Recognition and Price Computation of Products from their Image

Team Members:

- Akshat - akshat@wisc.edu
- Kundan kumar - kkumar36@wisc.edu
- Swati Mishra - smishra33@wisc.edu

Overview:

The goal of our project is to improve the shopping experience by scanning products directly from the cart and automating the billing process; thus effectively avoiding long queues and reducing cashier overhead along with dynamically estimating total price based on the addition and removal of items. We will be using three algorithms for the detection of cart items: SIFT(Scale Invariant Feature Transform), YOLO (You Only Look Once) and Faster R-CNN (Region-Convolutional Neural Networks). As per the changes till midterm report, we have implemented SIFT for grocery items detection and are working towards improving our results for SIFT along with exploration on Faster R-CNN and YOLO.

Progress:

1. Collection of dataset images

We have created a dataset of individual shopping items as reference images & cart images with multiple items in different orientations to test our implemented algorithms. The reference images were taken based on varieties in sizes, shapes and features.

2. SIFT on train dataset

Our code uses SIFT algorithm and vl_sift API to extract features and descriptors of images from every individual shopping items. We are also extracting bounding box, i.e, corner points of each image. The extracted features and corner points are then stored in a database.mat file which we are using later to compare with our cart items. An excel file has been added listing each item and its corresponding price.

3. Detection of one image using SIFT

The test set contained one item in cart image. We extracted features of the cart item using SIFT. The extracted features & descriptors were then compared against already extracted feature & descriptor datasets of reference images using vl_ubcmatch.

RANSAC was used to extract relevant features and then homography was applied to the bounding box of templates with most matches after RANSAC. Matching feature coordinates in query image were retrieved and if the query image feature was within the transformed bounding box of the reference image, we selected that image as a detected image with referenced one.

4. Detection of multiple images in cart using SIFT

The test set contained multiple items in cart image. In this case, we were extracting features of the cart image using SIFT. The extracted features of the cart image items were then compared against extracted feature datasets of reference images using vl_ubcmatch. Multiple iterations of RANSAC algorithm was run with a random subset of feature matches to determine a homography. The homography with most inlier matches were scored and chosen. We applied the homography to the bounding box of templates with most matches after RANSAC. Matching feature coordinate in query image were retrieved and if the query image feature was within the transformed bounding box, we detected that image and created a binary mask for the image in the cart. We repeated the same process to detect each item image from cart items until we covered all the items of the cart.

5. Detection of overlapped images in the cart (up to 10-30 percent)

Following through the approach of detection of multiple non-overlapped images and extracting images with most feature & descriptor matches, we were able to detect overlapped images with 70-75 percent accuracy.

6. No Detection of missing images in the database

The test set contained cart images with an item missing from the database. Our algorithm didn’t detect such images filtering out based on the minimum required number of features & descriptors match of reference images which weren’t fulfilled.

Results:

