Interim Report: Workout Recognition Using 2D and 3D Convolutional Neural Networks

James Peralta, Jeffrey Boyd
Department of Computer Science
University of Calgary
Calgary, Canada
{james.peralta, jboyd}@ucalgary.ca

***Abstract*—Physical activity has proven to be effective in the primary and secondary prevention of a variety of chronic diseases including cancer and cardiovascular disease. Although exercise provides many incentives, people have trouble motivating themselves to maintain their routine. Workout loggers are used to track progression, and have shown to motivate users by showing them their improvements before concrete results were visible in the body. Current workout loggers require users to manually enter data into their phones, or write it down into a notebook. To automate this, our system will analyze gym footage using computer vision, recognize the workouts being performed, and count repetitions. Convolutional Neural Networks (CNNs) has become the state of the art for many video recognition tasks, which has encouraged us to apply them to this specific problem.**

Keywords—Workout Recognition, Machine Learning, Transfer Learning, Convolutional Neural Networks

I. Introduction

In 2017, nearly 50% of deaths in Canada were caused by cancer and heart disease [1]. In fact, 44% of adults over the age of 20 have at least 1 of 10 common chronic diseases [2]. To help reduce these chronic diseases, we must invest in preventive measures such as empowering our society into a healthy lifestyle. However, creating the habits to routinely engage in physical activity is a long-term process that requires a lot of personal motivation. Only 20% of people succeed in long-term weight loss maintenance [3].

A promising area of technology - persuasive technology, is designed to influence and change human behaviors. Emerging persuasive technology in the fitness sector has already shown to improve the health and fitness practices of its users [4]. Within the space of persuasive technology, activity tracking is one of the most prevalent strategies. Users of activity trackers that monitor their workout progress commented on how this technology had motivated them and helped them make durable changes in their lifestyle [5]. Activity Trackers can be used for more simple data, such as counting steps but fail to capture advanced data, such as repetitions of workouts. In this paper, we focus on a specific type of activity tracking, workout logging.

Although workout loggers offer plenty of value to their users, they have shown very poor retention rates. [6] found that within the first two weeks, 62% of users that downloaded a workout logging app stopped using it. This was because manual entry of data can be too repetitive and tedious, eventually leading to these technologies to go unused [4]. There have been attempts at removing the step of manually entering data through wearables [7][8][9], computer vision [10], and adding sensors onto equipment [11].

Computer vision is a promising solution because it can capture a richer set of spatial features. Wearables lose accuracy depending on where the sensor is placed. For example, a sensor placed on the wrist will not detect leg exercises such as a leg press because the hand is stationary during this exercise. Furthermore, adding sensors to equipment will not be able to track bodyweight exercises such as pushups where the user is not using any equipment. Not only can computer vision avoid these pitfalls, it is also the only solution that can be implemented as a single-point solution, which will avoid instrumenting many users or many machines.

II. Explanation of the Problem

A workout is a routine performed in the gym which consists of several different exercises aimed to train certain groups of muscles. For example, a workout may consist of exercises such as squats, lunges, and leg press which all target the legs. The sets and repetitions are used to describe the number of times you perform an exercise. Each exercise is done for a certain amount of repetitions which we call a set and a full repetition is one complete motion of an exercise.

Overtime, a trainee will eventually need to increase the amount of weight they need to lift to continue to gain muscle. Typically, if a trainee is able to easily perform a certain amount of sets and reps at a specific weight they should move up. This results in them constantly being required to remember what happened last workout, so they are able to move up in weight if needed. Many often struggle recalling their last workout and use a workout logger to assist them. A simple workout logger will track the amount of sets and reps a trainee performs for an exercise for each given day.

Workout logs these days can still be manually written using a pen and paper, or manually typed into phone apps such as FitNotes, Jefit, and Gymbooks. As all commercial products require manual entry, in this paper we attempt to construct software that performs automatic workout logging. We define a workout logger to be completely automated if it is able to detect the type of workout, count the repetitions, the amount of weight

a trainee has performed, and save these results for future use without any user intervention.

