – This one generates scenes with multiple objects and annotation files for an object detector (including occlusions, rotations, etc.)

Steps:

Since there are approximately four weeks up to the mid semester report, I roughly approximated the tasks based on week in order to progress towards a solution. What is outlined may also not work in principal from the very start, thus further examination and reading will be required to confirm whether the method is viable for this purpose. The general steps are outlined below (which may be a bit ambitious because I am unaware of the time requirements for these procedures xD) :

*****Naive plan right now! Going to explore and see whether one method may work better than others by reading implementation papers and code, in which case will just work on one and refine over the weeks** ***

- Week 1 – Train Mask R-CNN for individual 10 objects from data set/ run with YOLOv2 and observe
- Week 2 – Train Mask R-CNN and run with YOLOv3
- Week 3 – Play with different parameters in FLoRIN to examine output segmentations and refine
- Week 4 – Combine segmentation results from FLoRIN with YOLOv2
- Week 5 – Combine segmentation results from FLoRIN with YOLOv3
- Week 6 – Generate data based on [4] and combine with YOLOv2/Week 7 – YOLOv3
- Week 8 – Compare results with my amazing team mates and ensemble A MASTERPIECE

For the mid-semester report the output from **one** of the methods will be documented and reported. Depending on the results obtained by Bhakti and Ying, we can reform our battle plan accordingly if something seems to be working quite well!

Method 2 (Bhakti Sharma): YOLOv3 for small object detection [5]

The most salient feature of v3 is that it makes detections at three different scales. The detection is done by applying 1 x 1 detection kernels on feature maps of three different sizes at three different places in the network. Detections at different layers helps address the issue of detecting small objects, which was one issue with YOLO v2. The upsampled layers concatenated with the previous layers help preserve the fine grained features which help in detecting small objects. Few papers have worked on modifying YOLO v3 based on the oriented bounding box (OBB) for object detection in remote images [6]. Including this will help in detecting small objects and the ones with different orientations for our project. [7]

Experiments with data :

Training the model with the individual objects and testing the yolov3 for detection of small objects first. Then including the oriented bounding box detection to the model and testing the tote bag cluttered objects.

Steps :

- Build the base model focusing on small objects detection with a subset of the dataset
- Work on the detection of OBB in the model and check if it improves the accuracy
- Test against the tote bag objects and work spirally to improve the model with the complete dataset.

Method 3 (Ying Qiu) : SSD-single shot multibox detector [1]

SSD is a method for detecting objects in images using a single deep neural network. Without resampling pixels and features for bounding box hypotheses, SSD achieved the improvement in speed as well as accuracy. This detector uses a small convolutional filter to predict object categories and offsets in bounding box locations, and separate filters for different aspect ratio detections (as shown in Fig.1), and then applies these filters to multiple feature maps to perform detection at multiple scales. By implementing these modifications, it can handle relatively low-resolution input images and detect small objects with various aspect ratios.

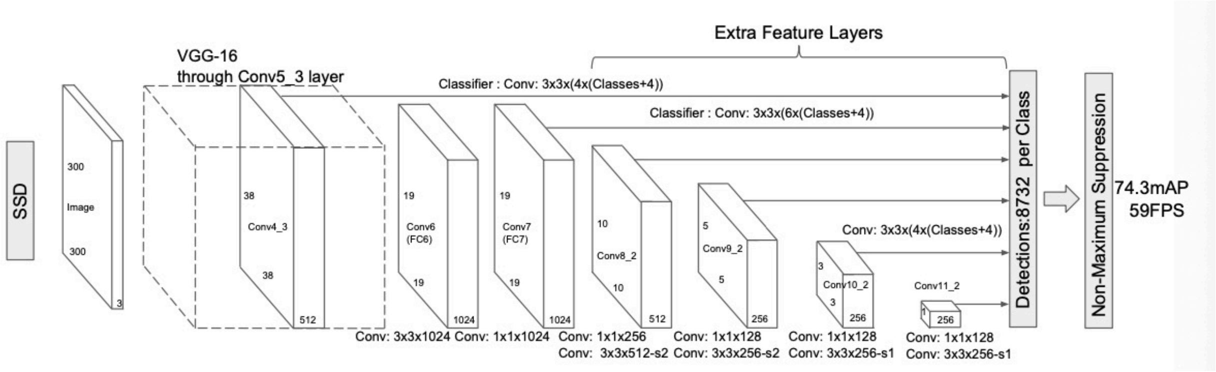

Fig.2 The extra feature layers at the end of a base network predict the offsets to bounding boxes of different scales and aspect ratios and their confidences [1].

SSD is robust to small objects with different aspect ratios. We Probably could treat the object with occlusion as a special type of small object. By implementing this approach, hope we can find out the answers to the following two questions in the semester project:

Question 2. How much occlusion can be present on a single object yet still result in a positive high confidence match?

Question 4. What is the robustness of the proposed methods to varying illumination, scale, rotation and selected properties o sensors?

Experiments with data

Training the model step by step with a small subset of pics, which are available to us and then do the inference. Next, extending the training and testing to a large subset for generalization. For consistency, we may use the same input images in order to do the following comparison among different methods

Steps:

Step one: build the base network

Step two: build the extra feature layers

Step three: inference with NMS

References

[1] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In *European conference on computer vision*, pp. 21-37. Springer, Cham, 2016.

[2] https://github.com/jatinmandav/Occluded-Object-Detection

[3] https://github.com/jeffkinnison/florin

[4] Dwibedi, Debidatta, Ishan Misra, and Martial Hebert. "Cut, paste and learn: Surprisingly easy synthesis for instance detection." *Proceedings of the IEEE International Conference on Computer Vision*. 2017.

[5] https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b

[6] J. Lei, C. Gao, J. Hu, C. Gao, and N. Sang, "Orientation adaptive yolov3 for object detection in remote sensing images," in Pattern Recognition and Computer Vision (Z. Lin, L. Wang, J. Yang, G. Shi, T. Tan, N. Zheng, X. Chen, and Y. Zhang, eds.), (Cham), pp. 586–597, Springer International Publishing, 2019.

[7] Zhang X, Zhu X. An Efficient and Scene-Adaptive Algorithm for Vehicle Detection in Aerial Images Using an Improved YOLOv3 Framework. ISPRS International Journal of Geo-Information. 2019; 8(11):483.