Adapting and Improving Activity Recognition for Kuri
Bang Luu and Ifrah Idrees

1. Abstract:

Robots have the potential to improve health outcomes for the elderly by providing doctors and caregivers with information about the person's behavior, daily activities, and their surrounding environment. We are trying to detect events that have happened in a video captured by robots by looking at semantic information over multiple frames of images. Challenges in performing activity recognition on videos captured by mobile robots include:

- Moving camera base
- The viewpoint of the camera not being fixed since mobile robots can now rotate their head by 360 degree.

According to the research that we have done on the existing models, less focus has been given to the aforementioned challenges and hence less accuracy has been achieved in this context. We believe that improving human activity recognition for assistive robots will further enhance its capabilities to cater for the needs of elderly people living alone or people with disability. We configure Temporal Network Segment [2] model to perform activity recognition on videos captured by Kuri. Dataset of videos for the activities of our interest was collected and TSN architecture was configured and trained on this dataset. Our trained model was able to recognise videos successfully with 97% accuracy with the usage of RGB modality. Whereas with flow the model performed, on average, up to 76.5% rate of succession. Our model performance on videos captured by Kuri dataset are much higher than the pre-trained model from HMDB or UCF101.

2. Introduction:

One potential use of robots in personal assistance of the elderly is to answer questions pertaining to their health or their surrounding environment. Getting these queries answered accurately and timely by robots involves the intersection of many disciplines - robotics, computer vision, databases, natural language processing, and human computer interaction. Mucchiani et al. [1] conducted a user study highlighting the top 14 activities important in daily lives of the elderly. We'll be using this list as a ground for constraining the set of activities that we will be focusing

on. The activities that we will be focusing on is conversation, fall on floor, drinking, eating, walking, sitting, sleeping, and picking object.

Activity recognition models have been developed in the past years using hand-crafted features in the early years and now use deep learning techniques. However, these techniques do not take into consideration scenarios where the camera base is moving or the scenarios in which the viewpoint of the camera is changing and is from a low camera angle. To address these challenges we propose to explore various activity recognition models that enable robots to recognize activities of our interest. In particular, we will be providing the social and interactive Kuri robot from Mayfield [3] with activity recognition capabilities. The dataset models doesn't account for the moving camera base and the low angle view point.

The contributions of the paper are:

1. Collection and annotation of Kuri videos for the activities of our interest e.g conversation, fall on floor, drinking, eating, walking, sitting, sleeping, and picking object.
2. Configuring and training the Temporal Segment Network architecture, which is primarily based on BN-Inception model
3. Collected a separated dataset of videos with low camera angle outside of the test split. This dataset is labelled as "low-camera angle dataset"
4. Testing the trained the model on the test split of video captured by Kuri and low-camera angle dataset.

Our trained model of Temporal Segment Network architecture produced more accurate results then the pre-trained models that we have found.

3. Related Work:

Deep Learning methods, which are now the dominant approach, are generally composed of a feature extractor network and a sequence generating recurrent neural network. Convolutional Neural Network models (CNNs) or Long short-term memory networks (LSTM) can be used to identify, learn, and localize objects in these frame rate and automatically describing the content of the images. LSTM can be trained in a way that pay specific attention to observations made in the input sequence (back propagation.) LSTM doesn't have a limitation of fixed window like CNNs does. Further limitation of LSTM include that it requires 4 linear layer (MLP layer) per cell to run at and for each sequence time-step. Linear layers require large amounts of memory bandwidth to be computed, in fact they cannot use many compute unit often because the system

has not enough memory bandwidth to feed the computational units. And it is easy to add more computational units, but hard to add more memory bandwidth.

Various activity recognition models exist and each have different approach to detect activities. Traditionally, events were detected while looking at static appearance of single frames while now multiple frames are taken into consideration for the optical flow that is pixel containing motion information [4]. Further, various deep learning techniques are now available [5],[6],[7],[8]. Further, a variety of 2D and 3D convNets are now available. [14], [18], [156], [160] are some of the options of the 3D convNet. While [19], [30], [153], [163], [164] are the 2D- convNets that use multiple streams. TSN is yet another 2D-convNet architecture that combines a sparse temporal sampling strategy and video-level supervision to enable efficient and effective learning using the whole action video. [2]

