TOMATONET - A REALTIME DEEP LEARNING VISION SYSTEM FOR CLASSIFYING FRUITS / VEGETABLES

[1] D Pratush Charan, [2]Keerthi Sagi
[1]MTech Scholar, [2]Research Scholar
Department of Computer Science & Systems Engineering (A), Andhra University, Visakhapatnam

Abstract:Vegetable classification is a fairly new problem in computer vision (CV) especially, the ability to use CV in Kitchens is a relatively unchartered field but this problem does share the same characteristics with any CV problems.We present TomatoNet, a novel computer vision system for classifying fruits, vegetables, humans, pets, kitchen items. This vision system is an end to end trainable deep architecture for real time classification in a kitchen context. TOMATONET system reaches a real-time performance of 8-12 fps (frames per second) and runs the object classification and localization with ~20 ms per object in an individual video frame on an intel i5 8th gen cpu with no dedicated GPU. Performance speed and inference speeds greater than the yolov4 has been achieved. To achieve real time classification, we first apply a novel Region Proposal Network (RPN) and then apply a single shot CNN detector to the full image (SSD/YOLO) to the output of the RPN Network. A comparison has been provided against the state of the art algorithms.

Index Terms:CNN, RPN, YOLO, Deep Learning, Fruit Rotting, Image Recognition

I. INTRODUCTION

CNN's have been well known to give good results for image related learning and classification tasks. For ex Faster RCNN [1] and other related algorithms, detect small objects well.Howeverthese two stage algorithms fail to do real-time detection with their inherent two step architecture. We propose a vision and classification system / model for a robot to operate in home environment along with humans. (c**o**llaborative ro**bot**s, cobots). In order to validate the performance of the proposed network architecture in terms of speed and accuracy, a comparison will be made against state-of-the-art methods using the mean Average Precision (mAP) metric.

A vision system should be clearly able to identify the following

Fruit / Vegetable Classification

Fruit classification via computer vision technology uses fruit texture, color and shape for visual feature evaluation[2]. In this thesis, efforts have been mainly emphasized on work for the algorithm development of fruit vegetable classification and individual rot detection.

Detecting Humans, Pets, Kitchen Items (Safe collaborative Environment)

The proposed vision system should also trigger alarms if humans, pets enter the field of view of the camera. So we train human, pet presence detection by labeling and classifying living objects as well in our network. In Fig shows how the final system is implemented, a separate classifier using TomatoNet is trained for this purpose. Most cobot (**co**llaborative ro**bot**)safety measures rely on lightweight construction materials, rounded edges, and limitationsonspeed and force. But, in our case we need our vision system as an additional step to detect certain objects and effectively respond if the humans, pets or rodents enter the field of view of the camera.

Fruit / Vegetable Rotting - Visual Characteristics

A fruit or a vegetable during a decay process appears in gradual changes, e.g., the growth of dark spots from oxygenizing and shrinkage [2] due to the loss of contained water.A single network would have been convenient to identify all the rots in all the fruits. But it is not possible. Each fruit / vegetable decays it its own way and the visual

characteristics are remarkably different. So one single network could not be generated to identify the rots in the foods.

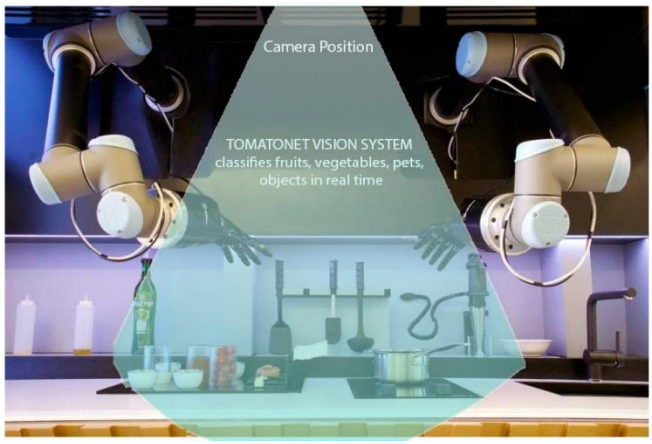


Fig 1. TomatoNet Vision System Practical Setup

II. TOMATONET DESIGN

The proposed TOMATONETis a hierarchical, two staged,deep learning model.It consists of an RPN (Region Proposal Network) and a Single shot detector to improve the speed of the network/system/ algorithm.For the system to work in tandem with a robotic arm in a robotic kitchen, our algorithm needs to be working in real time (24frames per second). Therefore, for an acceptable solution, processing the frames at at-least ~10-15fps classification speed is mandatory. Objects need to be localized close to 20-30ms per object.

The idea is to input the image frame in question into a novel region proposal network and obtain the important areas of the image ie foreground and store them in a region map. Then this region map is sent into yolo / ssd object detector for object bounding box and class confidence scores. Twostages of TomatoNet are mentioned below.

