Shopping Cart Item Tracker

Dian Huang
dhuang05@stanford.edu

Motivation: Customers may find it difficult to verify the items in the shopping cart. Tracking items in shopping cart involves with action recognition and feature matching. In this work, I will design an action recognition method without training samples and implement SIFT feature matching. Finally, these algorithms would be implemented as an Android App.

Algorithm Overview

1. Identify key Frame
2. Subtract Image
3. SIFT Matching
4. Action recognition

Identify key frame: background

Image Subtraction & SIFT Matching

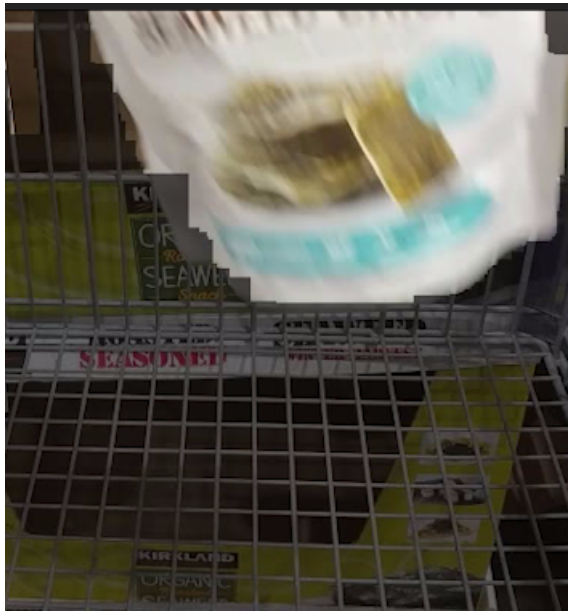
identify if item is being taken out

Key Frame Detection

1. Resize to 800x400
2. Apply mixture of Gaussian background subtraction
3. Identify key frame based on foreground pixel count

Background:

Static, no foreground Pixels



Entering

Foreground:

Static, no foreground Pixels

Leaving

Key Frame Detection

- Minimum frame: Background
- Threshold frame: Entering
- Local minimum frame: Item drops

