# **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**

*Inspired by the recent success of Recurrent Neural Networks (RNN) and Convolutional Neural Network (CNN) in classification, localization, and segmentation, we propose a neural network application in item removal detection for retail environments. Different Amazon Go which relies on both sensor fusion, and deep learning algorithms, we focus on only use only deep to enable customers to explore and shop more efficiently. Unlike traditional image classification, the input into our network is a steam of video while the output is the prediction of class of removed items as well as number of removed items. We will implement popular video classification algorithm, compare the performance, and explore method to enhance the prediction result. Specifically, C3D, two stream convolutional networks, long term recurrent convolutional network will be implemented. Eventually, we will explore the pros and cons of each networks and try to find the best network to solve this item removal detection problem.*

# 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

So far, Convolutional Neural Networks (CNNs) [1] have outperform most other algorithms in understanding image contents and shows the state-of-the-art performance in image classification, localization, segmentation, detection, and so on [2] [3] [4] [5] [6] [7]. Under these condition, CNNs is power enough to extract image feature [8].

However, currently there is not a single video classification benchmark that perform at the same level of current image data set. First, compare to images, videos are significantly harder to annotate. It takes large amount of time to collect large enough of data to feed CNNs. Second, videos have more dense information compared to image and it is much hard to classify. Last but not least, each frame of the video has temporal and spatial relationship.

Currently, there are three major approach to solve video classification problems: C3D, Long-term Recurrent Convolutional Networks, two-steam Convolutional Networks.

C3D stands for deep 3-dimentionsal convolutional networks (3D ConvNet) [9]. Different from image CNNs which apples a 2D convolutional networks, C3D simply stack each frame of video together into 3D space and apply a 3D convolutional filter in all intermediate hidden layers and some fully connected layers at the end.

Two-steam convolutional networks incorporate spatial and temporal networks [10]. Unlike C3D which feed stacked images which is extracted from video, Two-steam convolution networks combine both spatial stream convent from single frame and temporal steam convent from multi-frame optical flow. The result from both CNNs will be concatenated together and feed into fully connected layer eventually. Another two steams approach is also developed for large-scale video classification by using a context steam that learns features on low-resolution frames and a high-resolution fovea steam that only operates on the middle portion of the frames [11]

Inspired by RNNs which are widely used in natural language processing (NLP) [12], Long-term recurrent convolutional networks are also developed for visual recognition and description [13]. Different from CNNs, RNNs use LSTM structure. At each time step, it feeds in both the hidden information from last time step as well as one frame from the video.

# Methods

We first use pre-trained model from tensorflow for video classification dataset.

We first implement the C3D because it is easy to implement by only adding a depth direction for 3D convolution. Convolutional layer dimension was finely tune to 3x3x3 which give the best performance. Stochastic gradient decent is use to only update the parameters for the last fully connected layer while keeping the hidden layer parameters fix. Random search is used for find the best regularization to prevent overfitting.

Second, we are planning to implement two-steam by using two-stream architecture for video recognition. Similarly, we will use pretrained model. We will form two steam convolutional networks for spatial steam from single frame and temporal steam from multi frame optical flow. Eventually, two frame results will be either concatenated together through fully connected layer and apply stochastic gradient decent the minimize the softmax loss function.

Third, we will implement Long-term recurrent convolutional networks. Each RNN layer will use LSTM structure to prevent gradient explosion as well as gradient vanishing. Similarly to CNNs case, hidden layer RNNs will from pre-trained model and we will fine turn the fully connected layer parameter from our training data set.

# 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

Before running C3D, two-steam, and long-term recurrent convolutional networks, we first implement a dummy model as the baseline for comparing the result.

In the data preprocessing, we sample 5 frames per video and down sample its dimension to . As a result, our input data have a dimension of while each dimension is represented by batch size, input depth, input height, input width, and input channels.

In total, we have 101 data sample, we split the data into 81 for training, 10 for validation, and 10 for testing.

For the first approach, we are inspired by the CNNs for image classification, we first convert three color channels into one gray scale channel and reshape the data to the dimension of . In other words, we convert the input depth to channels after grayscale the color images. The later step is same as the classical image classification problem. We coded a two layer neural networks. The first layer being a convolutional layer with a filter dimension of and a total number of 16 filters. The second layer is a fully connected layer with an input dimension of 13456 which is the output from the first convolutional layer and an output of 4 classes. After training the model by using stochastic gradient decent for 20 epochs, we get a final training accuracy of 0.988, validation accuracy of 0.5, and test accuracy of 0.6 which outperformance the random guess which is 0.25.

Second, we implement a C3D two layer neural networks model. The first layer is a 3D convolutional filter and the second layer is a fully connected layer. For the first layer, the input for C3D model is , while each dimension is represented by batch size, input depth, input height, input width, and input channels. The filter for C3D is , while each dimension is represented by filter depth, filter height, filter width, input channels, and output channels. We pass the result from the first layer into a fully connected layer with a dimension of 7688 and an output dimension of 4. We also add a L2 regularization with a regularization weight of 0.05 to prevent overfit. Eventually, we get a training accuracy of 1, validation accuracy of 0.4 and test accuracy of 0.5. This also outperform compared random guess.

As for the next step, first, we will collect more data because we have limited amount of data 101 in total. Second, we will implement deeper C3D networks to get more accuracy result. Meanwhile, we will fine tune the hyperparameter as well as regularization to get best result. Third, we will implement two steam model. Forth, we will implement RNN model using LSTM. Eventually, we will compare all result to see which model will give the best performance.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[2] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst. 25*, pp. 1–9, 2012.

[3] C. Farabet, C. Couprie, L. Najman, and Y. Lecun, “Learning hierarchical features for scene labeling,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, 2013.

[4] D. Ciresan, A. Giusti, L. Gambardella, and J. Schmidhuber, “Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images,” *Nips*, pp. 1–9, 2012.

[5] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks,” *arXiv Prepr. arXiv*, p. 1312.6229, 2013.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[7] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: An astounding baseline for recognition,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 512–519.

[8] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks arXiv:1311.2901v3 [cs.CV] 28 Nov 2013,” *Comput. Vision–ECCV 2014*, vol. 8689, pp. 818–833, 2014.

[9] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2016, vol. 11–18–Dece, pp. 4489–4497.

[10] K. Simonyan and A. Zisserman, “Two-Stream Convolutional Networks for Action Recognition in Videos,” *arXiv Prepr. arXiv1406.2199*, pp. 1–11, 2014.

[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. F. Li, “Large-scale video classification with convolutional neural networks,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[12] A. Karpathy and F. F. Li, “Deep visual-semantic alignments for generating image descriptions,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07–12–June, pp. 3128–3137.

[13] J. Donahue *et al.*, “Long-term recurrent convolutional networks for visual recognition and description,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07–12–June, pp. 2625–2634.