Individual lab report #9

3/23/2017

Jin Zhu
Team E

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

1. Individual Progress

My major task was to train the Faster RCNN model for the new 2017 items and compare the performance with FCN.

1.1 Faster RCNN training

Faster RCNN was trained using larger image size, since higher resolution provides more features for training. The training image size changed from ¼ to ⅓ of original Kinect image size.

The new item set contains a total of 40 classes of items. Data were split into 3 sets, training, validation and testing. The training contains 52 images, which is around 60% of the whole image set. Images are items in red tote. Validation set contains 14 images, which is 20% of the whole image set. Validation set contains images for items in red tote or bin with different background colors. Testing set contains 16 images, which is 20% of all the data. These images are randomly selected images containing items in red tote or bins with different background.

During training, both training set and validation set were used, so the final model was overfit to the training and validation images. Figure 1 shows the loss function value, which indicates how close the model prediction was to ground truth. The loss reaches a very small value and stayed there. This confirms that the model after 100000 iterations overfits to the training image, in another word, remembers the training images.
This model was used during progress review for item classification.

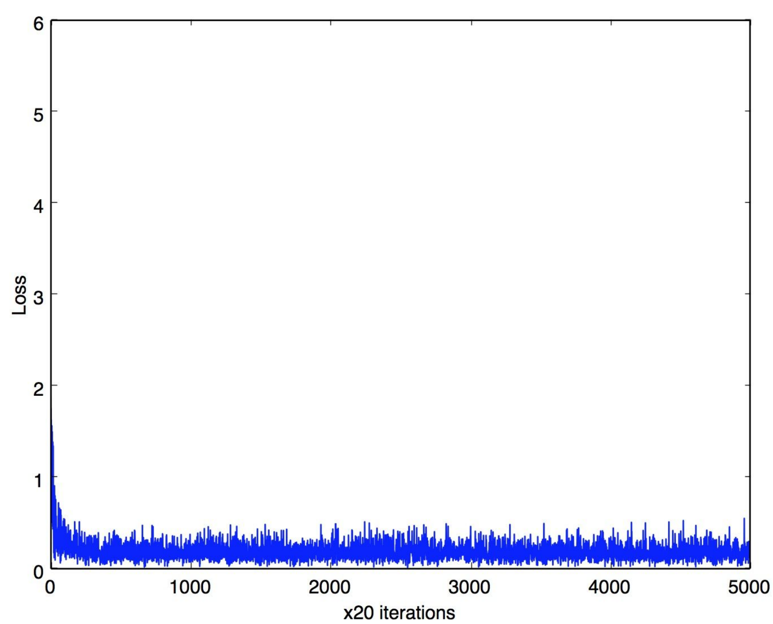

Figure 1. Training loss for Faster RCNN

Figure 2 shows a result of Faster RCNN prediction (red bounding box) compared with the ground truth labeling (green bounding box) using images from test set.

Figure 2. Faster RCNN prediction using testing set

1.2 Test trained model on Asus images

The RGB model was trained on Kinect, so it wasn't guaranteed to work on Asus. Thus, before integration, images for different bin configurations were captured using Asus and passed to both Faster RCNN and FCN for testing. Figure 3 is a comparison between FCN and Faster RCNN result using data from Asus camera. Figure 3(a) is the pixelwise labeling for FCN, and Figure 3(b) is the bounding box result for Faster RCNN. This is a result without knowing what are the 6 items in the bin, which is an information we will be given during competition and was also proven to help increase the accuracy.

Both Faster RCNN and FCN gave lower accuracy on Asus images, since the scale, resolution and the aspect ratio are all different for the two different cameras, and there weren't enough white background in the training set. After we get the new shelf, we will collect a large set of images and train on them, so that the model would predict more accurate results.

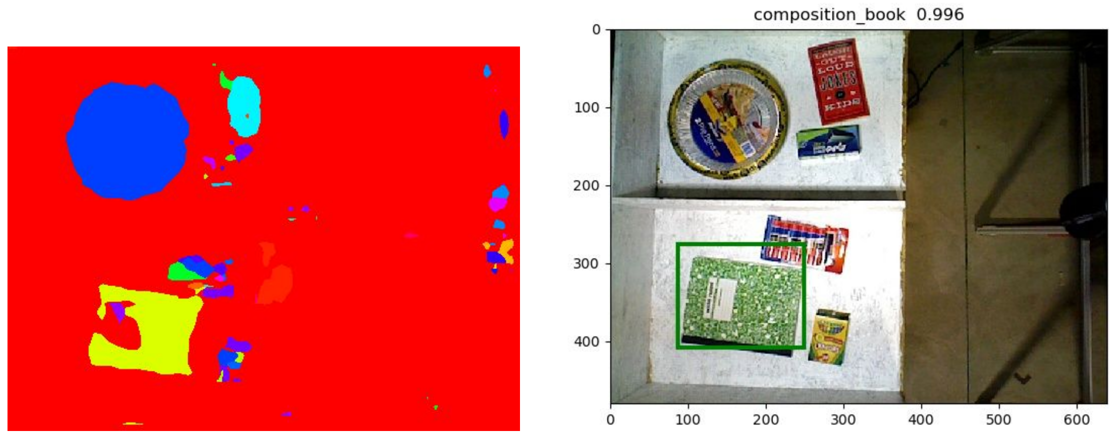

Figure 3. FCN and Faster RCNN result comparison

1.3 Familiarize with FCN and look for ways to extract feature
In order to utilize the RGB images of the unknown items that would be released 30 minutes before the competition, we decided to use an SVM to do a quick learning of these items.