All computer vision problems have a common set of challenges we will need to overcome (see Table 1 and Figure 1). Although we would like to solve most of these challenges, some of these challenges we will limit at first. This is done to control the scope of our problem, but some challenges do not apply to the gym environment anyways. Since we have to build the dataset ourselves, we will limit the viewpoint variation by only proposing to solve one view. If we were to collect data points of different angles, we may spend a considerably longer amount of time collecting data. If we are able to construct a classifier and have enough time to come back and add more data from different angles, we will do so.

We will allow for any amount of background clutter to appear in the videos. For example, any amount of trainees are allowed to pass behind another trainee performing a workout. Trainees are also free to perform the workouts they choose and may have been taught to do the workout in a different way. This will lead to intra-class variation in workouts that we will not limit. Other challenges do not present themselves in our gym environment such as scale variation, occlusion, and illumination. This is because we set up the cameras in front of each machine so there will be nothing blocking the trainee. Furthermore, a user won't stand 30 feet away from the squat rack he is using and gyms are usually very well lit for safety.

III. Proposed Solution

Our approach is a pipeline of CNNs as follows. The raw frame is passed into a person detection network to detect humans in the field of view. Once a human is located we will crop a rectangle around them to focus in on the user. This frame will be fed into a Neural Network that will predict if the person in the frame is performing an exercise. If the person is performing an exercise, we will begin creating a clip using all subsequent frames until the user stops performing the exercise. Lastly, we will input this clip into a classifier that will predict the workout being performed and the number of repetitions.

TABLE I. Challenges common to all computer vision problems and their descriptions.

Challenges	***Descriptions***
Viewpoint variation	Viewing an object from a different angle can completely change its appearance.
Scale variation	Viewing an object up close will make it look very different from far away.
Occlusions	The object of interest is hidden from view or partially cut off.
Illumination	Objects cans look dramatically different in low-light vs normal.
Background clutter	Having too much noise in the background can distract the Neural Network.
Intra-class variation	The variation between data points in the same class.

Fig. 1. Illustraitions of the different type of variations found in images.

Creating this system will involve phases for data collection, data labelling, data preparation, data preprocessing, data exploration, and then building the classifiers. We will use CNNs as our classifiers due to the improvements in speed and performance over feature-based approaches. W will use Scikit-learn to implement traditional machine learning techniques and Keras to implement neural networks.

A. Data collection

Our goal is to collect a diverse dataset that will allow us to train a classifier to overcome all of the challenges within the scope of our project. We invite participants of different genders, ages, level of weight lifting experience, and physical shape. We have arranged to book an athlete gym at the University of Calgary. We will strategically place cameras around the gym to capture each machine. Participants will be invited for a 30-90 minute workout session where they will perform 6 different exercises of their choice. Each camera will only store footage when it detects a participant in the field of view. This will create clips that start when a participant enters and ends when there is no longer any participants in the view. 60 participants, performing 3 sets per exercise will leave us with 1080 exercises captured on video.

B. Data labelling

After building up the dataset of exercises, we will have to label them before training our Neural Network. We will use GymCam's annotation tool which is optimized for labelling workouts and their repetitions. For each clip, we will draw a bounding box around a user when they begin performing a workout and remove it when they finish the set. Along with this

bounding box we will label the exercise they performed along with the number of repetitions. This will create a dataset that can be used for exercise segmentation, exercise localization, exercise recognition and repetition counting.

C. Data exploration

Each data point will be a video of a user performing a set of an exercise. If our data collection goes as planned, we will have a dataset of 1080 videos. We will aggregate basic statistics of our dataset such as the number of classes, videos per class, repetitions throughout the whole dataset, and the number of repetitions per video. The amount of videos per class is specifically important because we need to ensure that we minimize the class imbalances in our training, validation, and test sets. We also want to quantify the difficulty of our dataset. We will do so by calculating the amount of background clutter and illumination of our dataset. We can calculate the amount of background clutter by using the YOLO V3 network [12] to locate and report the number of objects recognized in each frame. We can calculate the illumination of each frame by calculating the sum of R+G+B for all pixels divided by 3 then by the total number of pixels. This will give a value between 0 (dark) and 255 (bright).

D. Data preparation

We will store the data on a Google Drive with folders for the training, testing, and validation datasets with a split of 65/15/20 respectively ensuring the classes are balanced between each set. We will want this data to be stored in a format that allows for fast retrieval so we will explore storing them in HDF5 or NPY formats. Also, since our videos are broken down into sets and a trainee may perform multiple sets of the same exercise. We need to ensure that sets of the same trainee performing the same exercise is not spread across the three folders as it will leak information between the datasets. Lastly, we will have a master csv file which contains the file name for each video along with its corresponding labels.

