

Individual lab report #10

4/6/2017

Jin Zhu
Team E

Teammates: Michael Beck, Akshay Bhagat, Matt Lauer, Leo Lu, Jin Zhu

1. Individual Progress

1.1 Fix confusion matrix

Confusion matrix would be a useful tool when we can choose to pick easily confused items in separate bins. We will know the list of items in each bin through JSON file, and we can choose which bins to populate which items, avoiding confusing items in same bin would help the classification result.

The confusion matrix generated using the classification results from FCN and FasterRCNN trained on new items wasn't correctly indicating which items were confused. Thus, one of my tasks this week was to find out the bug in confusion matrix.

FCN pixel wise classification result confusion matrix is shown in Figure 1. The numbers in cells along the diagonal indicates the number of correctly classified pixels. This matrix was normalized across each row. The confusion matrix had 3 yellow rows. Since yellow here is set to representing high value, and high values off the diagonal line means easily confused, this confusion matrix is indicating misclassification across all classes. However, when I check the ground truth and classification result, these 3 items turned up to be correctly classified. Also, the values along each of the 3 rows turned out to be zeros, and that's why they were all yellow color after normalization along row.

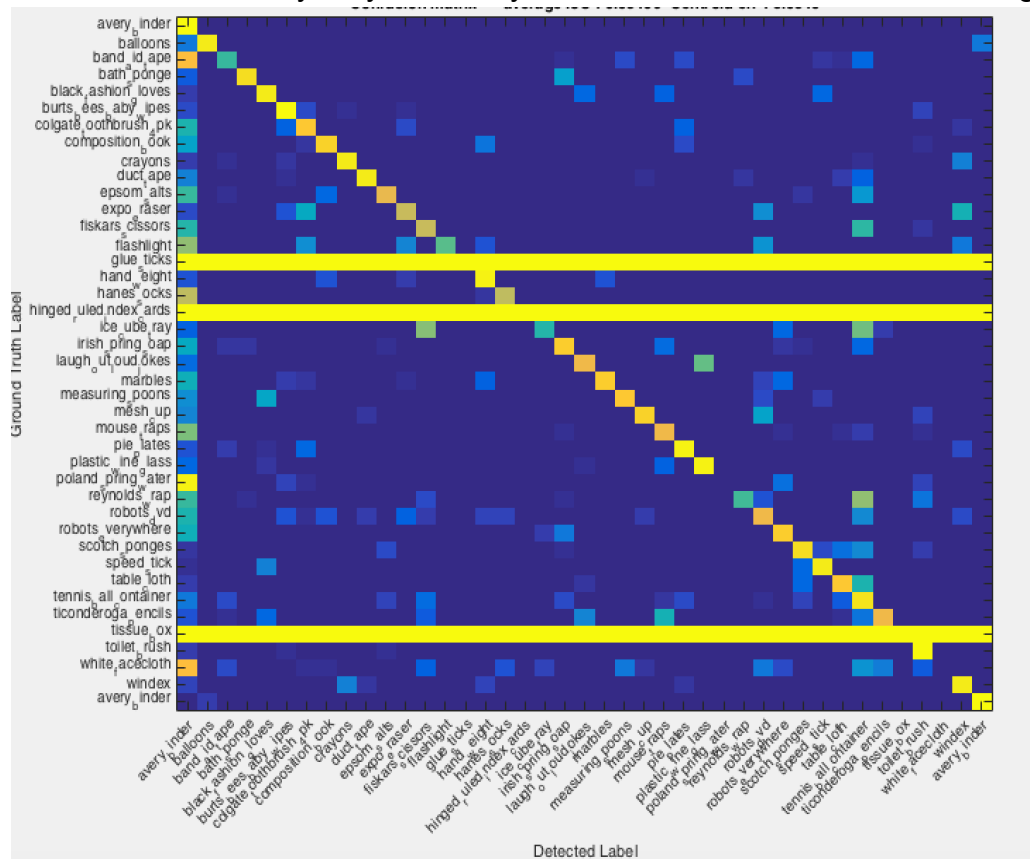


Figure 1. Confusion matrix before fix

The mistake turned out to be missing the background class, and the different convention between python and MATLAB. All the items are having high values for the first class, “avery binder”. It turned out that background class was missing from the code, and background is the class, which is easily confused with other class. The second mistake was caused by the convention of python and MATLAB. When we randomly select the testing images, there were 3 items not showing up in the testing set, this caused some rows had all zero values. However, these 3 items were not the 3 items with yellow cells along the row. The ground truth bounding-box was turned to pixel wise labeling using python file, and this result was combined with pixel wise classification result for MATLAB to generate confusion matrix. Data parsed using Python file had index zero, and data in MATLAB had index starting from 1. This caused the mismatch of confusion matrix result and the real result for the 3 missing items. The fixed confusion matrix is in Figure 2.

