Activity Recognition to Assist Healthcare Workers in the Emergency Department

Student: Nabil Koney-Laryea
Computer and Information Science Engineering
University of Florida
nabilkoneylaryea@ufl.edu

Mentor: Angelique Taylor
Information Science
Cornell University
amt298@cornell.edu

***Abstract*—This work seeks to build a vision based human activity recognition (HAR) system that can recognize the activities healthcare workers (HCWs) perform in different scenarios within the emergency department (ED). Using a Robot Operating System (ROS) based data collection system, video data will be collected from medical settings to create a domain specific HAR dataset. This data will be used to train pretrained computer vision models to make robust predictions about what tasks HCWs are doing in real time. Ultimately, this system could inform a robot that on how to best assist the HCWs in managing their equipment. This paper describes the process of creating the data collection system and beginning experiments with a HAR pipeline created using PyTorch.**

I. Introduction

Healthcare Workers (HCWs) rely on different equipment and medication to help them treat patients effectively. Depending on the setting in which HCWs are providing care, the amount of time it takes to access equipment may be more critical to the patient's outcome. It is common for emergency departments (EDs) to have "crash carts" that contain the equipment and medicine HCWs require to resuscitate their patients [1]. These carts have become instrumental in providing quicker care to patients that visit the ED. The goal of this work was to improve the usability of ED crash carts.

While crash carts make resources readily available to HCWs, there are other obstacles that may prevent HCWs from accessing their inventory quickly. In severe cases the ED can become very chaotic due to medical personnel operating quickly within a tight space while treating the patient. Since HCWs in the ED work in teams, poor communication could also prevent members of the team from getting the equipment they need quickly. Additionally, the dense inventory held within the crash cart can make it more difficult to find equipment. For HCWs not as familiar with the crash cart's organization, this effect could be worsened. Recent works in human computer interaction (HCI) and human robot interaction (HRI) have addressed other issues in healthcare settings by proposing and even utilizing intelligent systems. [2] proposes using autonomous robots to deliver hazardous materials, disinfect facilities, and distribute medical equipment. In some medical facilities, social robots have already been successfully deployed as companions and caregivers for children and the elderly [3].

A more intelligent crash cart made using similar methods could potentially reduce the number of obstacles that prevent HCWs from accessing equipment quickly. For example, if the crash cart could navigate itself through the room more efficiently than the medical team, it could reduce equipment retrieval time. Having a crash cart that could autonomously navigate around HCWs could also reduce foot traffic in the room and improve congestion. While these forms of intelligence could be useful, it is unlikely the crash cart would be able to assist HCWs without understanding what they are doing in real time.

By learning to recognize the actions of HCWs in real-time using human activity recognition (HAR), it might be possible to create an intelligent crash cart that can better assist HCWs. HAR is the act of recognizing the tasks an individual or a group of individuals is performing [4]. While there are several ways this could be devised, recent HAR studies have used vision, audio, or sensor data to accomplish this task. This data is typically used with deep learning algorithms to create systems that can generalize to new data. If a crash cart can utilize HAR to make intelligent decisions, it is possible that it could even assist HCWs by providing the appropriate equipment based on the recognized activity.

This work seeks to build a vision based HAR system that can recognize the activities HCWs perform in different scenarios within the ED. The first challenge in completing this goal is the amount of existing HAR datasets collected in medical contexts. The other major challenge for this project is finding pretrained models that can generalize well to new data from a medical context.

TABLE I
Commonly Used Datasets

Dataset	# of Clips	# of Classes	Ego / Exo
Kinetics400	306,245	400	Exo
UCF101	13,320	101	Exo
HMDB51	6,766	51	Exo
SSv2	220,846	174	Ego
Epic-Kitchens-100	700	397	Ego

My contributions are as follows:

- Created a data collection system with ROS that can record and store video data from multiple cameras.

- Conducted experiments with pretrained HAR models provided by PyTorch.

II. Related Works

While reviewing the literature, I recognized several trends in the datasets used to perform HAR. Table I describes the commonly used datasets from the literature. The most common datasets used were Kinetics400 [5], UCF101 [6], HMDB51 [7], SSv2 [8], and the Epic-Kitchens-100 [9] dataset [10, 11, 12, 13, 14, 15, 16, 17]. # of Clips describes the number of clips contained in the dataset. It is possible for two clips in the dataset to come from the same video. # of Classes refers to the number of possible activity classes. Ego/Exo describes whether or not the data is taken from an ego-centric (first person) or exo-centric (third person) perspective.