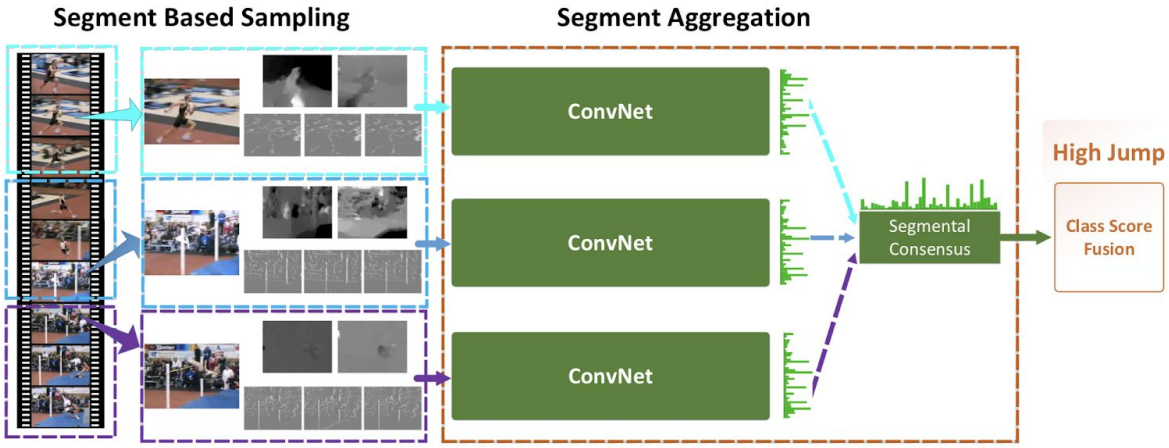
4. Technical Approach:

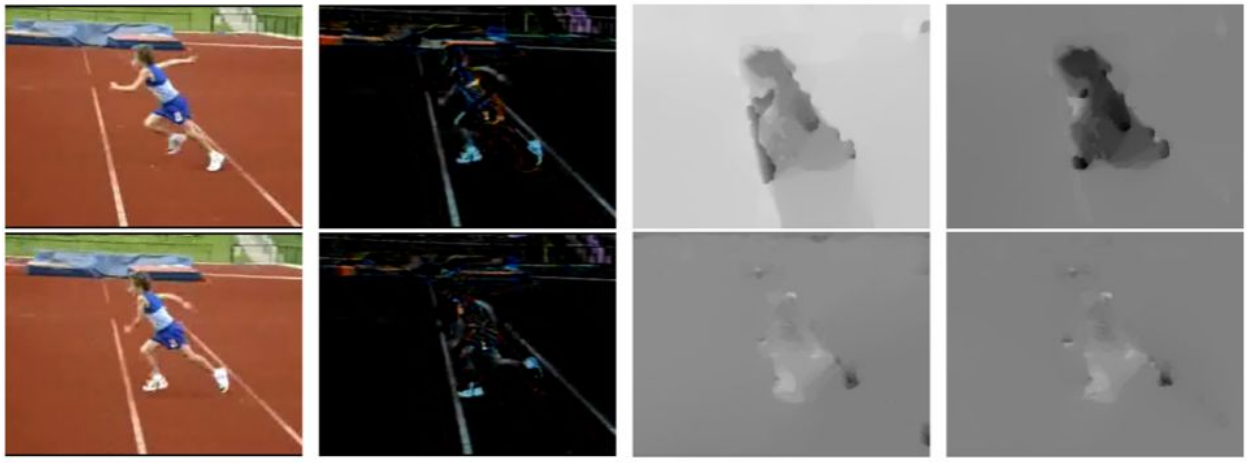
We started exploring various activity recognition models and the first decision that we had to make was whether to choose hand-crafted shallow techniques or to use a CNN model. We leaned towards a CNN model since it allows for local dependency and scale invariance. Our next line of decisions included whether to use a 2D or 3D convolution techniques and further we also had to take into consideration which models incorporated changing viewpoints. After exploring a range of models we chose Temporal Segment Networks(TSN) [2] .

4.1 TSN's architecture

The network used by Temporal Segments is Bare Network Inception Model. Their method sample clips sparsely across the video to better model long range temporal signal instead of the random sampling across entire video which is an effective solution aimed at long range temporal modeling. See Eq.1 of [2] for more details.