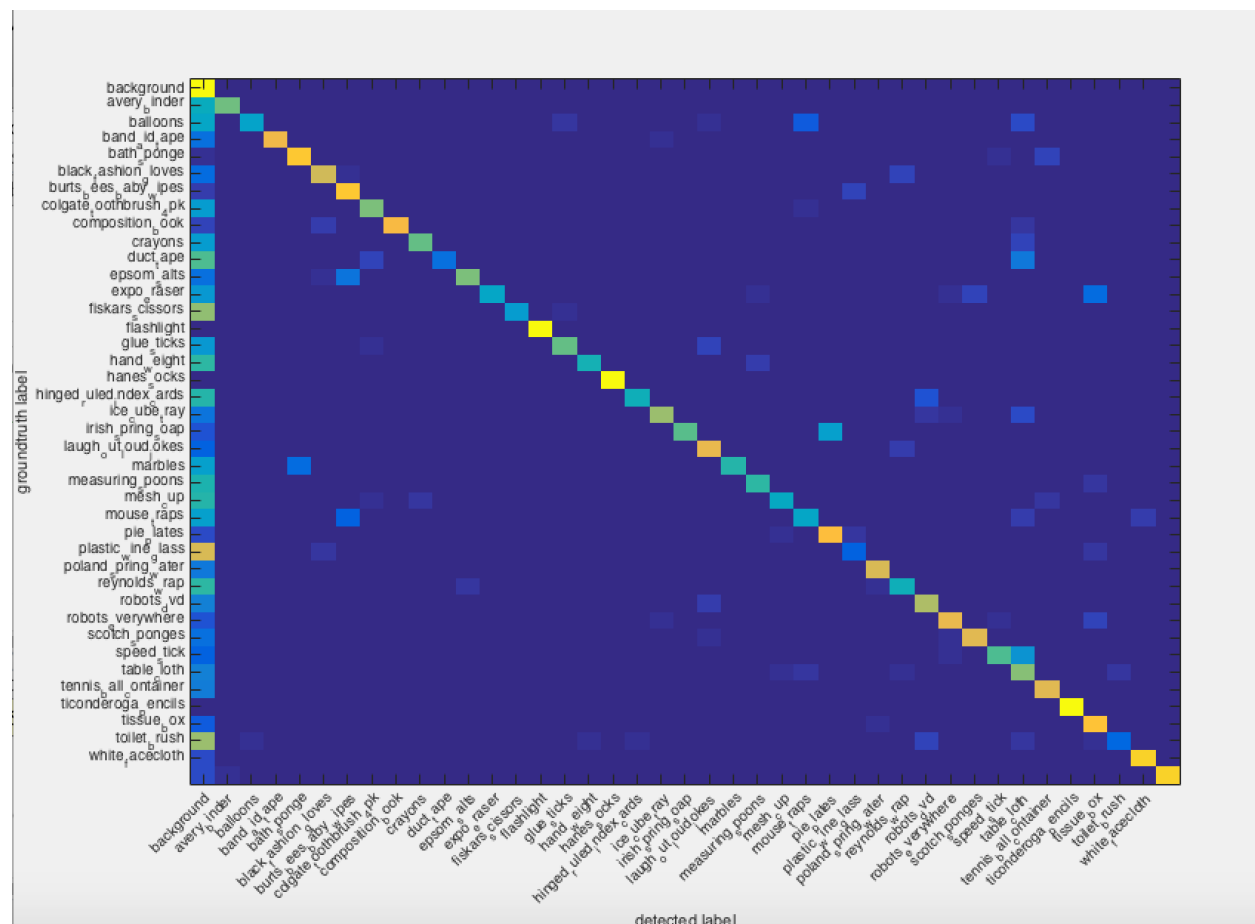


Figure 2. Confusion matrix after fix. The missing items are 'flashlight' 'hanes_socks' and 'ticonderoga_pencils'

1.2 Background subtraction

One possible solution for recognizing unknown items is to train an SVM using data gathered during the 30 minutes before competition time. The FCN output feature for each pixel belonging to the item will be used to train the SVM for the corresponding item class, as shown in Figure 3. Since we won't have time to label images to give ground truth of object and background within the 30 minutes, we will need a way to separate object and background.

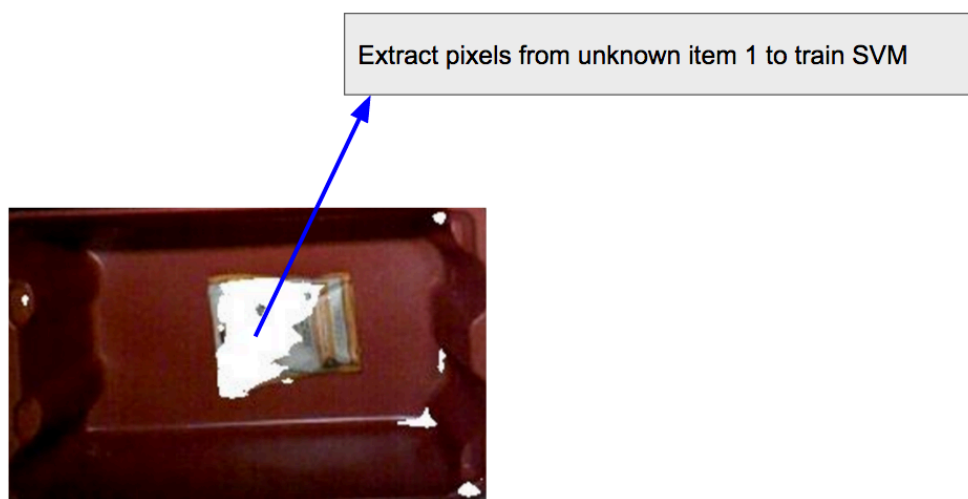


Figure 3. Proposed method to do automatic background subtraction

Since the FCN was trained on the 40 items plus background, I tried to use the FCN result to separate background and non-background, which would be the unknown object. One of the results was shown in Figure 3, where white areas were classified as non-background, and the colored area were recognized as background. This result showed that FCN would be confused when given a new item, and classify part of the background as object, and part of the object as background.

The current FCN give a score for each pixel for each of the 41 classes. Out of the 41 scores, the class with the maximum score would be the class of the pixel. To generate result similar to Figure 3, if the class with the maximum score was not background class, it would be considered as unknown object. Thus, one way to improve the classification result for background and non-background would be using a different way to assign a pixel to background or object. The way I tried this week was to find a threshold, if one pixel's background class score is above the threshold, that pixel would be classified as background. Otherwise, it is object.