I also saw patterns in the pretrained models, and evaluation metrics used. The pretrained models that appeared frequently in the literature were I3D and R(2+1)D [18, 19]. It is worth noting that these are both CNN-based architectures, however, more recent studies have also experimented with vision transformer-based architectures as well [20]. Finally, the most common performance metric used was conventional (Top-1) accuracy [13, 14, 15, 16, 17, 18, 21, 22].

While some works attempt to improve upon the HAR methods of others by modifying the HAR model architectures, [23] successfully improved HAR by modifying the way they train their model. One of the core motivations for this work is that there are not as many video datasets as there are image datasets. Additionally, images and videos can help a model effectively learn different representations of the data. Following these intuitions, the authors devised a new “co-training” paradigm and pretrained a TimeSFormer model on image data and then fine-tuned on both image and video data. This work achieved a new state-of-the-art by co-training a TimeSFormer model. This training scheme could be applied to the medical domain to compensate for the lack of HAR in medical contexts.

III. Methods

A. Data Collection System

To create a more reliable vision based HAR system, it is important to have data that is representative of the environment in which the system is deployed. To meet this need, I created a system that could record and store data from multiple cameras. My initial system consisted of 2 Oak-d Pro cameras connected to a computer running Robot Operating System’s (ROS) Humble distribution on Ubuntu 22.04. I utilized existing code from the depthai-python repository to connect to the cameras simultaneously. Additionally, I adapted this code to create a ROS node for each camera and publish the video frames to their individual rosbags.

B. PyTorch Experimentation

After creating the data collection system, I began creating a pipeline capable of processing video datasets. All code was written using PyTorch’s built-in classes and wrappers for the HMDB51 dataset. First, the raw data and the data splits were downloaded from the HMBD website and wrapped using the built-in PyTorch HMDB51 class. For the training data the videos were randomly cropped, while center cropping was used for the testing data. For experimentation purposes I randomly sampled 25% of the training instances to use. Additionally, I created a validation set using 20% of the sampled training instances and all datasets were processed in batches of 4.

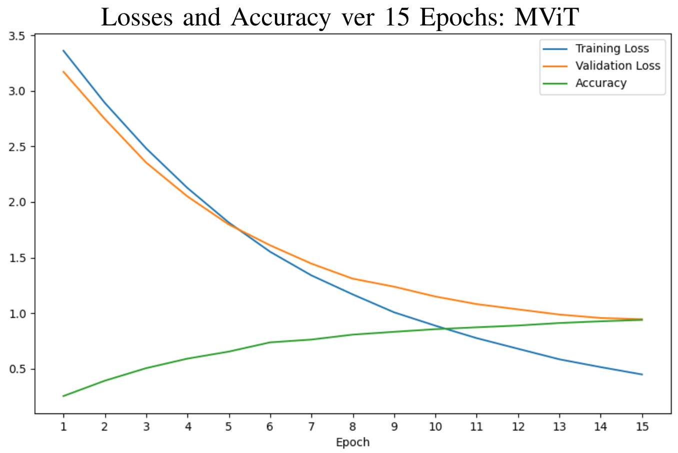

Fig. 1. Learning and performance curves for pretrained MViT over 15 epochs on HMDB51 training and validation sets.

To experiment with different architectures and hyperparameters I instantiated multiple models using PyTorch’s built-in classes. First, I instantiated X3D, Res3D, and MViT all of which used pretrained weights optimized for the Kinetics400 dataset. Since each model was trained to predict from 400 classes, I added a final linear layer that would make the model predict from 51 classes instead. After this, I got the opportunity to begin running experiments with various models. For the MViT, I used the Adam optimizer with a learning rate of 1e-5 and calculated the loss using the cross-entropy loss function.

IV. Results

As shown in Fig. 1, I created a learning curve as well as a performance curve, for MViT over 15 epochs. Overall, I found MViT was learning but did not converge within the specified number of epochs. The final training loss was .45 compared to a final validation loss of .947. Additionally, the model performed well by the end of training, showing an accuracy of 94%. Comparing the training and validation loss curves, the model is underfitting, but would converge given more training time.

V. Conclusion and Future Work

While I did not fully accomplish the goal I defined at the beginning of the summer, I made considerable progress on the project and learned a great deal about ROS and PyTorch. I created a data collection system that can publish and store data from multiple Oak-d Pro cameras using ROS. This system can later be used to collect domain specific data which could be annotated and used to fine-tune existing pretrained models. I also began running experiments on the HMDB51 dataset to explore different models and outline a vision based HAR pipeline.

