

Exercise Motion Classification from Large-Scale Wearable Sensor Data Using Convolutional Neural Networks

Terry Taewoong Um¹, Vahid Babakeshizadeh² and Dana Kulić¹

Abstract—The ability to accurately identify human activities is essential for developing automatic rehabilitation and sports training systems. In this paper, large-scale exercise motion data obtained from a forearm-worn wearable sensor are classified with a convolutional neural network (CNN). Time-series data consisting of accelerometer and orientation measurements are formatted as *images*, allowing the CNN to automatically extract discriminative features. A comparative study on the effects of image formatting and different CNN architectures is also presented. The best performing configuration classifies 50 gym exercises with 92.1% accuracy.

I. INTRODUCTION

In recent years, several computer [1] and robotic [2, 3] systems have been proposed to coach users during rehabilitation or physical training. The coaching systems demonstrate exercises to the users through videos or robot motions and evaluate the users' movements to determine if they are performing the exercises correctly. However, the ability of the current coaching systems to recognize human movements is limited to a small number of distinctive movements. To enable these coaching systems to be used for a wider array of exercise regimens in both sports training and rehabilitation, it is necessary for the systems to be equipped with better recognition algorithms to classify a large number of exercises.

Classification of human motions, called human activity recognition (HAR), has been widely investigated in the past decade, based on vision data [4], 3D data [5], wearable sensor data [6], etc. Wearable sensors provide a convenient way to collect data without the need for extensive installation and setup, thus, they are often preferred when data should be collected in real life settings. Moreover, the rapid growth of the wearable device market provides access to a large amount of wearable sensor data. In this research, we take advantage of large-scale wearable sensor data to enhance classification ability.

In large-scale data classification problems, convolutional neural networks (CNN) have demonstrated excellent performance over the past decade (e.g. [7]). Although CNNs were originally proposed for static 2D images [8], they have been successfully extended to other nonstatic domains, e.g., speech recognition [9] and sentence classification in natural language [10]. In these works, CNNs learn useful features

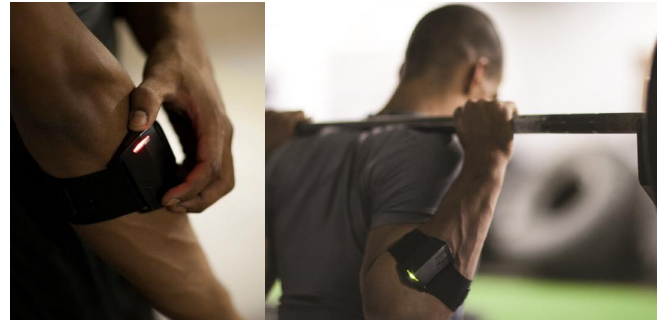


Fig. 1. The wearable band, PUSH [11], is a device for measuring performance of gym exercises with an accelerometer and gyroscope.

directly from raw data, i.e., no hand-crafted features are needed. As these researches have shown that learned features from CNNs can outperform classical hand-crafted features in various domains, we may expect that CNNs can also learn useful features from raw wearable sensor data and use these features for successful classification, if large-scale data is available.

In this research, exercise motion data obtained from a forearm band, PUSH [11] (Fig. 1), are classified using CNN. The exercise motion data, collected from gym exercises performed by athletes, are considerably more difficult to classify than usual daily activity data, since many gym exercises have similar arm movements (e.g. front and back lat pull-down). Moreover, there are a large number of unique exercise classes (more than 300), which is a significantly larger number of classes than usual HAR with a single wearable sensor [6].

The challenging large-scale classification problem is tackled by using CNN in this research. We employed CNNs rather than recurrent neural networks (RNNs) [12], which are usually employed for sequence data, because exercise repetition (rep) data are relatively short (mostly less than 4 seconds), and therefore, may not need long-term memory for training. By creating image-like 2D data from the raw acceleration and orientation data, CNN efficiently learns discriminative features and a classifier for differentiating between the exercises. The proposed approach can be used for computer or robotic exercise coaching systems as well as for human trainers to assist with training management and performance monitoring.