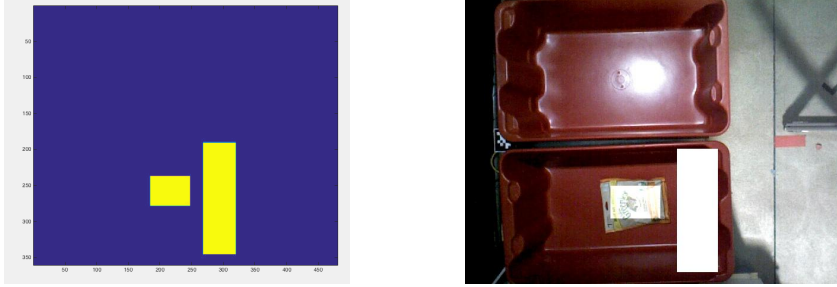


Figure 4. Regions to select the 200 random pixels for object and background.

The way to select threshold was to plot out the frequency that background and object receive different scores for the background class. Assuming that the background pixel is assigned a score x for the probability of being predicted as “background class”, and object is assigned a score y for being predicted as “background class”. If x is always larger than a value, and y is always smaller than a value, then the threshold can be found. Figure 4 shows the regions to select the 200 random pixels for object and background for generating the histogram in Figure 5.

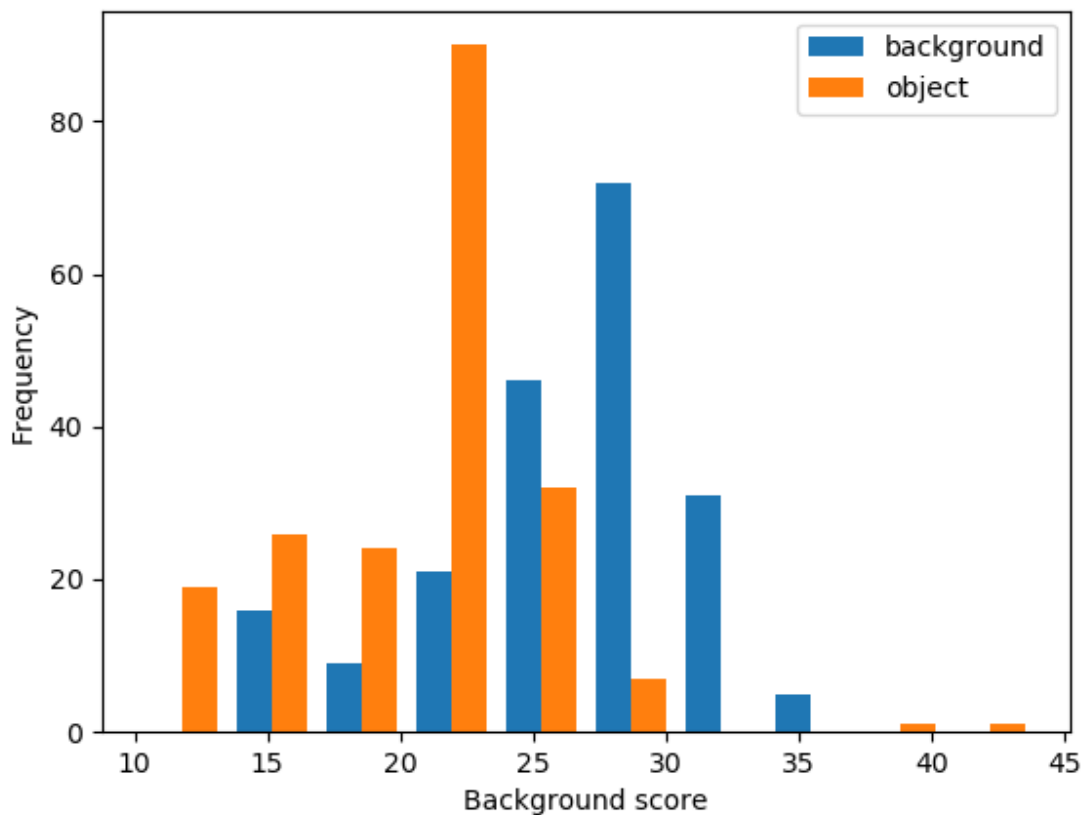


Figure 5. Histogram showing the frequency of background and object pixels receiving different background class scores