"Temporal segment network: One input video is divided into K segments (here we show the K = 3 case) and a short snippet is randomly selected from each segment. The snippets are represented by modalities such as RGB frames, optical flow (upper grayscale images), and RGB differences (lower grayscale images). The class scores of different snippets are fused by an the segmental consensus function to yield segmental consensus, which is a video-level prediction. Predictions from all modalities are fused to produce the final prediction. ConvNets on all snippets share parameters."[2]

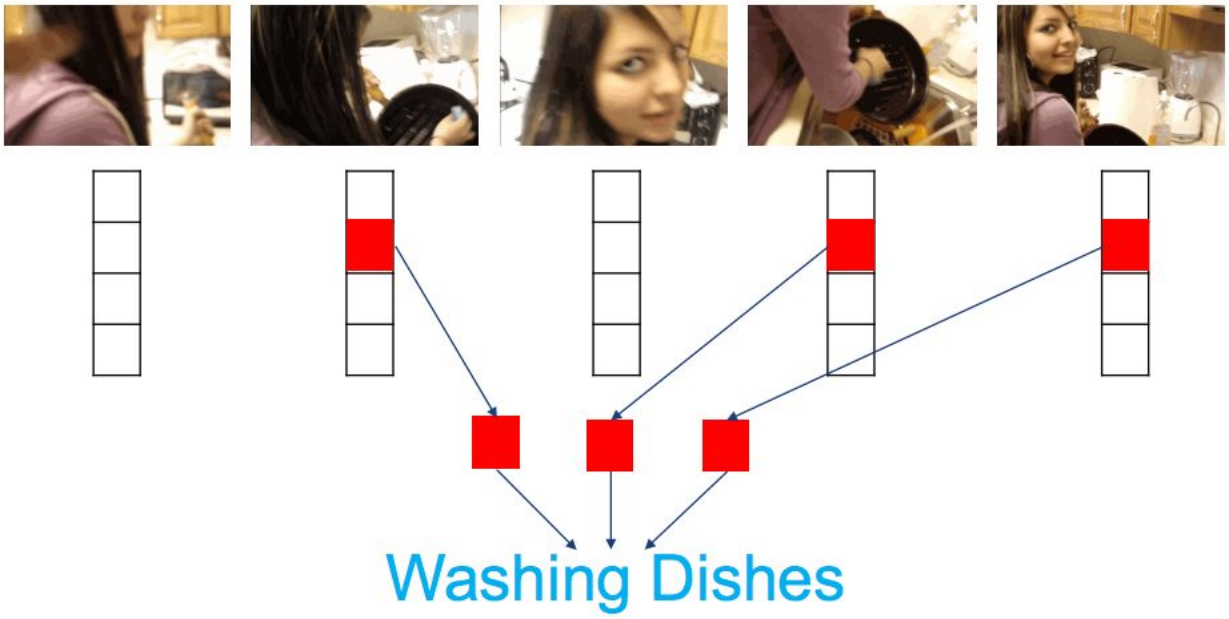




Examples of four types of input modality: RGB images, RGB difference, optical flow fields (x,y directions), and warped optical flow fields (x,y directions) [2]

4.2 Aggregation Function

The aggregation function we used is Top-K pooling that was in the TSN repository. It is able to determine a subset of discriminative snippets adaptively for different videos. It has the capacity of jointly modeling multiple relevant snippets while avoiding the influence of background snippets.



In the prediction stage for video, the strategy they used was to combined scores of temporal and spatial streams (and other streams if other input modalities are involved) separately by averaging across snippets. Another strategy they used was fusing the score of final spatial and temporal scores using weighted average and applying softmax over all classes. We find that these strategies could be optimal in analyzing snippets of video clips from Kuri.

4.3 Sliding Window

For varying duration of action clips we combined the topK pooling aggregation function with sliding window mechanism. “Sliding windows with different sizes are then applied on the frame scores. The maximum scores of the classes within a window are used to represent it. To alleviate the interference of background contents, windows with the same length are then aggregated with a top-K pooling scheme. The aggregation results from different window sizes then vote for the final prediction of the whole video.”[2]. For more formal working of this mechanism, please refer to Eq.13 of [2].

4.4 Evaluation of pre-trained models

We first test the pre-trained models on testing splits of UCF101 and HMDB51 dataset. The two mainstream action recognition dataset has about 152 action category and 20,320 video clips to use and test video on. The accuracy achieved on these testing splits is same as accuracy mentioned in their paper. Next, we test the pre-trained model on videos we capture from Kuri Robot[3]. The top activity result that we had found is mopping. Since the floor is detected in every frames, the models assumed that the activity that is taken place is mopping. This lead us to gather our own kuri dataset to

emphasizes on the activities with the low-angle camera. The quantitative results of can be seen in section 5.

