# **Item Removal Detection for Retail Environments by using Neural Networks**

|  |  |
| --- | --- |
| Lingjie Kong  Stanford University  Department of Mechanical Engineering  ljkong@stanford.edu | Xingchen Fan  Stanford University  Department of Mechanical Engineering  xcfan@stanford.edu |

**Abstract**

*The ABSTRACT is to be in fully-justified italicized text, at the top of the left-hand column, below the author and affiliation information. Use the word “Abstract” as the title, in 12-point Times, boldface type, centered relative to the column, initially capitalized. The abstract is to be in 10-point, single-spaced type. The abstract may be up to 3 inches (7.62 cm) long. Leave two blank lines after the Abstract, then begin the main text.*

# Introduction

Neural network models have been successfully applied to recognize human actions from images and videos. This paper explores how deep neural networks with computer vision can be used for action recognition in a very specific setting, namely item removal detection in retail environments. The most related technology in the market today is Amazon Go, where computer vision is combined with weight measurements from scales embedded in shelves to detect item removal in grocery stores. Our approach differs from Amazon Go such that we only use visual information to determine item removal and addition.

The input to the deep neural network model is video of people interacting with items on shelves in front of a vending machine. The camera is mounted at the top of the machine and triggered to record video only when the door is open.

We will then use several different deep neural network architectures to classify whether items have been removed or added to the shelf within the time frame of the video. Since our own dataset of videos is small, we will use pretrained models available online and implement transfer learning to avoid overfitting. Different approaches will be investigated and compared in terms of classification accuracy as well as computational efficiency. We start with single-frame image-based approaches and gradually increase the complexity of our models. We will gradually transition from image-based approach to video-based approaches as we incorporate more temporal information into our models.

# Related work

# Methods

Single-frame, early-fusion, late-fusion, C3D, dense trajectories, optical flow, two-stream, LRCN

# Dataset and features

Despite the fact that the pretrained models have been trained on some popular video datasets like UCF-101 and HMDB-51, we will retrain the models on our own dataset, which currently consists of 14 manually recorded and labeled videos but hopefully will expand in the future. Snapshots of our video data are shown in Figure ??.

|  |  |
| --- | --- |
|  |  |
|  |  |

The raw videos are recorded at 24 frames per second with 960×540 pixels. The videos will be downsampled spatially to give a size of no more than 256×256 pixels per frame. For each video, how the person interacts with items on the shelves could vary a lot. This could change the lengths of videos significantly. In order to simplify our task, all videos in training and validation sets will be split into several snippets that consist of only one direction of hand motion: either moving towards or away from the shelf. When the hand is moving closer to the shelf, we will classify the action as either one item added or nothing added; when the hand is moving away, we will classify it as either one item removed or nothing removed. During test or real-world implementation, we assume that some hand motion detection algorithm will split the video automatically before the classification by our neural network. Details of hand detection and more variations on addition and removal are beyond the scope of this paper and will be interesting to study in the future. The sampled video snippets of interest will have 20 to 40 frames. Depending on the method, we will either use all frames or further extract a subset of frames for classification.

Furthermore, we will explore whether certain handcrafted features or data augmentation procedures could improve classification. Ideally, convolutional layers will learn to extract features from frames at various scales automatically throughout training. However, some preprocessing of data could speed up training and reduce the complexity of the network. Common techniques to implement include principal component analysis (PCA) and histogram of oriented gradients (HOG).

# Experiments/Results/Discussion

# Conclusion

# References