1. Region Proposal Network (new proposition of this paper)
2. SSD-YOLO (Apply existing single shot detector to the output of the above RPN)

Region Proposal Network Algorithm (Input from Camera)

1. Downsampleinput frame into ¼ of the original size.
2. Encode / transcode image into ICC profile.
3. Use trivial and fast foreground extraction mask. (Background is static)
4. Segment the extracted foreground into separate regions.
5. Create a Region Map.

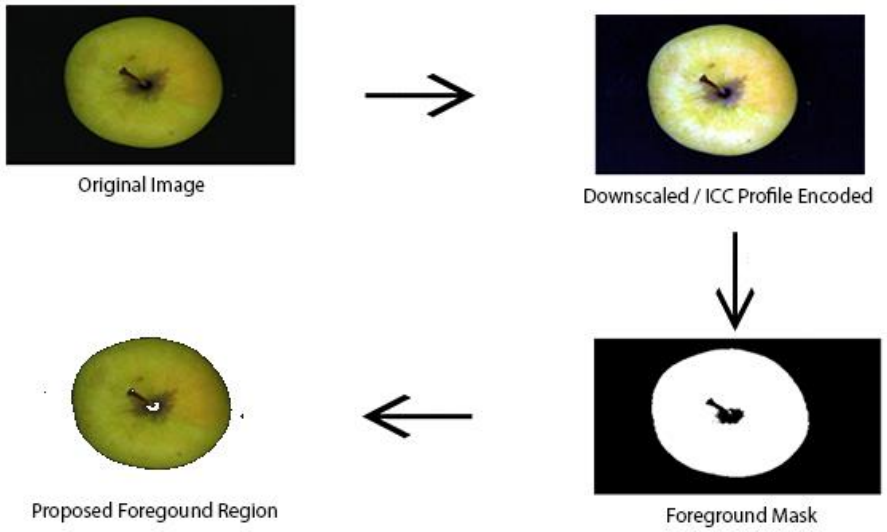


Fig 2. Steps in Region Proposal Network

The Region Proposal Network gives the region Map to the SSD-Yolo algorithmThe RPN is a very low cost operation that takes up minimal CPU operations and extract the foreground regions.

Applying SSD-YOLO Object Detector on the Region Map

1. Process the individual regions from above Step 1. Input the Region Map from the Region Proposal Network to SSD Yolo object detector.
2. **Feature Extraction** - We use a C based implementation of Darknet-53 for feature extraction. The network uses successive 3x3 and 1x 1 convolutional layers. It has 53 convolutional layers. Darknet-53 achieves better performance than, ResNet-101 as well as ResNet-152 [3].
3. **Class Prediction** - Each region from the RPN output is send to the object detector and the object detector predicts the classes with confidence scores and the bounding box. Softmax approach is CPU intensive and therefore, is eliminated in our approach for the speed advantage and real-time performance. In TomatoNet, we use a multi-label approach for classifying objects. That is even if an object has been classified as a specific class. It might still be classified into another broad or specific. We get results with many overlapping labels for example Man and Person. We believe that a multi-label approach better models the data and gives context to the detection results.

Backbone	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
ResNet-101[3]	77.1	93.7	19.7	1039	53
ResNet-152 [3]	77.6	93.8	29.4	1090	37
Darknet-53	77.2	93.8	18.7	1457	78

Table 1. Comparison of feature extractor backbone networks

III. TRAINING

For detecting fruits / vegetables, Kaggle 360 Fruit Data set was used. This is a dataset with 90483 images of 131 fruits and vegetables. Detecting Humans - Inria Person Dataset / MS COCO dataset was used. Pedestrian and person data in different poses is obtained from real life scenarios. Therefore pose bias is eliminated.

IV. DEPLOYMENT

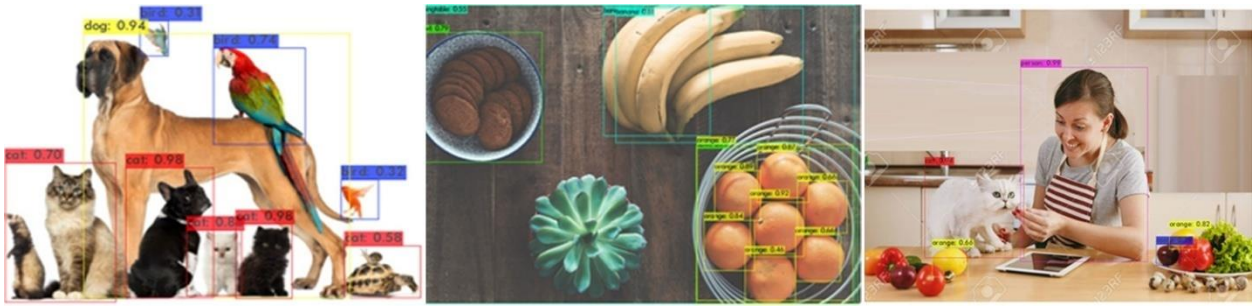
TOMATONET was deployed on anIntel i5 8th generation cpu, 32gb ram and on windows 10. OpenCV [4], Cuda were used to apply the most common image processing libraries and filters for our image frames. For ex resize, transcode, encode etc operations. Darknet-53 libraries in c,c++ environment were compiled for our windows environment using the vcpkg package manager.