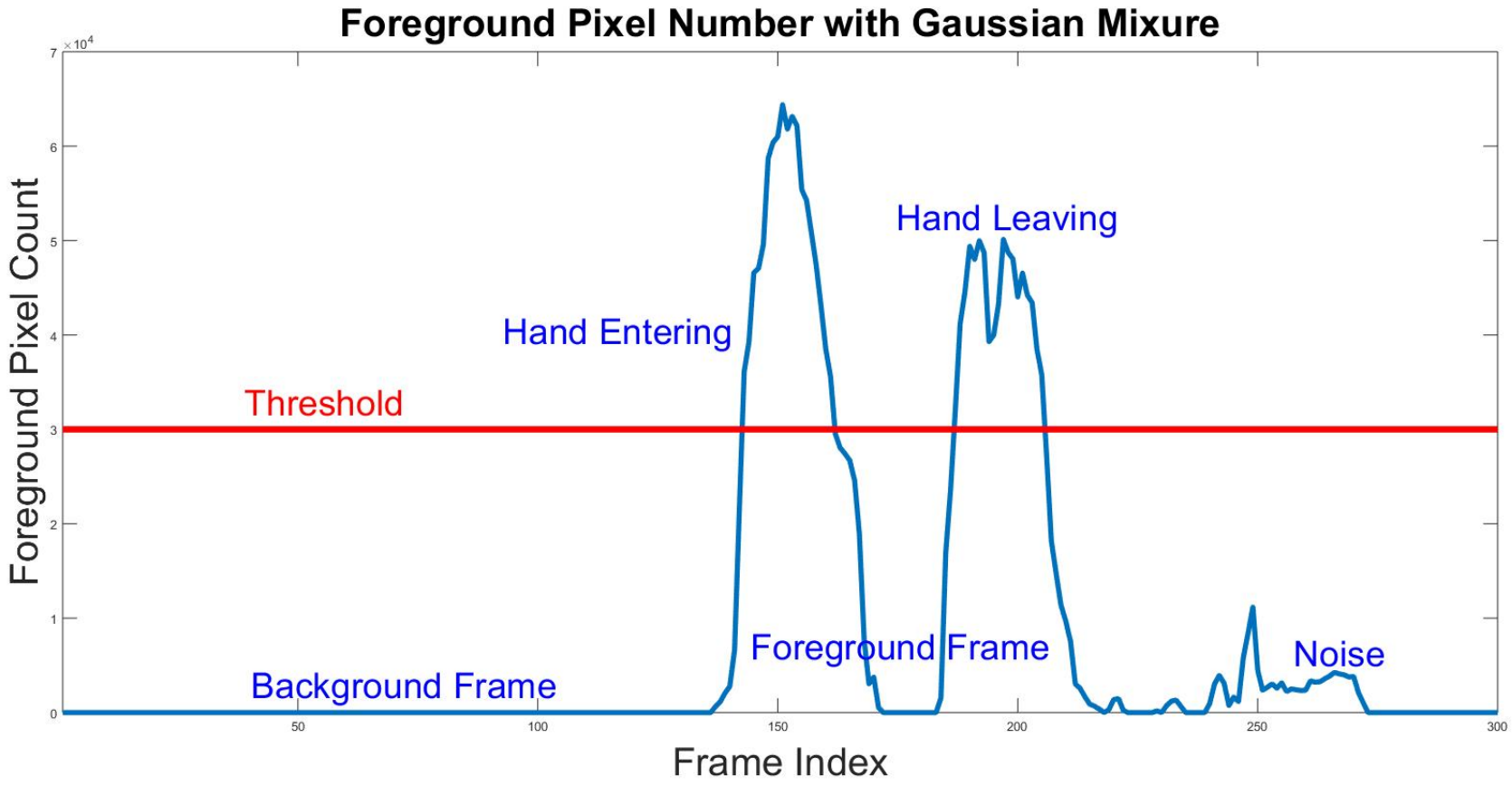


Image Subtraction: Issue

Issue: Difficult to choose threshold for various conditions: illumination, item , etc.

Background: no shadow

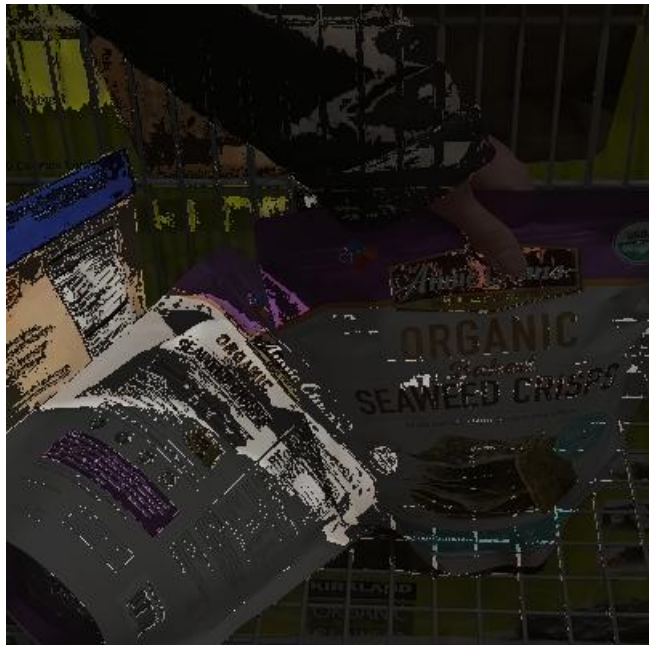
Foreground has shadow

Item not fully shown yet, but shadow has been revealed.