4.5 Data Collection

For our video collection for initial testing, we allow the Kuri Robot [3] from Mayfield Robotics to make observations and collect video in an indoor environment setting. Kuri has 2 RGB-D cameras that capture video feed with frame rate of 6FPS. The default size of the images is 1920*1080 pixels but for speeding up the preprocessing we resize the images to 1067*600. We collect 5 seconds of video which is equivalent to 45 frames. The video that we collected has activities like sitting, talking, walking, smiling, turning and standing.

Additionally, we wanted a model that focus only on the activities that occurred with a static environment. The green background allows us augment the surroundings and lets the neural network focus on the activities itself.

4.6 Data Augmentation

Data augmentation allows us to generate more data and give our model a more diverse training sample. In the original videos, random cropping and horizontal flipping are employed to augment training samples. We exploit combinations of the four data augmentation techniques: corner cropping and scaling, flipping orientation, and shearing.

4.7 Model Training and Testing

We extract the rgb frames and optical flow images for each of the 121 video organised under the 8 categories. We then had to create the file list of videos for training and validation. The file lists had to be annotated in the same pattern as the authors of the temporal segment network were using. Every row in the file list contained a tuple of the path, number of frames and video ground truth class. We split our 121 videos into 85% train and 15% test set. We use the initial weights of the model as provided by the authors. The docker the correct cuda installations was able to detect the gpu in the system correctly. We train our network on a single GTX 1070 GPU for both rgb and flow modalities. The rgb model took 4 hours to train on average while the training of the flow model took 8 hours.

The two modalities extraction methods for our videos that we had used was: RGB and Flow. In Kuri videos, there are lots of camera motion and movement, to optimizes for this-- optical flow was used to help filtered out background movements. Videos that are low-quality are better detected by RGB.

The trained model was then tested on

1. Video from the test dataset(15% of 121 videos for each label)
2. Video outside of the testside. 8 videos were collected for each label. The videos were collected while keeping in mind that the camera angle of the videos corresponded to the height of kuri's camera angle.

5. Technical Challenges:

5.1 Pre-midterm Challenges

According to their paper TSN could be applied to detect videos in untrimmed videos which means that the each video in testing dataset can have more than one activity performed in it. We had to change the aggregation function of TSN to incorporate sliding windows for untrimmed videos which we will be a common case for videos we will capture from Kuri. However, this aggregation needs to be further tested.

Kuri was build to be autonomous and this makes it difficult for us to gather our own videos. We tried multiple methods for teleoperating Kuri that is through rviz and keyboard teleop. Both of these method was successful in controlling Kuri movement temporarily. We still needed to be able to control Kuri's camera angle and stop Kuri's autonomous movements. We later made a few changes to the Kuri application and were able to override Kuri's autonomy enabling the teleoperation mechanism. We had to invest quite some time in figuring out this teleoperation mechanism.

5.2 Post-Midterm Challenges:

After midterm we move onto collecting dataset. 23 participants volunteered for capturing videos of 8 activities: conversation, eating, drinking, sitting, walking, picking up object, falling and laying down. We collect 5 seconds of video which is equivalent to 45 frames per activity for each of the participant. Before collecting the dataset we had to make decisions as to what angle should Kuri be set to for capturing the videos and also in which area of the lab should these video be captured. We finalized to capturing the videos with a green background.

There were other technical challenges that we had to face while training our model. The authors had originally written scripts to train the model over 4 GTX Titan X GPUs so the scripts have to be changed to perform training on 1 GPU. Further, we had to create the deploy.prototxt files for our dataset catered to our output labels of length 8. However, we were able to resolve these issues and move forward with the training.

6. Evaluation:

6.1. Qualitative Results with pre-trained model:

We test TSN's model pertained on UCF101 on untrimmed video captured by Kuri. The untrimmed video of 5 minutes duration, that is 3130 frames. In this case, the scores of each frame are aggregated to give one label and the label we get for the Kuri's video is mopping the floor. To check the variance of our model we test this model multiple times on our collected video and every time we get the same label, showing low variance. We are able to justify the label "mopping floor" that we get for our video by looking at the training dataset for mopping the floor and those videos mostly contain the floor which is common for the videos captured with Kuri because of its default posture of looking down at floor. This was a general problem we knew we would be facing for Kuri because of its height and its default view of looking at the floor.