There are, of course, several steps I would take to make further progress towards my initial goal if given more time. There were several topics I was exposed to in the literature that I did not have time to investigate further. Some promising topics could be comparing ego-centric versus exo-centric data, utilizing object agnostic labels for generalizability. I would also improve the data collection system with additional stress testing and experimentation with different depth and wireless camera setups. Additionally, I would conduct more experiments on domain specific data to understand how different HAR methods work in a medical context.

Another interesting idea that follows from the lack of domain specific data is to improve HAR by using multiple modalities and new training schemes. For vision based HAR it is difficult to create robust models from the small set of video data taken in medical contexts. This gap can be partially filled using image datasets to pretrain and fine-tune HAR models in addition to video data. Another way to address these shortcomings could be to train models using multiple modalities using a cross-modal training method. Future work should implement or combine similar methods to overcome the data shortage for HAR in the medical domain.

References

[1] Karim A. Diab Rasha Sawaya Gilbert Abou Dagher Eveline Hitti Jamil D. Bayram Gabrielle A. Jacquet, Bachar Hamade. The emergency department crash cart: A systematic review and suggested contents. *World Journal of Emergency Medicine*, 9(2):93, 2018.

[2] Mahdi Tavakoli, Jay Carriere, and Ali Torabi. Robotics, smart wearable technologies, and autonomous intelligent systems for healthcare during the covid-19 pandemic: An analysis of the state of the art and future vision. *Advanced Intelligent Systems*, 2(7):2000071, 2020.

[3] Maria Kyrarini, Fotios Lygerakis, Akilesh Rajavenkatanarayanan, Christos Sevastopoulos, Harish Ram Nambiappan, Kodur Krishna Chaitanya, Ashwin Ramesh Babu, Joanne Mathew, and Fillia Makedon. A survey of robots in healthcare. *Technologies*, 9(1), 2021.

[4] Md. Milon Islam, Sheikh Nooruddin, Fakhri Karray, and Ghulam Muhammad. Human activity recognition using tools of convolutional neural networks: A state of the art review, data sets, challenges, and future prospects. *Computers in Biology and Medicine*, 149:106060, 2022.

[5] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[6] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012.

[7] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition, 2011.

[8] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The ”something something” video database for learning and evaluating visual common sense, 2017.

[9] Dima Aldamen, Davide Moltisanti, Evangelos Kazakos, Hazel Doughty, Jonathan Munro, William Price, Michael Wray, Tobias Perrett, and Jian Ma. Epic-kitchens-100, 2020.

[10] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition, 2022.

[11] Akash Kumar and Yogesh Singh Rawat. End-to-end semi-supervised learning for video action detection, 2022.

[12] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Mingqian Tang, Zhengrong Zuo, Changxin Gao, Rong Jin, and Nong Sang. Hybrid relation guided set matching for few-shot action recognition, 2022.

[13] Jay Patravali, Gaurav Mittal, Ye Yu, Fuxin Li, and Mei Chen. Unsupervised few-shot action recognition via action-appearance aligned meta-adaptation, 2021.

[14] Yuan Zhi, Zhan Tong, Limin Wang, and Gangshan Wu. Mgsampler: An explainable sampling strategy for video action recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1493–1502, 2021.

[15] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Learning self-similarity in space and time as generalized motion for video action recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13045–13055, 2021.

[16] Shizhe Chen and Dong Huang. Elaborative rehearsal for zero-shot action recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13618–13627, 2021.

[17] Jaeyoo Park, Minsoo Kang, and Bohyung Han. Class-incremental learning for action recognition in videos. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13678–13687, 2021.

[18] Zhensheng Shi, Ju Liang, Qianqian Li, Haiyong Zheng, Zhaorui Gu, Junyu Dong, and Bing Zheng. Multi-modal multi-action video recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13658–13667, 2021.

[19] Shuaicheng Li, Qianggang Cao, Lingbo Liu, Kunlin Yang, Shinan Liu, Jun Hou, and Shuai Yi. Groupformer: Group activity recognition with clustered spatial-temporal transformer, 2021.

[20] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feicht-

enhofer. Multiscale vision transformers, 2021.

[21] James Hong, Matthew Fisher, Michaël Gharbi, and Kayvon Fatahalian. Video pose distillation for few-shot, fine-grained sports action recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9234–9243, 2021.

[22] Rasha Friji, Hassen Drira, Faten Chaieb, Hamza Kchok, and Sebastian Kurtek. Geometric deep neural network using rigid and non-rigid transformations for human action recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12591–12600, 2021.

[23] Bowen Zhang, Jiahui Yu, Christopher Fifty, Wei Han, Andrew M. Dai, Ruoming Pang, and Fei Sha. Co-training transformer with videos and images improves action recognition, 2021.