Image Subtraction: Shadow Detection

Convert RGB to HSV

- $|H_{fg} - H_{bg}| \leq \tau_H$
- $S_{fg} - S_{bg} \leq \tau_s$
- $\beta_1 \leq \left(\frac{V_{fg}}{V_{bg}}\right) \leq \beta_2$



Detect Shadow with Chromaticity

Without Shadow Removal

With Shadow Removal

- Remove shadow partially
- Still need to find threshold for various condition

Image Subtraction: Double Threshold

- Divide the image into top and bottom and process them independently
- High threshold: Select only the largest region
- Low threshold: Select only the region that has some part of the high threshold region as subset

High threshold:
Missing key features

Low threshold:
has unwanted region

Keep only the region that has part of the high threshold region as subset

SIFT Feature Matching

- Perform SIFT matching against 50 different items image

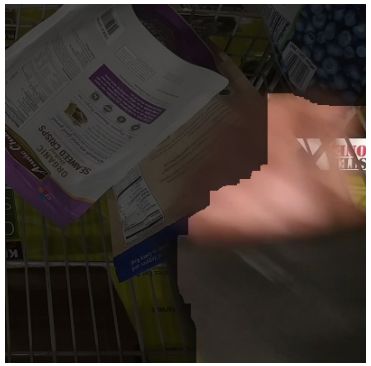
Action Recognition with LDA

- Need to figure out if user is placing in or taking out an item
- Classify action with LDA: work if the clothes and the item have different color
- Map entering foreground pixels to LDA eigenvector

- **Color data for clothes**
- Ignore the middle part
- **Color data for Item**

LDA Eigenvector

Typical Entering Scene



Taking out

Putting in

Demo: Setup

- Phone placed on top of the cart
- Only half of the cart region is being captured