The outline of the paper is as follows: First, related research on classification with single arm-worn wearable

¹Terry Taewoong Um and Dana Kulić are with the Department of Electrical & Computer Engineering, University of Waterloo, Waterloo, ON, N2L 3G1, Canada. (email: terry.t.um@gmail.com; dana.kulic@uwaterloo.ca)

²Vahid Babakeshizadeh is with Push Inc., Toronto, ON, M5B 2G9, Canada. (email: vahid@trainwithpush.com)

sensor data are briefly surveyed in Section II. In Section III, the CNN architecture used for classifying wearable sensor data is described, including a description of the dataset. Classification results with various CNN architectures and input formats are presented in Section IV. Finally, findings from the research and directions for future work are summarized in Section V.

II. RELATED WORK

Although most research on HAR with wearable sensors uses multiple sensors attached to different body parts, there have been several works based on a single arm-worn sensor. In [13], 5 activities (walking, running, cycling, driving, and sports) are classified with 72.3% accuracy using decision tree, naive Bayes and naive Bayes with PCA. In this work, 19 features are first extracted from time, frequency and spatial domains and classified with the aforementioned classifiers. In [14], 5 activities (running, walking, lying, standing and sitting) are classified with 91.1% accuracy by using 13 statistical features from time and frequency domains and a multilayer perceptron classifier.

Recent works have attempted to classify more than 5 activities based on a single arm-worn sensor. [15] classifies 9 daily activities (brushing teeth, washing dishes, watching TV, etc.) with 82.8% accuracy by using 12 statistical features from an accelerometer and a support vector machine classifier. They also increase the accuracy to 90.2% by using an additional 2 features from a temperature sensor and altimeter. Lastly, [16] shows that 26 activities including ambulation, cycling, sedentary poses, and others, can be classified with 84.7% accuracy using 13 statistical features and a support vector machine classifier.

The most relevant previous work to our research is *RecoFit* [17], a wearable sensor to recognize and count repetitive exercises. Similar to *PUSH*, *RecoFit* measures 3-axis accelerometer and gyroscope data from the upper forearm. The collected data are first smoothed with a Butterworth filter and 60 statistical features are extracted from each of 5-second sliding windows. The extracted features for each window are classified using a multiclass support vector machine, and finally, the prediction is made by voting with the prediction results from all windows.

The *RecoFit* system gives 100%, 99.3%, 98.1%, 96.0% accuracy for 4, 4, 7, 13-class classification, respectively. It is an impressive result, however, there are several points to remark for comparison with our research. First, they collected data from a controlled experiment environment while we used real-world data obtained from actual training sessions of athletes. Although each exercise session used in [17] lasts at least 20 seconds, our data have many sets of exercises shorter than 20 seconds, and 11.5% of the data contains fewer than four repetitions. As presented in Section IV, few-rep data are more difficult to classify than many-rep data.

Furthermore, our research classifies 50 classes that contain many similar exercises, while the *RecoFit* research classifies at most 13 classes, which are relatively distinctive (See

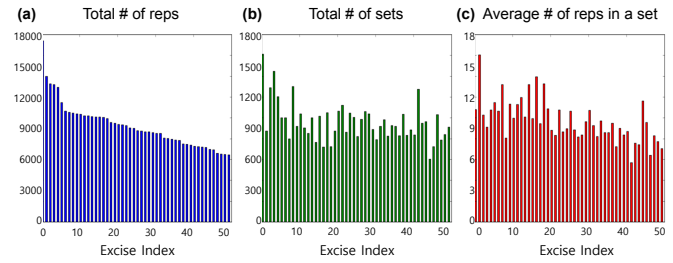


Fig. 2. *PUSH* data consist of 449,260 reps from 49,194 sets of exercises. The above graphs show (a) the total number of reps, (b) the number of sets and (c) the average number of reps in a set, for each exercise. The index of the top-50 exercises are sorted by the number of reps as in (a).

Appendix). For example, 7 kinds of bench press and 4 kinds of squat, which could be confusing even to humans, are considered as distinct labels in our experiment. On the other hand, the 13 exercises in the *RecoFit* research are relatively well-spread over a range of body-part exercises. Hence, the exercise motion classification in this research can be regarded as a more challenging and realistic problem.

III. CNN FOR WEARABLE SENSOR DATA

A. *PUSH* dataset

PUSH [11] is a forearm-worn wearable device for measuring athletes' exercise motions. *PUSH* is actively being used by over fifty professional sports teams; athletes have been voluntarily collecting exercise motion data for enhancing their training performance. From the privately available dataset provided by *PUSH Inc.*, we used a subset of the data consisting of the top 50 most frequently-performed exercises out of more than 300 exercises for this research (See Appendix).