E. Data preprocessing

To increase the size of our dataset we will perform some data augmentation such as flipping videos horizontally, adjusting the brightness, saturation, and randomly zooming in some videos to simulate scale variance. All of these operations can be applied using OpenCV libraries. Furthermore, we will normalize the dataset by performing a standard scaling technique which subtracts the mean of the training data and divides by the standard deviation. We will also experiment with some low level image processing operations to help improve the performance. These techniques include histogram equalization, sharpness, and low and high contrast. These image processing operations have already been shown to improve the accuracy of Pose Estimation Systems [13].

F. Creating a person detector

The importance of using a person detector in our algorithm is to focus in on the region of interest where a user may be performing a workout (see Figure 2). Doing so will remove the background noise and center the trainee, which will remove some variance in the dataset. We will use the YOLO V3 network as our person detector. YOLO V3 is built on top of a CNN that takes as input a full image and returns to bounding boxes where people are located with the probabilities for each region. This network was chosen because it is very fast which allows us to integrate it into our real-time system.

Fig. 2. The YOLO V3 network recognizing when a trainee begins using a machine and notifying us with a green light.

G. Recognizing workouts

To recognize the workouts, we will attempt two different approaches. The preferred approach would be to use a 3D CNN because it provides an end-to-end system without having to chain classifiers. If we do not have a large enough dataset to train a 3D CNN from scratch, we may opt to use a pre-trained open-sourced model such as Google Deepminds I3D as our feature extractor [14]. This technique would work by taking a window of contiguous frames from the video stream and passing this block of frames through I3D which will generate a feature vector. Afterwards we can train a classifier using supervised learning with the I3D feature vectors as input. Possible classifiers could range from a simple SVM to more complex solutions such as a DNN if needed.

The second approach is to use a pose recognition algorithm built using a 2D CNN. This pose recognition system will be applied on each frame to extract key points of the trainee's body parts. The body parts we are interested in are the wrists, elbows, shoulders, knees, and feet. This will allow us to track how certain parts of their body move overtime and we can use this information to detect the type of workout they are performing. Recognizing workouts will be a classification task where we are predicting the class of workouts as our labels using key points over a window of time as the features. Classifiers could range from an SVM that analyzes the whole window at once, to an LSTM that is able to capture how the key points progress over time. The key insight in why this could work is that it will remove the noise that is inherent from images and reduce the size of the feature vector. For example, if we were to track 10 key points in the body, that will leave us with a vector of 10x2 for x and y coordinates of each body part instead of 256x256 for an image.

H. Counting repetitions

When a trainee is performing a workout, they are usually not moving side to side but instead moving up and down in some fashion. For example, a trainee performing bicep curls will have their wrists moving up and down and a trainee performing squats will have their torso move up and down. Therefore if we can recognize when the trainee is on the up or down portion of their repetition, we can put the up and down portions together and confidently count that as a repetition. As we have two different approaches to recognizing workouts, we also have two ways we could detect these ups and downs.

If we went with the 3D CNN approach, we would have to create a classifier to calculate the amount of repetitions in the video. [15] found that using a classifier over a regressor for this task achieved considerably better results. The classifier will use optical flow to quantify the motion in the video. Our classifier will use these optical flows as input and predict which portion of the repetition a user is doing. If we went with the key point estimator, it would be a lot simpler to count the repetitions and we could use calculus equations to do so. We can sample the key points periodically to calculate the rate at which these key points are changing across the y-axis overtime and see if they are increasing or decreasing in value. If they are increasing, the trainee is on the up portion and down otherwise.

IV. Evaluation of solution

Similar to [10], we will separately evaluate the performance of exercise recognition and repetition counting. For all the models in this experiment that tackle classification problems, accuracy is used as a primary metric for model performance. Accuracy is defined as the total number of correct predictions divided by the total number of predictions. Although accuracy is a simple yet effective way to evaluate model performance, it may not work well with skewed dataset. If our dataset is very skewed, F1 score will be used alongside accuracy to give a better understanding of the true performance of the models. The F1 score considers both recall and precision and provides a good estimate for skewed classification datasets [16]. Before, precision and recall can be defined, confusion matrices need to be introduced.

