Activity Recognition on Kinect-3D Videos using Transfer Learning

Deep Learning Final Project Report

Jianhang Chen
School of Electrical and Computer Engineering, Purdue University
West Lafayette, IN
chen2670@purdue.edu

***Abstract*—This project is to develope an algorithm to recognize daily actions on 3D Kinect videos. The final result is generated by a combined CNN and LSTM network. After decided hyperparameters like sequence length and max frames by experiment first, we trained the network on UCF101 2D video dataset and fixed the CNN part and retrained LSTM on 3D dataset.**

***Keywords*—video classification; 3d kinect video; LSTM; transfer learning**

I. INTRODUCTION

In this project, we develop an algorithm to analyze videos to detect daily activities such as falling, grasping, running, sitting, etc. At first, we directly trained CNN+LSTM model on a 3D video dataset named TST FALL DETECTION DATASET V2 [1], which is shown in Figure 1. Unfortunately, the result was unsatisfactory to classify daily activities. The reason of failing might be the limited number of videos (approx. 88x3 videos) for training. Finally, we pretrained an CNN+LSTM network on UCF101 2D dataset [2], shown in Figure 2, and transferred it to 3D dataset. The validation accuracy of 3D videos is 52% and the test Accuracy of 3D videos is 46%.

The following part consists of 4 sections: section II is a brief introduction to other work. In section III, we introduce three sessions in our final work in detail. Section IV is the results of our experiment and section V is discussion and conclusion.

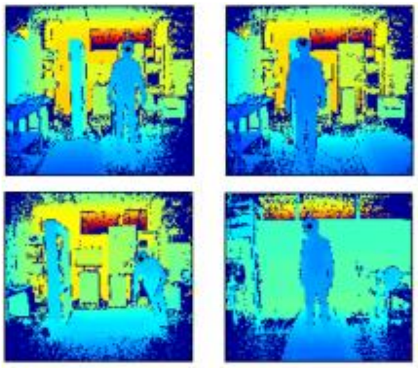

Figure 1 TST FALL DETECTION DATASET V2

Figure 2 UCF101 activity dataset

II. OTHER WORK

There are several methods to classify videos. Andrej Karpathy, et al., [3] extended the connectivity of a CNN in time domain to train the network to understand the activities in videos. It has successfully classified video of all kinds of outdoor sports. Figure 3 shows the Multiresolution CNN architecture developed by Andrej Karpathy for video classification. But intuitively, it only uses the current image to identify activities, and doesn't explain how to build a CNN based model that spans a few or more images.

Joe Yue-Hei Ng, et al., proposed a method explicitly models the video as an ordered sequence of frames [4]. The method employs a recurrent neural network that uses Long Short-Term Memory (LSTM) cells which are connected to the output of the underlying CNN. It shows that the use of LSTM's RNN (84.6%) may outperform the pure CNN model (72.6%). Figure 4 is the overview of CNN+LSTM approach by Joe Yue-Hei Ng, et al. In CNN, a large amount of information cannot be extracted if only from a single image that forms a video without considering the time sequence. Therefore, the similar method is chosen for this project except the optical flow information and feature pooling part. Our work is an extension of this method to depth images of 3D videos from Kinect.

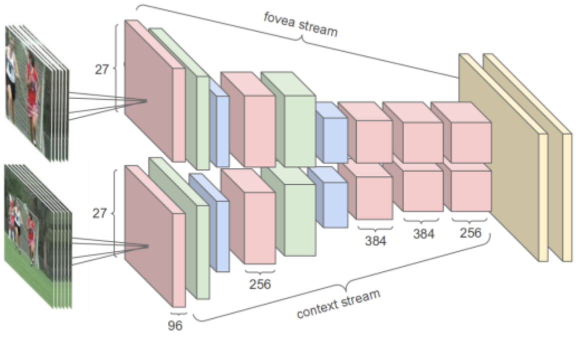

Figure 3 Overview of the pure CNN approach

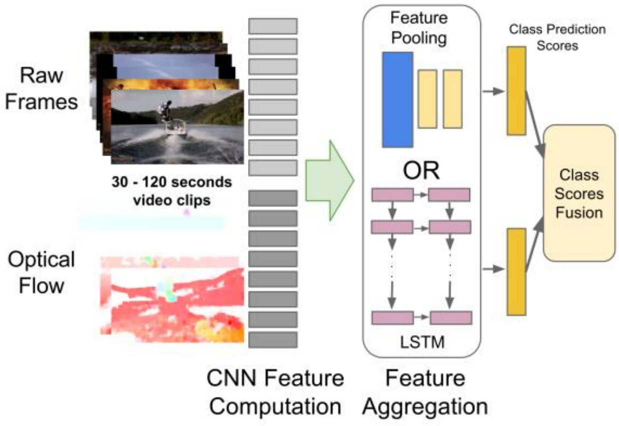

Figure 4 Overview of the CNN+LSTM approach

III. Our contribution

As mentioned above, we first directly trained our custom CNN feature extraction + LSTM model on a 3D video dataset named TST FALL DETECTION DATASET V2.