V. RESULTS

TomatoNet has delivered very good real-time performance better than other detectors, when trained only for kitchen items. The usage of the RPN before the single shot detector has improved the real time classification speeds. The metrics were average precision, total inference speeds and speed with which single object in the frame is localized. Precision remained the same as we use the same backbone as our SSD /Yolo.

	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++	ResNet-101-C4 [3]	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN by G-RMI	Inception-ResNet-v2	34.7	55.5	36.7	13.5	38.1	52.0
One-stage methods							
SSD513	ResNet-101-SSD [3]	31.2	50.4	33.3	10.2	34.5	49.8
RetinaNet	ResNet-101-FPN [3]	39.1	59.1	42.3	21.8	42.7	50.2
SSD-YOLO 608 x 608 [6]	Darknet-53 [5]	33.0	57.9	34.4	18.3	35.4	41.9
TomatoNet	Darknet-53[5]	33.0	57.9	34.4	18.3	35.4	41.9

Table 2. Tomato Net doesn't show any increase in Precision when compared to SSD-YOLO since the backbone is virtually the same.



27ms per object 21 ms per object 29 ms per object

Fig 3. Inference Time per Object using TomatoNet

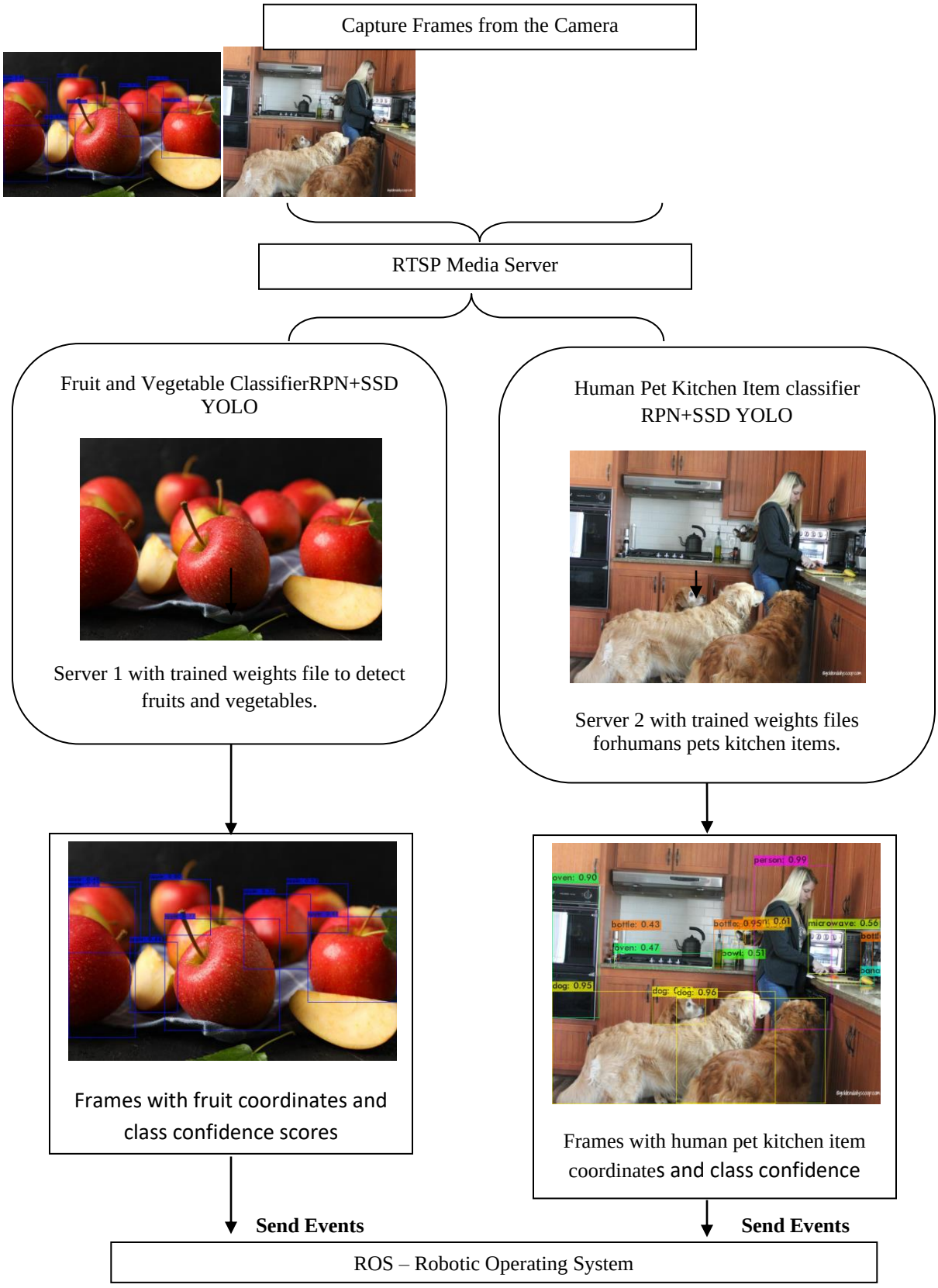


Fig 4. Deplyment of the TomatoNet vision System

The below graph indicates that the inference time has decreased after introducing the Region Proposal Network. The metrics for average performance of all six fruit species are calculated to evaluate the overall performance of YOLO [7] for the task of classifications.

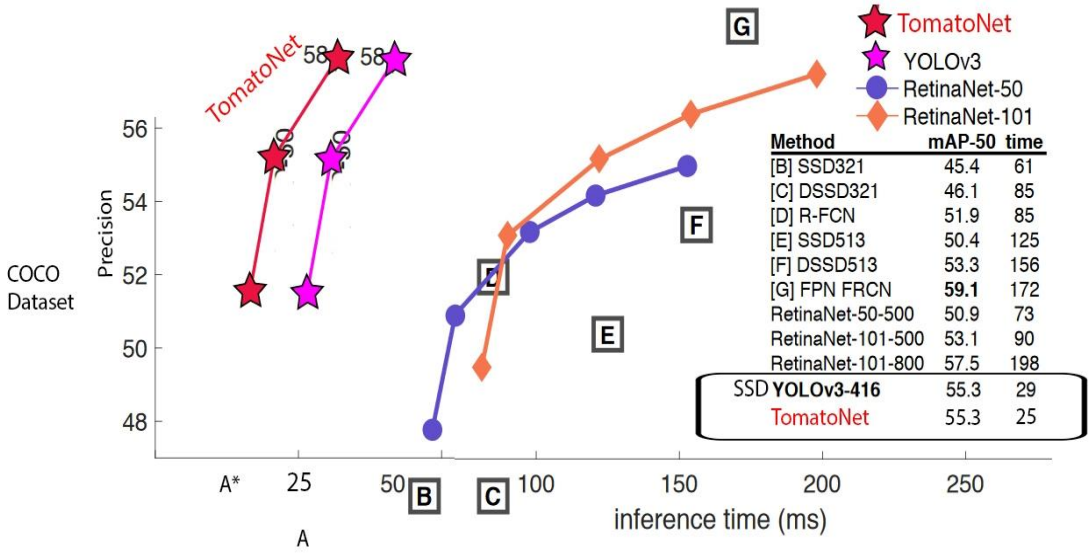


Fig 5 Inference time for TomatoNet (Red)