Confusion matrix further splits the prediction results into 4 classes, namely True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). A FP is when a datapoint does not belong to a class but is classified to be in that class and a FN is when a datapoint belongs in the class but is classified to not belong in the class. TP and FP is when the classifier correctly classifies a datapoint to be part of a class or not part of a class. This can give a good insight into the predictions by showing where the model is getting its right and wrong predictions. The precision is simply TP divided by (TP + FP), and it is used for cases where the number of false positives needs to be limited [16]. Similarly, recall is defined as TP divided by (TP + FN) and it is used where false negatives needs to be controlled [16]. When generating these metrics we will perform stratified ten fold cross validation.

There will be two areas where we need to evaluate our classifiers. When choosing hyper parameters for our classifiers and at the end of the study with our finalized classifiers to report the final results. Therefore as mentioned in the data preparation phase, we have created training, validation, and test sets with a split of 65/15/20 percent of the whole dataset respectively. The validation set will be used when searching for the optimal hyper parameters to use. For example, CNNs require tuning of the number of convolution layers, filters in each convolutional layer, kernel sizes, and kernel stride lengths. We will train our classifiers on the training data and test them using the validation data while aggregating the metrics specified above. The results of the validation set will inspire tuning of the hyper parameters, and we will iterate until we are satisfied with the results. Lastly, we will evaluate our final classifiers with the test set and report on these results.

V. Related work

A. Using optical flow

The current state of the art for this problem is Gymcam [10]. GymCam used optical flow to generate a feature vector of 27 features. This feature vector was passed along a pipeline of three separate multilayer perceptrons. With this procedure they were able to obtain an accuracy of 84.6% for exercise segmentation, 93.6% for exercise recognition, and count the number of repetitions within +/-1.7 on average. Other related contributions were for similar video analysis tasks; video classification and human gesture recognition.

B. Feature based versus neural networks

Patsadu et al. evaluated feature-based and neural network classifiers for human gesture recognition [17]. In their study, they used a Kinect camera to extract a body-joint position vector containing the locations of 20 joints including the head, hands, and feet. Using this vector as input into a classifier, they classified between three actions; standing, sitting down, and laying down. They observed the performance of multiple classifiers such as Back Propagation Neural Networks, SVMs, Decision Trees, and Naive Bayes. Out of every classifier, neural networks performed the best

C. Using 2D Convolutional Neural Networks

Karpathy et al. surveyed the performance of two dimensional convolutional neural networks (2D CNNs) for video classification [18]. 2D CNNs avoid the feature extraction phase required by Gymcam and feeds raw frames as input into the network. This simplified the pipeline by removing the need to generate custom features and was shown to have significant performance gains over the feature-based approaches. Since 2D CNNs take in a single frame as input, it can only encode spatial features. This can be a problem due to the temporal nature of videos. As part of their contribution, they proposed three ways of encoding temporal information using 2D CNNs (Figure 3).

D. Using 3D Convolutional Neural Networks

Tran et al. introduced a three dimensional Convolutional Neural Network (3D CNNs) they named, C3D [19]. The 3D CNN takes videos as input and encodes both spatial and temporal information without the need to perform workarounds

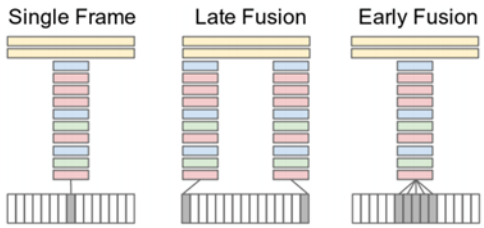

Fig. 3. The single frame approach only encodes information from a single frame, late fusion encodes temporal information from two frames which are 15 frames apart, and the early fusion encodes temporal information from several contiguous frames. Figure from Karpathy et al.

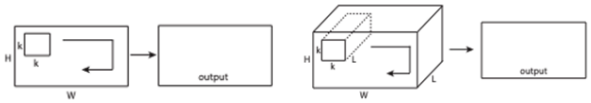

Fig. 4. 2D Convolutions (left) vs. 3D Convolutions (right). Figure from Tran et al.

such as early fusion. The difference between a 2D CNN and a 3D CNN is that the convolution operations use two dimensional and three dimensional kernels respectively (Figure 4). They went on to show how 3D CNNs were better than 2D CNNs at preserving temporal information and outperformed 2D CNNs on various video analysis tasks. More interestingly, 3D CNNs we're shown to run 2x faster than optical flow solutions.