The top-50-exercise dataset consists of 49,194 sets and 449,260 reps of exercises obtained from 1441 male and 307 female athletes (Fig. 2). Each set consists of one or more reps of a single exercise, self-labeled by the athlete. Note that segmentation is performed by the device in preprocessing to extract each rep. Training and prediction process are performed based on each rep data. The number of sets and reps are necessarily unbalanced for each exercise; the exercise that has the most reps is 0:Standing triceps extension with dumbbell (1613 sets, 17380 reps) while the exercise that has the fewest reps is 49:Wide-grip bench press with barbell (772 sets, 6133 reps). The number of reps in a set is also necessarily varied, from 1 to 254 reps, although 94% of the data have under 20 reps in a set. The lengths of the reps are also varied, but 99% are shorter than 784 samples, which is equivalent to 3.92 seconds.

Since each rep has a different length, while CNN takes fixed-size data, each rep is zero-padded, i.e., filled by zeros at the tail to make the length 784 samples. The reps that have more than 784 samples are simply cropped to have 784 samples. Zero-padding is often used for solving the variable-length problem in time-series data (e.g. [10]). Since zero-padding preserves the length information, it is helpful

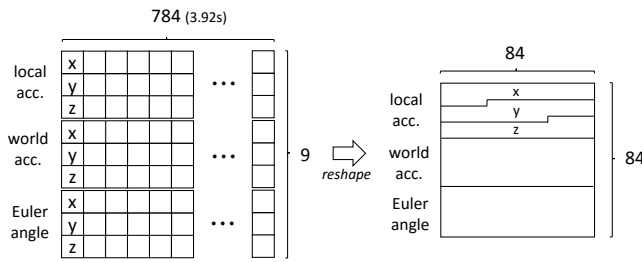


Fig. 3. A 84 by 84 square image is created by reshaping the 9 by 784 raw data.

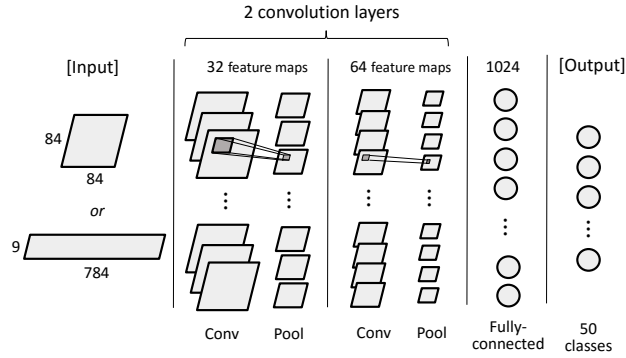


Fig. 4. The baseline CNN architecture has two convolutional layers followed by a fully connected layer. In the experiments, additional convolutional layers and fully connected layers are examined.

for discriminating between short duration and long duration exercises.

The device is tightly bound on the upper forearm and measures accelerations and orientations with a built-in accelerometer and gyroscope, respectively. As a result, time-series data with 9 features - (Acc_x , Acc_y , Acc_z) in the local frame, (Acc_x , Acc_y , Acc_z) in the world frame and $EulerAngle_x$, $EulerAngle_y$, $EulerAngle_z$) in the world frame - are obtained with a 200 Hz sampling rate.

B. CNN Architecture

CNN was originally developed for 2D image recognition [18]. Rather than using hand-crafted features, CNN directly learns features from raw pixels without any prior knowledge about features. CNN sweeps convolution and pooling windows over the image to create various feature maps. The convolution window convolves pixel values in a local region, called a receptive field, to determine the spatial correlation between them. After that, the pooling window downsamples the convolved data by, e.g., selecting maximum values only.

To apply CNN to the human motion data, we first need to create 2D *images* from the raw sensor data. We create 2D images with three different approaches: (1) regard the 9 by 784 time-series data as a rectangular 2D image, (2) treat the three different feature groups - local accelerations, world accelerations, and Euler angles - as RGB channels in an image and create a 3 by 784 by 3 tensor, (3) reshape the 9 by 784 time-series data into an 84 by 84 square matrix