Since we did not get satisfactory result, we implemented 3 sessions to achieve our final goal for 3D video classification which is illustrated in Figure 5. First we selected hyper parameters including Sequence Length, Max Frames, Image Dimension and Epochs for training. Sequence Length is the number of frames to represent the video. Max Frames is the max number of frames of a qualified training video. We selected hyper parameters using pre-trained model of Inception v3[5] on UCF101 2D dataset. Then a small CNN network is trained to extract features along with a simple LSTM to train the sequence of features extracted representing the video. Finally, the trained CNN model is frozen (the weights are retained) and extracted features of the 3D videos from the CNN is fed into a simple LSTM for training.

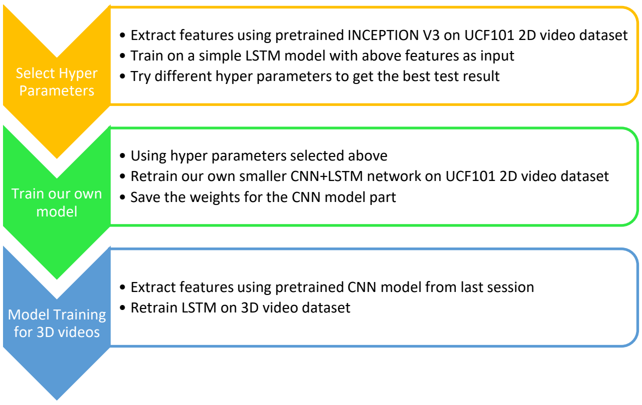
Select Hyper Parameters
• Extract features using pretrained INCEPTION V3 on UCF101 2D video dataset
• Train on a simple LSTM model with above features as input
• Try different hyper parameters to get the best test result
Train our own model
• Using hyper parameters selected above
• Retrain our own smaller CNN+LSTM network on UCF101 2D video dataset
• Save the weights for the CNN model part
Model Training for 3D videos
• Extract features using pretrained CNN model from last session
• Retrain LSTM on 3D video dataset

Figure 5 Overview of the 3 sessions of our work

A. *Hyper Parameter Selection*

1. Load the 2D video UCF dataset and partition it into images and save them.

2. Split the extracted video dataset into Training and Testing according to the split version files provided with the dataset

3. Select a sequence length that will represent a single video as a collection sequenced images

4. Clean the dataset that needs to be loaded (i.e). If the no. of images for a video is less than sequence length, drop it.

5. Load the images of a video in order but not exceed the sequence length (skip intermediate images)

6. Extract features for each image in the sequence from the penultimate layer of pre-trained Inception v3 model and save them.

7. Load the 2048 feature map of training dataset and validation dataset (a split from test dataset)

8. Fit a simple LSTM model with no. of input as 2048 and output nodes as no. of classes

9. Repeat the above process for various sequence length and max frames to get an optimized parameter with highest accuracy on test dataset.

B. *Model Selection*

1. Now, load the 2D video dataset using optimum hyper parameter found in previous step.

2. Generator based data loading is used (since loading all the images causes memory overflow). Images are resized using Cubic Interpolation.

3. Use a simple CNN model shown in Figure 1 to extract the features (Input is the normalized gray scale version of the RGB data).

4. Feed the Output of CNN to a simple LSTM with the number of inputs as 512 and the number of output nodes as the number of classes