We also test TSN on video captured of Kuri with model pretrained on HMDB51 dataset and with a sliding window aggregation function returning multiple labels for one video. We are still making the sliding window technique work. The top five results we got for this approach were not that satisfactory and returned labels (in ascending order of probability): handstand, drawing sword, clap, smoke, turn. Turning (with the highest probability did happen in the video) but the other activities haven't occurred.

When we first tested the Kuri dataset with model pretrained on HMDB. We find that the model detects standing [64%], turning [87%], and running [78%] in walking videos. In eating videos, we find that chewing was in the top 5 activities that were listed. 92% of eating videos are labelled as clapping. In top 1, the results without sliding windows (default aggregation) are worse. The results of these observations illustrated that the camera angle influence the activities that were being selected. Additionally, other activity labels that are in relation to the activity itself is often selected within the top 5 activities.

6.2. Quantitative Results on (HMDB dataset) with pre-trained models:

Below are the result from testing the videos from HMDB51 dataset:

HMDB | flow_deploy | split 1 | SCORE_FILE_FLOW_1 = 62.156863%

HMDB | flow_deploy | split 2 | SCORE_FILE_FLOW_2 = 91.24% -> most accurate

HMDB | flow_deploy | split 3 | SCORE_FILE_FLOW_3 = 89.74%

Conclusion for flow deploy | split 2 is the most accurate in HMDB

HMDB | rgb_deploy | split 1 | SCORE_FILE = 54.575163%

HMDB | rgb_deploy | split 2 | SCORE_FILE2 = 88.43% -> most accurate

HMDB | rgb_deploy | split 3 | SCORE_FILE3 = 87.91%

Conclusion for rgb deploy | split 2 is the most accurate in HMDB

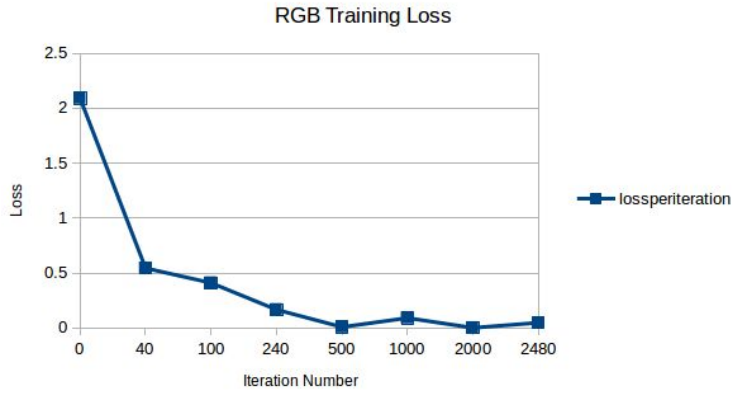
6.3. Quantitative Results on trimmed videos captured by Kuri with pre-trained models:

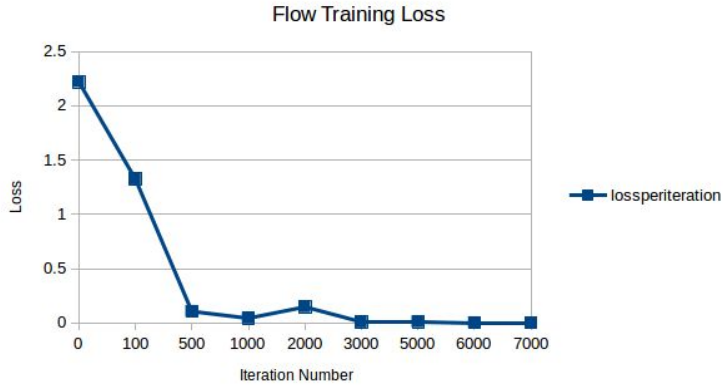
	Pre-trained model on HMDB dataset			
	Top 5%		Top 1%	
Modalities / Activities	FLOW	RGB	Flow	RGB
Sitting	3%	15%	~1%	~1%
Walking	9%	13%	~1%	~1%
Drinking	8%	17%	~1%	~1%
Eating	7%	2%	~1%	~1%

As you can see in the above table, the pretrained model did not perform well on the videos for Kuri.