VI. CONCLUSION

With the current limitations in workout logging technologies, our project explores the application of 2D and 3D Convolutional Neural Networks to automate this. Our work involves the full machine learning pipeline from data collection, data preparation, and eventually to creating the classifiers. We have been approved by the ethics board to setup cameras in a gym and collect data in an unconstrained environment. The data we collect will be of users performing workouts and we will use these data points to create a dataset for exercise segmentation, exercise localization, exercise recognition, and repetition counting. Once are dataset is built we will create a system that contains multiple neural networks, all performing separate tasks in a pipeline to achieve the final goal. Once our full system is created, we will perform stratified ten fold cross validation to obtain the accuracy and F1 scores and report the final results.

REFERENCES

[1] Statistics Canada, "Leading causes of death, total population, by age group," *https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1310039401,* vol. , no. , p. , 2017.

[2] Goverment of Canda, "Prevalence of Chronic Diseases Among Canadian Adults," https://www.canada.ca/en/public-health/services/chronic-diseases/prevalence-canadian-adults-infographic-2019.html, vol. , no. , p. , 2019.

[3] R. R. Wing and S. Phelan, "Long-term weight loss maintenance.," *The American journal of clinical nutrition,* vol. 82, no. 1 Suppl, pp. 222S-225S, 2005.

[4] A. Ahtinen, E. Mattila, A. Väätänen, L. Hynninen, J. Salminen, E. Koskinen and K. Laine, "User experiences of mobile wellness applications in health promotion: User study of wellness diary, mobile coach and SeltRelax," in *2009 3rd International Conference on Pervasive Computing Technologies for Healthcare - Pervasive Health 2009, PCTHealth 2009*, 2009.

[5] T. Fritz, E. M. Huang, G. C. Murphy and T. Zimmermann, "Persuasive technology in the real world: A study of long-term use of activity sensing devices for fitness," in *Conference on Human Factors in Computing Systems - Proceedings*, 2014.

[6] D. Ledger, D.; McCaffrey, "How The Science of Human Behavior Change Offers The Secret to Long-Term Engagement," *http://endeavourpartners.net/assets/Endeavour-Partners-WearablesWhite-Paper-20141.pdf ,* 2016.

[7] D. Morris, T. S. Saponas, A. Guillory and I. Kelner, "RecoFit: Using a wearable sensor to find, recognize, and count repetitive exercises," in *Conference on Human Factors in Computing Systems - Proceedings*, 2014.

[8] M. Muehlbauer, G. Bahle and P. Lukowicz, "What can an arm holster worn smart phone do for activity recognition?," in *Proceedings - International Symposium on Wearable Computers, ISWC*, 2011.

[9] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors (Switzerland),* vol. 16, no. 1, 18 1 2016.

[10] R. Khurana, K. Ahuja, Z. Yu, J. Mankoff, C. Harrison and M. Goel, "GymCam," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies,* vol. 2, no. 4, pp. 1-17, 27 12 2018.

[11] H. Ding, L. Shangguan, Z. Yang, J. Han, Z. Zhou, P. Yang, W. Xi and J. Zhao, "FEMO: A platform for free-weight exercise monitoring with RFIDs," in *SenSys 2015 - Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 2015.

[12] J. Redmon, A.Farhadi, "YOLOv3: An Incremental Improvement", 2018

[13] J. Pederson, "Improving the Accuracy of Intelligent Pose Estimation Systems Through Low Level Image Processing Operations", 2019

[14] J. Carreira, A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset", 2018

[15] O. Levy, L. Wolf, "Live Repetition Counting", 2015

[16] A. Müller and S. Guido, "Introduction to machine learning with Python: A Guide for Data Scientists (1sted.)", 2016.

[17] O. Patsadu, C. Nukoolkit and B. Watanapa, "Human gesture recognition using Kinect camera," in *JCSSE 2012 - 9th International Joint Conference on Computer Science and Software Engineering*, 2012.

[18] 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.

[19] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," 1 12 2014.