5. Train the combined model of CNN and LSTM together for the 2D video dataset

6. Save the weights for the CNN model only.

7. Test this model on the Test dataset

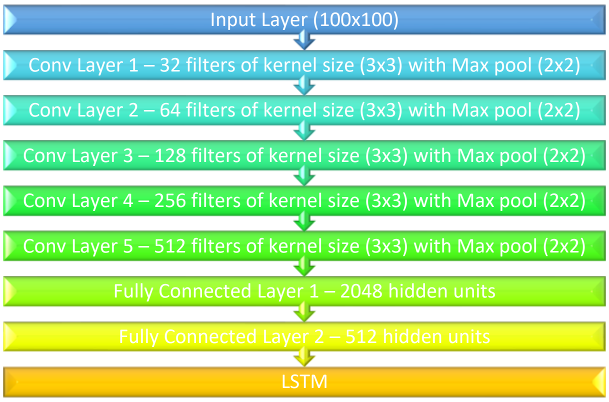
Input Layer (100x100)
Conv Layer 1 – 32 filters of kernel size (3x3) with Max pool (2x2)
Conv Layer 2 – 64 filters of kernel size (3x3) with Max pool (2x2)
Conv Layer 3 – 128 filters of kernel size (3x3) with Max pool (2x2)
Conv Layer 4 – 256 filters of kernel size (3x3) with Max pool (2x2)
Conv Layer 5 – 512 filters of kernel size (3x3) with Max pool (2x2)
Fully Connected Layer 1 – 2048 hidden units
Fully Connected Layer 2 – 512 hidden units
LSTM

Figure 6 Overview of the CNN architecture

C. Model Training for 3D videos

1. Load the 3D video files which are a sequence of binary files.

2. Normalize the depth files to fit the previous data input.

3. Extract features using the CNN model finalized in the previous method.

4. Train the LSTM to fit the 3D videos on Training and Validation split

5. Test its performance on the Test Split of the 3D dataset

IV. RESULTS

By our custom CNN+LSTM model, the validation accuracy of 3D videos is 52% and the test Accuracy of 3D videos is 46%.

Figure 9 shows some of the best hyper parameters selected in experiment. We also tried sequence length like 1 or 100, but the result is not so good.

Figure 8 and 9 shows sample results of 2D and 3D dataset.

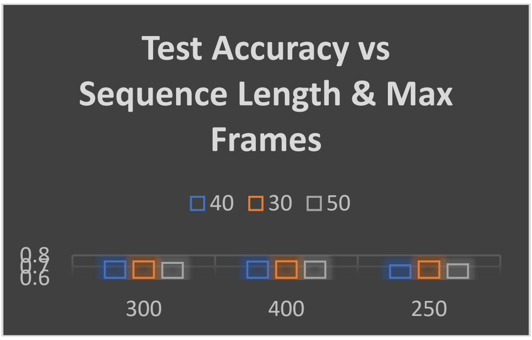

Figure 9 Hyper Parameters Selected

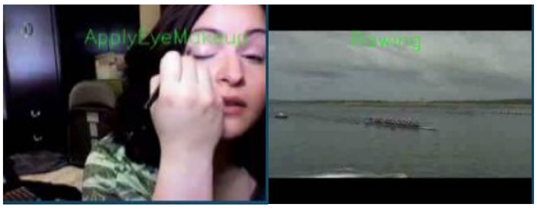

Figure 8 Sample results of 2D dataset

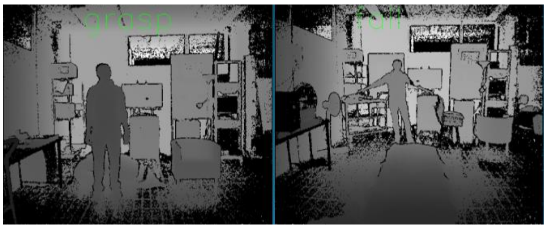

Figure 9 Sample results of 3D dataset

V. DISCUSSION

Concepts of Transfer Learning and RNN were understood through this project. We failed at first directly trained on 3D dataset. We also failed several times because we chose other hyper parameters. Through the experiment, we learned the significance of hyper parameters such as Sequence length, no. of Epochs and Max frames. Also, the Kinect could generate joint information of human body beside 3D depth data. The inclusion of joint data for training LSTM could improve the accuracy in classifying 3D - videos to predict actions in daily activity.

We use CNN+LSTM method which explicitly models the video as an ordered sequence of frames. This mothed could integrate information over time and have a better performance compared to pure CNN method.

REFERENCES

[1] S. Gasparrini, E. Cippitelli, E. Gambi, S. Spinsante, J. Wahslen, I. Orhan and T. Lindh, *Proposal and Experimental Evaluation of Fall Detection Solution Based on Wearable and Depth Data Fusion*, ICT Innovations 2015, Springer International Publishing, 2016

[2] Khurram Soomro, Amir Roshan Zamir and Mubarak Shah, *UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild.*, CRCV-TR-12-01, November, 2012.

[3] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L., *Large-scale Video Classification with Convolutional Neural Networks*, 2014

[4] Yue-Hei Ng, Joe, et al., *Beyond short snippets: Deep networks for video classification.*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015

[5] Szegedy, Christian, et al., *Rethinking the inception architecture for computer vision.*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.