6.4 Trained model

Shown below is the graph of the Cross entropy loss that we observed as we trained our model for RGB and Flow modalities. As expected our training loss decreased with increasing iteration numbers.





6.5 Quantitative Results on test split of dataset captured by Kuri with trained model

The results are shown for detecting activities on test splits of dataset of augmented videos.

Models	**Trained model**	
	Top 1%	
Modalities / Activities	Flow	RGB
Conversation	76%	100%

Drinking	84%	100%
Eating	64%	100%
Falling	64%	100%
Laying	96%	100%
Picking object	64%	100%
Sitting	80%	100%
Walking	84%	100%
Average Total	76.5%	100%

6.6 Quantitative Results on videos with low camera angle with trained model

After testing the videos on the test split. We moved on testing our model on more unstructured environment, that is it did not always have a sofa or a green background. For this we had to collect yet another dataset. While collecting the dataset, we kept in the mind the low camera angle that Kuri has. We collected 8 videos of nearly 5 seconds for each of the 8 labels - we call this dataset “Non-kuri low camera angle video dataset” and tested the accuracy for the output returned by our model for each video against the ground truths. The results that we obtained are shown below:

Models	**Trained**			
	Top 3%		Top 1%	
Configuration	FLOW	RGB	Flow	RGB
Conversation	25%	25%	0	12.5%

Drinking	37.5%	50%	25%	12.5%
Eating	0	62.5%	0	12.5%
Falling	75%	62.5%	50%	37.5%
Laying	100%	87.5%	87.5%	62.5%
Picking	100%	25%	100%	0
Sitting	100%	87.5%	25%	75%
Walking	50%	75%	0	37.5%
AVERAGE TOTAL	60.9375%	59.375%	35.9375 %	31.25%

Models that are trained on the dataset of activities captured by Kuri generally perform better than pre-trained. We noticed that the tested videos had a 97% successful detection with the usage of RGB. Whereas with flow the model performed, on average, up to 76.5% rate of succession. Our model performance on test and Non-kuri low camera angle video dataset are much higher than the pre-trained model from HMDB or UCF101.

7. Discussion

Since our focus is only on the activities, for the data collection we filter out the environment so that our train model concentration is on our actor and the activities that they performed.

Our model accounted for multiple activities that would be detected in a single video. In case of multiple activities being performed in a single video. All the popular activities were detected in the top 5%. The results of this can be seen in the demonstration section 8.

7.1 Observations of pretrained model on videos captured by Kuri

HMDB model was used to test videos that were captured by Kuri. In walking videos, the model was able to detect standing, turning, and running. We find that chewing was in the top 5 activities that were listed in eating videos. The majority of eating videos are labelled as clapping. The results without sliding windows (default aggregation) are worse. The results of these observations illustrated that the camera angle influence the activities that were being selected. Additionally, other activity labels that are in relation to the activity itself is often selected within the top 5 activities.

7.2 Comparison of Flow versus RGB modality

Recognition of activities using our trained model with rgb modality performed better on test split of dataset of Kuri videos.RGB modality tends to perform better on low quality videos. On the other hand, for the videos with low camera angle(8 for each label), flow modality performed better than the RGB, since optical flow is a good source of motion representation, it performed better than just looking at the differences in the pixel values.

The analysis of the videos that works best with Temporal Segment Network is flow. With Kuri dataset trained model, what we observe is that the model works better when the video is divided in RGB frames for the test data. In our controlled data gathering and training, the environment that are common in every frame is the background. Flow is only important when we are looking at dataset that has variety of background motions.

7.3. Observations of trained model on videos captured by Kuri

Some of the observations that we made for this low camera angle dataset where our model failed to produced accurate results for top 1% are justified as follows:

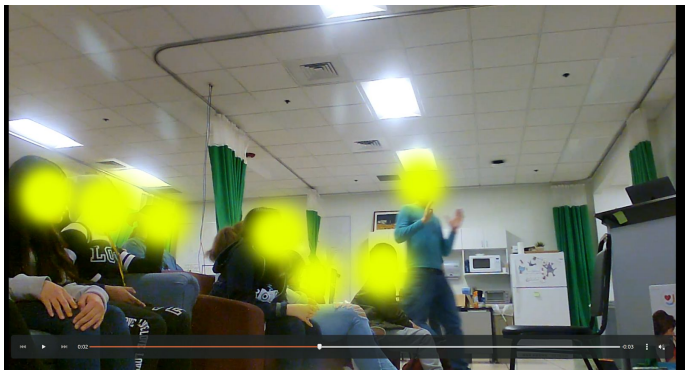
1. For video in which people were picking object, the rgb modality had 0% accuracy. The most popular label for these videos was sitting. This was the case even when sitting was not performed by the people in the video. The second popular for such videos is falling which again is not performed in the video. To reason about these observations, we need to see what low level features correspond to sitting. We will be doing this as a part of our future work.
2. In videos where walking activity was performed. The flow modality had 0% accuracy and the most popular activity that was was picking followed by falling