I looked into the FCN architecture to find features that we can extract for SVM training. The architecture is shown in Figure 4. After we receive the input image of unknown items 30 min before competition, the images would be passed through FCN. At 3 stages inside FCN, feature vector for each pixel can be extracted at each stage and concatenated to one. Then the SVM can learn each class using feature vectors.

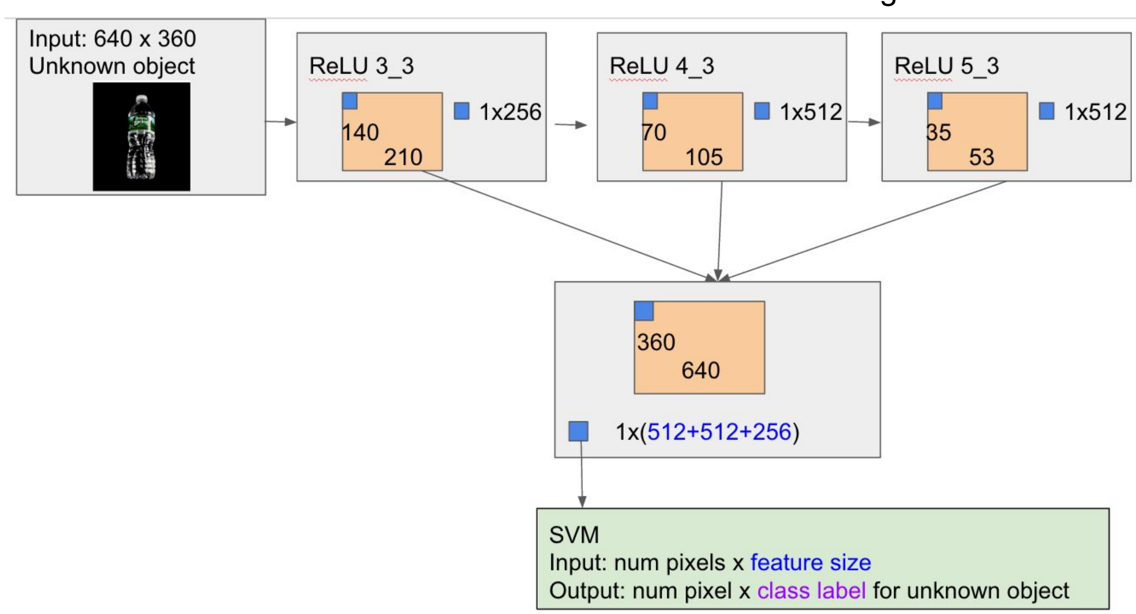

Figure 4. Process to extract feature vector for SVM training

2. Challenges and problems

2.1. Not enough time to do data labeling for intermediate stage

An ideal situation is that we will collect data and label them for the intermediate stage, wood shelf, before collecting images on the final aluminum shelf. However, labeling images is a very time consuming task, so we didn’t collect and train images for this temporary wood shelf.

Since the red tote would be used for picking during competition, but the current shelf wouldn’t be the same one we use during competition, we decided to collect training images on the red tote first. The model trained on RGB data from the red tote doesn’t work very well on the bin. This isn’t surprising for us, because the plan was taking separate training images on the red tote and the shelf to train models separately.

This challenge was alleviated by letting the system know the 6 -8 items in the bin, which eliminates some obvious false classification. This is close to the condition we would have during competition, since there would be a list of JSON files specifying list of items in one bin.

2.2 Learning curve for neural networks and familiarize with FCN

Reading the code and familiarizing with the neural network architecture was challenging for me. I’m currently taking the course that teaches neural network, learning the concepts in class and at the same time applying to this project is challenging.

When I’m not familiar with the architecture, small things such as setting image size and caffe model path can trick me up cost me extra time. For example, the first two times I trained on the new images for the new data set, the first model didn’t work because I didn’t tell the neural network new class names, the second time model didn’t work because I didn’t change the input image size.

3. Teamwork

Since last progress review, Matt worked on planning scene, preposition poses, and linear actuator. Michael worked on shelf and vacuum hardware. He also worked on magnets, which would be used for picking metallic objects. Leo worked on perch to estimate 6 dof pose for new items. Akshay, Sharon, Leo were mainly responsible for integration of FCN. Akshay also worked on camera calibration.
Sharon helped me with familiarizing with the FCN network architecture and think about how to extract feature vector for SVM training.

4. Plan

Next stage of this team would be improving the performance of each subsystem based on this minimum capable product.

My task for the next period would be keep working on the unknown item identification. I need to be able to extract the feature vectors, upsample the data from a small size to a larger one and concatenate feature vectors. Also, this Friday, Matt, Sharon and I will be working on integrating photo capturing on Asus with arm planning. This would be used to go to different poses of arm and take images when the new shelf is ready.