The classification was achieved on intel i5 8th generation based system with 32 gb ram.

VI. CONCLUSION

In this paper we proposed a real time vision system and developing classification models for a Robotic Kitchen.We construct a hierarchical deep learning model, where the regions of interest are cropped from the source images via a region proposal network and then fed into a single shot object detector like SSD/YOLO [6];the following major problems were addressed in the paper.

Problem Statement	Results
Real Time Classification	8-12fps was achieved with ~20ms object localization speed.
Fruits and Vegetables Classification	55.3 map achieved.
Achieve Cobot Functionality	Human, Pets, Kitchen Item detection.

Table 6. Achieved Results

VII.REFERENCES

[1]Girshick, R. (2015). Fast R-CNN. *International Conference on Computer Vision*, 1440-1448
Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 580–587.

[2] Barrett, D. M., Beaulieu, J. C., & Shewfelt, R. (2010). Color, Flavor, Texture, and Nutritional Quality of Fresh-Cut Fruits and Vegetables: Desirable Levels, Instrumental and Sensory Measurement, and the Effects of Processing. *Critical Reviews in Food Science and Nutrition,, 50*, 369-389.

[3] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition, 770-778.

[4]Kaehler, A., & Bradski, G. (2016). *Learning OpenCV 3: Computer Vision in C++ with theOpenCV Library*. Newton: O'Reilly Media, ISBN 1491937998.

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

[6] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 6517–6525. IEEE, 2017.

[7] J. Redmon and A. Farhadi. Yolov3: An incremental improve-ment. ArXiv, 2018.