which again is not representative of actual activities performed in the video. We noticed that most of the videos part of this dataset had people walking directly towards or away from the camera and since our training dataset did not have any videos, our neural network failed to capture and categorize such features as walking

3. We noticed that conversation was not detected at all. Sitting was listed as the most popular and falling is the second popular. This demonstrate that our model needs to be retrain for better result.
4. Flow was not able to detect eating in both top 5% and top 1%. Instead, we find that picking object was detected as the most popular activity.

8. Demonstration

8.1 Demo 1



Labels: 6: sitting, 7: walking, 2: eating, 1: drinking, 3: falling (In the video both walking and sitting has been enacted)

8.2 Demo 2

Labels: 5: picking object, 2: eating, 4: drinking, 6: sitting, 3: falling (In the video the participants are eating, drinking and picking objects)

8.3 Demo 3

In falling videos top 1% accuracy of fall ground truth class was impacted since the model recognized laying as the final label.

9. Future Work:

9.1 Multiple labels and Untrimmed Videos

We are confidence that we can achieve the same result for untrimmed video in the future works with TSN architecture. What we wanted to continue to work on is to be able to extrapolate multiple labels for a singular video. This is less applicable to Kuri because Kuri only takes short videos.

9.2 TSN Aggregating Function - Attention Weighting

Another method that we wanted to explore was the adaptive weighting method, called attention weighting in TSN [2]. In this aggregation function, it is aim to learn a function to automatically assign an importance weight to each snippet according to the video content. The advantage is the enhancement of the modeling capacity by automatically estimating the importance weight of each snippet based on the video content. Additionally, since the attention model is based on ConvNet representations **R**, it leverages extra backpropagation information to guide the learning process of ConvNet parameter **W**. This may accelerate the convergence of training.

9.3 Retraining and adding other datasets

What we also really want to explore is combining our dataset with HMDB. This will allow the recognition to be even more accurate considering the variety in camera angles. This will also help the model from bias result or focus. Lastly, we would like to experimented more with various parameters such as learning rates and different splits to analyze their effect on the output.

10. Conclusion:

In this paper, we showed that our trained model of Temporal Segment Network architecture produced more accurate results. It is able to recognized activities for the videos captured by Kuri and other non-Kuri low camera angles videos. The performance of our trained model was better than the pre-trained network. The accuracy of the model trained on RGB modality was 97% while of the flow modality was 76.5%. In the future, we intend to retrain our model by changing the hyperparameters to further analyse the

effects of it on accuracy. We also wish to extend our architecture to work on untrimmed videos.

11. References:

[1] Mucchiani, S. Sharma, M. Johnson, J. Sefcik, N. Vivio, J. Huang, P. Cacchione, M. Johnson, R. Rai, A. Canoso, T. Lau, and M. Yim. Evaluating older adults' interaction with a mobile assistive robot. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) , pages 840–847, Sep. 2017. doi: 10.1109/IROS.2017.8202246

[2] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, Luc Van Gool, "Temporal Segment Networks for Action Recognition in Videos" in arxiv 2017. https://github.com/yjxiong/temporal-segment-networks

[3] Ian Richardson. Genesis of kuri, Mar 2018. URL https://futurism.media/Genesis-of-kuri

[4] HiddenTwoStream: https://github.com/bryanyzhu/Hidden-Two-Stream

[5] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[6] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal multiplier networks for video action recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 7445–7454.

[7] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[8] A. Kar, N. Rai, K. Sikka, and G. Sharma, "Adascan: Adaptive scan pooling in deep convolutional neural networks for human action recognition in videos," in *CVPR*, 2017.

[21] Y. Kong, Z. Tao, and Y. Fu, "Deep sequential context networks for action prediction," in CVPR, 2017.