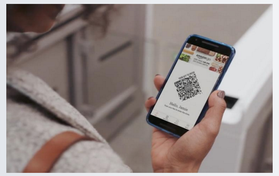
DETECTION OF HAND GRASPING TASKS FOR "GRAB AND GO" GROCERIES

Xianlei Qiu, Shuying Zhang
Stanford University

BACKGROUND

- Recent advancements in computer vision have made the "grab-and-go" grocery stores like Amazon Go a reality. Shoppers can simply walk in the store, grab the items and walk out, without waiting in a long line for checkout. The automation of the checkout process relies heavily on computer vision systems capable of tracking items grabbed by a certain customer.
- Hands detection is a key component in such an intelligent system. Customers' behaviors during shopping are complex and hard to predict. For instance, it is common for a shopper to grab items from shelf and later put them back. To further understand customers' behaviors, it is extremely important to detect and track their hands in video streams.
- Many computer vision based methods have been proposed to detect hands in an image[1][2]. However, most of these computer vision based approaches rely on detecting skin tone pixel, which could be ambiguous if faces or other skin regions are also present in the image.
- The YOLO (You Only Look Once) network[3] is a state-of-the-art object detection system, providing accurate and fast prediction in real time. It achieved the mAP on VOC 2007 of 78.6% and a mAP of 48.1% on COCO test-dev. It runs a single convolutional network on the image and thresholds the resulting detections by the model's confidence. We adopted the YOLO network for our hand detection system because it is accuracy and it provides real-time object detection.



Problem Statement

We are aimed to design and implement a fast and accurate hand detector which is capable of detecting hands in video streams.

CLASSIFICATION METHODS

Detection Method

An image is divided into an SxS grid. Each grid cell predicts B bounding boxes and confidence scores for those boxes. The confidence score for each grid is calculated as the Intersection over Union (IoU) between predicted box and The ground truth box: $\Pr(object) * IoU_{pred}^{truth}$.

Each grid cell also predicts C conditional class probabilities, $\Pr(Class_i | Object)$. At test time, the prediction is made as: we multiply the conditional class probability and the confidence score to get the class-specific confidence score for each box.

$$\Pr(Class_i | Object) * \Pr(Object) * IoU_{pred}^{truth} = \Pr(Class_i) * IoU_{pred}^{truth}$$

These predictions are encoded as an S × S × (B * 5 + C) tensor. For our model, we used S = 13, B = 2. PASCAL VOC has 20 labelled classes so C = 20.

Architecture

Neural Network Architecture

We used the tiny version of the YOLO9000 network (Figure 2), which is based on the Darknet framework [4].

Yolo vs. Yolo9000

The Yolo9000 network is the upgraded version of Yolo network. The original Yolo network suffers from a significant number of localization errors comparing to Fast R-CNN. Yolo9000 made several improvements including batch normalization and high-resolution classifiers. It is more accurate and faster.

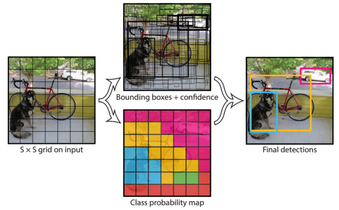


Figure 1. Principle of Yolo Detection Method

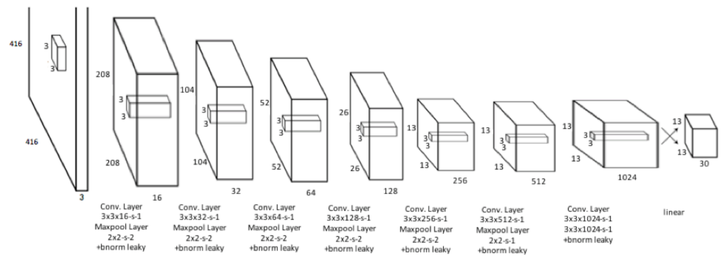


Figure 2. Tiny Yolo Voc Hands Architecture

DATASETS

Egohand Dataset

We first used the EgoHands dataset [5] to train our model. The dataset consists of a total of 48 Google Glass videos. Each video has 100 frames that are annotated with ground-truth boxes of hands (Figure 3). For each video, We selected the first 70 frames for training and the last 30 frames for testing. Therefore, the total training dataset consists of 3360 images and the test set consists of 1440 images.