According to Figure 5, there were a lot of common areas between object and background for their scores of being predicted as background. Thus, I might need to find some other ways that's more reliable for separating background from the object. One way I will try next week would be using the depth data for segmenting out the background. The depth method might also have issues when the object is too flat.

2. Challenges and problems

2.1 Lighting

Shadow and reflection due to lighting condition was one of the reasons for misclassification of items. Figure 6 shows the shadow in the image captured by Asus. Image was the output from Leo and Akshay's integrated vision pipeline. I will look into methods to reduce reflection and improve lighting condition next week.



Figure 6. Shadow in the image captured by Asus. Image was the output from Leo and Akshay's integrated vision pipeline.

2.2 Background color selection

Another challenge we are facing now is to select appropriate background color for the bins. Figure 7 showed a comparison between classification result on different backgrounds. Sometimes black background gives better result, but some times the white background gives better result. Since the current vision Caffe model was trained using Kinect on red tote, it's hard predict which background color is better for Asus using the new shelf. I need to look into that next week too, before we finish building the new shelf.



Figure 7. Confidence score given by FasterRCNN on different backgrounds for the same item

3. Teamwork

This week, Michael focused on shelf design, path constraints for the arm, system integration and troubleshooting. Leo worked on the one click launcher and testing the integrated vision pipeline with JSON file. Matt worked on planning and the controller for slider. Akshay worked on IK server for arm and checked if poses were reachable. Leo and Sharon helped me with background subtraction task. Michael and Sharon also helped me with figuring out what went wrong with confusion matrix.

4. Plan

Last Saturday we listed out the tasks to be done for SVE and competition based on our current progress, as shown in Figure 8. Each week tasks will be assigned to individual task holders and finished tasks will be reviewed by the teammates.

I would keep working on finding a way to do background subtraction, experiment with lighting condition and background color for the shelf.

A	B	C	D
			Task Owner Week April 2nd
Competition	Travel+shipping	Plane tickets	Leo, Sharon
		Air Freight availability	
	Unknown items	Random Forest	
		Get a scale	
		Item placement list generation	
		Item masking	Jin
	Stowage	Scale vs Jumble items	
		Tote camera : get extra Asus	
	30 min Unknown item	Practice training	
		Talk about schedule with intern companies	
	Spare parts		
SVE	Shelf	1.1 Procure Aluminum	MB
		1.2 Water jet	MB
		1.3 Weld	
		1.4 Assemble 80/20	
		1.5 Mounting	MB
		1.6 Actuation for drawers	
	Planning	2.1 Slider	ML
		2.2 Constraints, prep poses	ML
		2.3 1-DOF gripper	
		2.4 E graph	
		2.5 Reaching goal: Integrate magnet	
	Vision	3.1 Shelf color, lighting, background	Jin
		3.2 Train for tote on the Asus : April 7th	
		3.3 Arm take images automatically	
		3.4 Data augmentation: try inside tote first	Sharon
		3.5 Multi apritag localization	
		3.5 E1 + E2 calibration (extrinsics for camera)	
		3.6 Perch integration	
		3.7 Amazon Turk?	
		3.8 How to merge FCN and Faster RCNN?	
		3.9 Fusion	Leo
		3.10 JSON	Leo
	Grasping	4.1 Priority list	
		4.2 grasping pose : create a list of potential weight	AB
		4.2 IK grasp pose + distance from normal	AB
		4.3 Item picking priority decision maker	
	System	5.1 Software diagrams/manuals	
		5.2 Hardware failure contingencies	

Figure 8. Tasks list