Background Subtraction

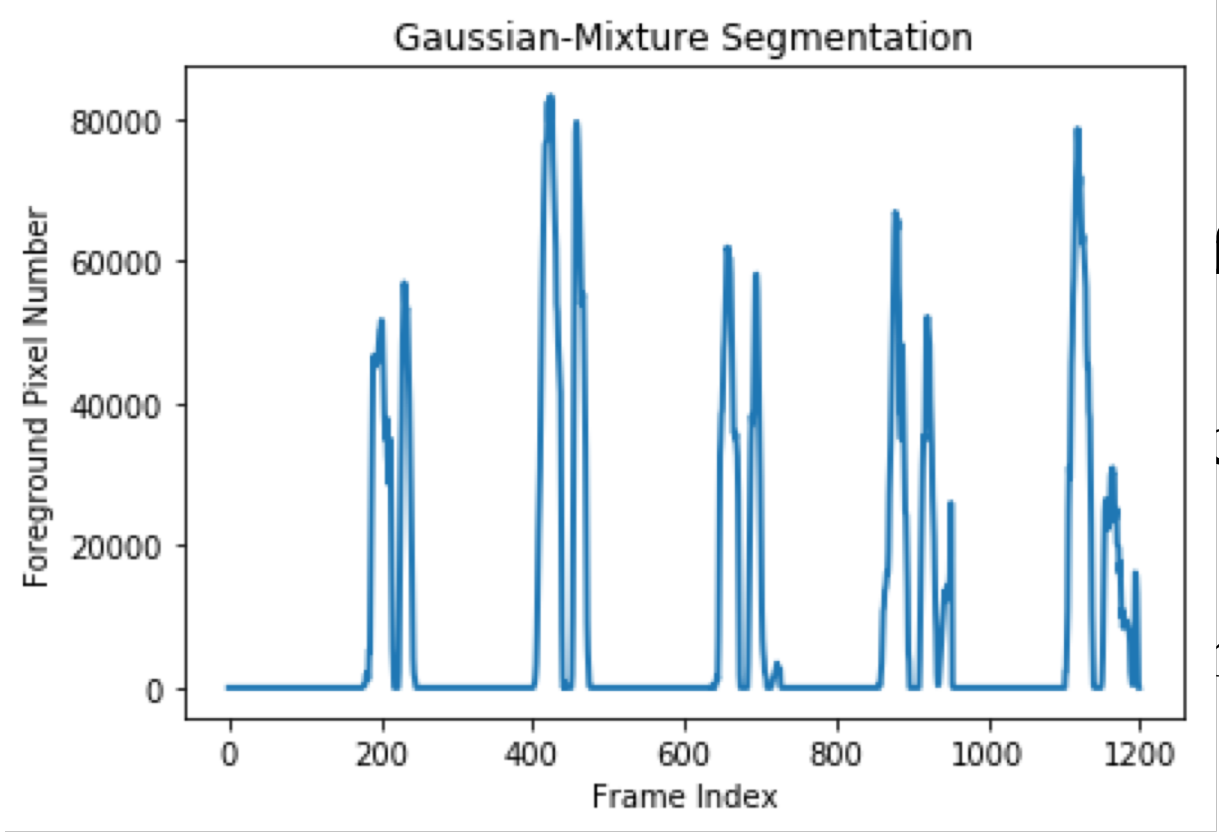
Mixure of Gaussian: Find Local minimum

Basic Algorithm:

1. Identify local minimum in number of foreground pixels
2. Subtract this frame with previous background frame
3. SIFT matches it to the correct item
4. Classify placing in/taking out an item with LDA

Image Subtraction

Goal: Count the number of foreground pixels



f foreground

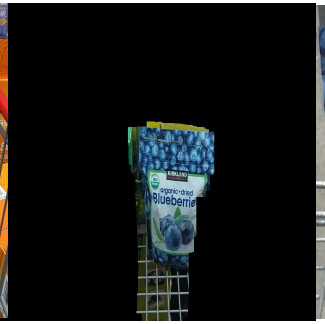
ckground frame

n with LDA

Dataset and Initial Results

- 1 photo for each item (50 items)
  - Camera on top of shopping cart
  - Hold item for about half second

- Image subtraction: Detect temporary stable background and subtract them. Issue: Could not identify taking out almond or putting in blueberry (could not identify action)



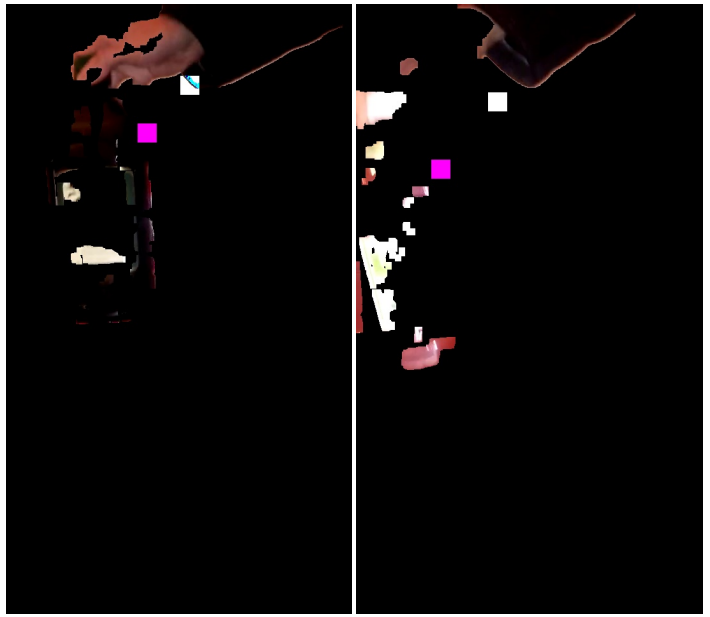
Original

Take out almond or put in blueberry?

Image Difference for SIFT

Sample image taken for each item

- Action recognition with no training data
  - Use PCA with Kalman filter to track arm and item location and predict current action



Pink square: item location

White square: arm location