Fridge Video Stream Dataset

We were also provided with 10 videos, which record customers' behavior of removing food items from a self-serve fridge. We randomly selected 50 frames for each video and labeled the ground-truth boxes of **hands** and **arms** using the labelImg labeling tool and generated 500 annotated frames for training and testing.

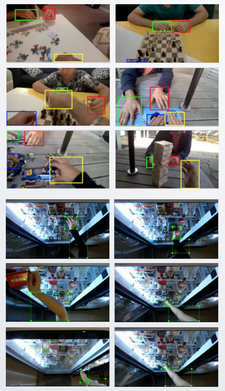


Figure 3. Sample Dataset with Ground Truth Bounding Box. top: Egohand. Bottome: Video Stream Dataset

Figure 4. Visualization of the Activation of the First Conv Layer.

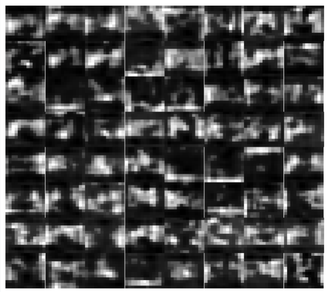


Figure 5. Visualization of the Most-activated 64 Neurons of the Sixth Conv Layer

RESULTS

Training Process:

We fine-tuned our model based on a **pre-trained** tiny-yolo-voc weights. The training process can be divided into 2 stages:

- learning rate: 1e-5, batch size: 16, for 10 epochs to quickly reduce loss.
- learning rate: 1e-6, batch size: 64, for 20 epochs to fine tune model.

The final loss is around 3.0.

Accuracy (IOU = 0.5):

Training mAP = 91.54 %

Test mAp = 90.63 %

Figure 6. Sample Test Result.

Speed:

CPU: 4.815 frames/s

GPU: 30.00 frames/s (on NVIDIA Tesla K80)

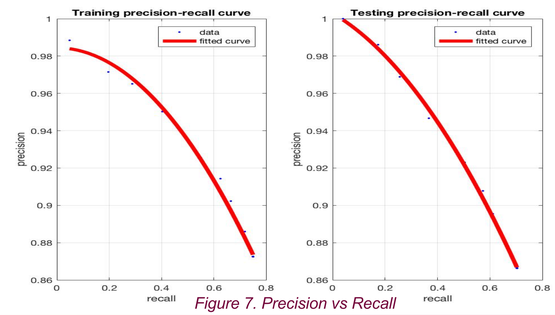


Figure 7. Precision vs Recall

CONCLUSIONS

- The Yolo V2 model was able to deliver promising hand detection results on the Egohand data. It achieved both high accuracy and high speed. Our best model was able to yield 90.63% mAP on the test set and a blazingly fast speed of 30 FPS on GPU , achieving the real-time requirement.
- Demo video [6].

FUTURE WORKS

- Next step we will continue fine-tuning and training our framework on the fridge video stream dataset.
- We also would like to use this framework to pinpoint and detect grasping events in video streams from the fridge video stream dataset.

REFERENCES

1. Betancourt, Alejandro, et al. "A sequential classifier for hand detection in the framework of egocentric vision." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2014.
2. Kölsch, Mathias, and Matthew Turk. "Robust Hand Detection." FGR. 2004.
3. Redmon, Joseph, and Ali Farhadi. "YOLO9000: Better, Faster, Stronger." arXiv preprint arXiv:1612.08242(2016).
4. Darknet tensorflow library: https://github.com/thtrieu/darkflow
5. EgoHands Dataset: http://vision.soic.indiana.edu/projects/egohands/
6. https://goo.gl/04fXni