1. Dataset images

Collection of dataset items used in results

2. Single item

Detection of single cart item

3. Multiple items

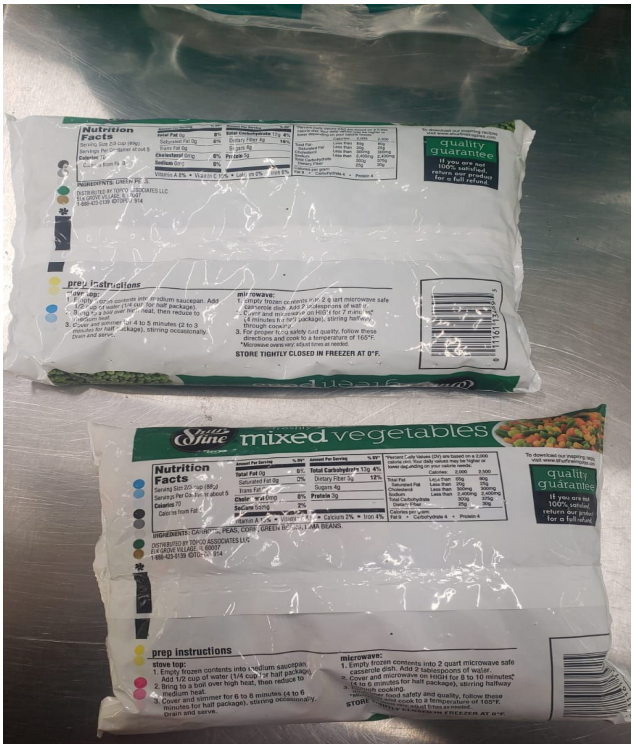
Detection of multiple items in the cart

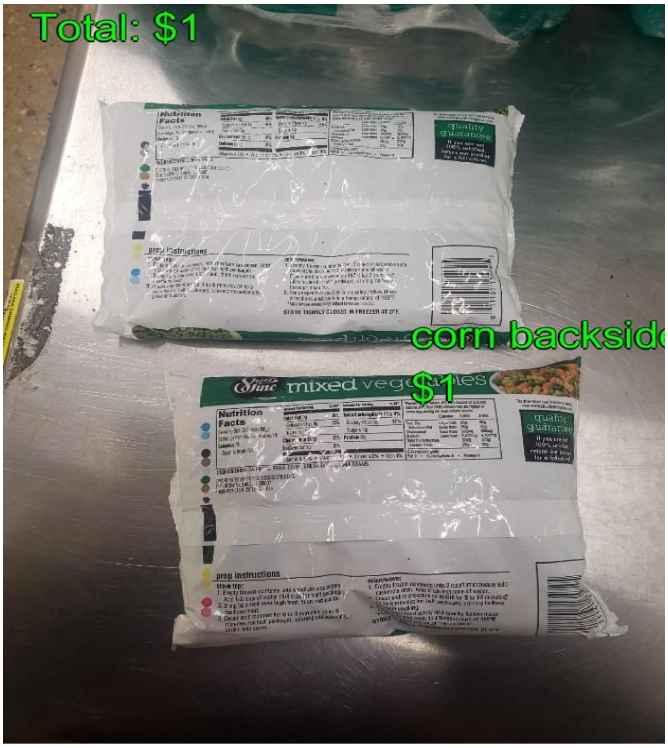
4. Cart images with some overlap

Images getting detected even with some overlap

5. Items with insignificant features mis-detected/not detected

The packet of mixed vegetables (lower packet) is detected as corn and the corn packet (upper packet) is not getting detected due to lack of significant features.





6. Cart items with front and back sides

Although few items in the cart are placed with their backside facing up (which usually contains fewer features), items are still getting detected as they still have sufficient features to be compared against dataset items.

7. Cart images with false positive and false negative

False positive(incorrect detection): sausage

False negative(no detection): shoe-polish

8. Cart images with missing training data of an image

Sharpie is not detected as it is not present in the database.

9. Cart images with unwanted image detection

Noise is sometimes detected as cart items (like sausages in the image below) which led us to include a pre-processing step to crop out the unwanted part of the cart image like cart posters in this case.

Challenges:

1. Automatic computation of RANSAC threshold

Some corresponding points computed using automatic interest point detectors may not concur with the actual homography. These points are called as the outliers. In order to address the outlier problem, we need to specify a RANSAC threshold for each image specifying the acceptable alignment error in pixels which incurs additional overhead. In order to automate image detection, we need to generalize RANSAC threshold value or calculate it based on the given image.

2. The requirement of a pre-processing step

Sometimes, running our algorithm on cart images detected false positives (detection of items not present in the cart). RANSAC threshold values seemed to be affecting the issue. On altering that value we ended up with results containing few false negatives (cart items present in the cart not getting detected). But the major problem was detected to be coming from noisy images which contain advertisement poster on the side of the cart. This led us to the requirement of clean images which we accomplished with the preprocessing step of cropping out cart posters.

3. The scope of performance improvement

The algorithm doesn't have a 100% accuracy. It often misses detecting a few items that are present in the cart and often detects items that are not part of the cart image. These

issues were occurring in cases where image items didn't have many distinguishable features that could be extracted for a good match. Even with the items containing images of both back and front sides in the database, this problem persists as some sides of an image doesn't provide sufficient features for detection. Moreover, some images have similar sides thus causing an issue of a mismatch.

4. Trade-Off between Runtime and Accuracy

Extraction of features using vl_sift gives us more than 3000 features. Comparison of such large number features in vl_ubcmatch takes more than 4-5 minutes. In order to avoid that we are using edge and peak thresholds as 7 and 10 respectively which captures less than 3000 features per images, thus significantly reducing the computation time of vl_ubcmatch. This often results in a compromise in accuracy but gives reduced computation time. This leads to a tradeoff between runtime and accuracy.

Changes to proposal in light of current progress:

The aim of our project has been changed from implementing SIFT algorithm as a stand-alone project to exploring machine learning algorithms for image detection with two new approaches - YOLO and Faster R-CNN.

Future Work:

1. Stress testing on implemented SIFT algorithm

Although SIFT is illumination-invariant, our stress testing will involve images of cart items with extreme lighting conditions,i.e., cart images with high and low illumination.

2. Improvement of SIFT accuracy

Current gained accuracy is 70-75 percent and our future work aims to improve the accuracy to 80-90 percent.

3. Comparison of accuracy among the three algorithms (SIFT, YOLO, Faster R-CNN)

Our aim is to extend our algorithm of SIFT implementation and include YOLO & Faster R-CNN as well. Subsequently, compare the results obtained by the 3 algorithms with respect to accuracy, run-time complexity and robustness. We shall also compare the feasibility of adding new items in the dataset and the time taken by the algorithms to train on the same.

References:

1. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi.”You Only Look Once: Unified, Real-Time Object Detection”(2016)
2. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”(2016)
3. Digman, Michael and Crawford L. Elder. “Mobile Banknote Recognition and Conversion.” (2013).
4. Lowe, D.: “Distinctive image features from scale-invariant keypoints, cascade filtering approach”. IJCV 60, 91–110 (2004)
5. http://en.wikipedia.org/wiki/Scale-invariant_feature_transform
6. Lowe, D.G.” Object recognition from local scale-invariant features. In International Conference on Computer Vision”, Corfu, Greece, pp. 1150–1157.(1999)