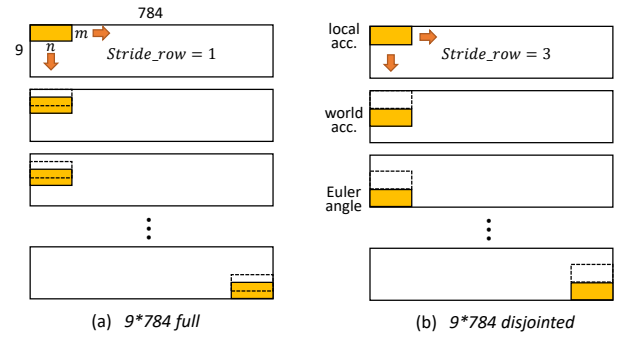


Fig. 5. If the stride size along the y-axis is smaller than three as in (a), the windows will convolve elements from different feature groups, e.g., (Acc_y , Acc_z , $EulerAngle_x$). On the other hand, if the stride size along the y-axis is three as in (b), the windows will convolve elements within the same feature group.

(Fig. 3). Note that different choices of image formatting lead to different convolutions with different neighboring elements, which may include convolutions between irrelevant elements.

There are several hyperparameters to be chosen for the CNN architecture; depth and width of the CNN architecture, convolution and pooling window sizes, their stride sizes, activation functions, etc. The baseline CNN architecture that is used for the experiments is presented in Fig. 4. In the baseline CNN architecture, two convolution layers, which have 32 and 64 feature maps, are followed by a fully connected layer which has 1024 nodes. Rectified units [19] are employed as activation functions and softmax functions are used for evaluating the final 50 output node values. Experiments with different CNN architectures and input formatting will be presented in the next section.

For optimization, the Adam optimizer [20] is employed with a learning rate of 0.0005. Also, dropout [21] with a probability of 0.5 is applied to each layer to avoid excessive dependency on certain nodes. Cross-entropy loss is employed for learning one-hot-encoded 50 exercises. Although the CNN is learned based on rep-based predictions, we also provide prediction results for each set by taking a majority vote of the rep-based predictions in each set. It is expected that set-based predictions give better results than rep-based predictions because they include the benefit of majority voting.

IV. EXPERIMENTS

The experiments are performed with an open source deep learning library, Tensorflow [22]. Built-in Adam optimizer and dropout functions in Tensorflow are used for learning CNN models. The computer used for the experiments is equipped with 3.6 GHz quad-core processors and NVIDIA GTX980 GPU.

A total of 404,200 training reps from 44,240 sets are trained with 100-sized minibatches. Also, 45,060 reps from 4,954 unseen sets serve as the test set. Note that reps from a single set belong to either the training or the test set, i.e., training and test sets are from separate sets. The data are

TABLE I

EFFECTS OF VARIOUS IMAGE FORMATTINGS AND CONVOLUTIONS: IT APPEARS THAT TREATING DIFFERENT FEATURE GROUPS AS DIFFERENT CHANNEL ($3*3*784$) SHOWS THE BEST PERFORMANCE.

| - | Input | 84*84 | | | 9*784 full | | | 9*784 disj | | | 3*784*3 | | |
|--------------------------|-------|-------|--------|-----------------|------------|--------|-----------------|------------|--------|-----------------|--------------|--------------|-----------------|
| | - | Win | Stride | Dim | Win | Stride | Dim | Win | Stride | Dim | Win | Stride | Dim |
| CNN Archi- tecture | conv1 | 3*3 | [1,1] | 84*84 | 3*3 | [1,1] | 9*784 | 3*3 | [3,1] | 3*784 | 3*3 | [1,1] | 3*784*3 |
| | pool1 | 3*3 | [3,3] | 28*28 | 3*4 | [3,4] | 3*196 | 1*4 | [1,4] | 3*196 | 2*4 | [1,4] | 2*196*3 |
| | conv2 | 3*3 | [1,1] | 28*28 | 3*3 | [1,1] | 3*196 | 3*3 | [1,1] | 3*196 | 3*3 | [1,1] | 2*196*3 |
| | pool2 | 2*2 | [2,2] | 14*14 | 1*4 | [1,4] | 3*49 | 1*4 | [1,4] | 3*49 | 2*4 | [1,4] | 1*49*3 |
| | | Rep | Set | Set (>7reps) | Rep | Set | Set (>7reps) | Rep | Set | Set (>7reps) | Rep | Set | Set (>7reps) |
| Accuracy (%) | Train | 99.54 | 99.90 | 99.96 | 99.58 | 99.89 | 99.99 | 99.54 | 99.88 | 99.96 | 98.64 | 99.53 | 99.75 |
| | Test | 81.46 | 84.86 | 89.01 | 81.20 | 85.47 | 88.85 | 81.70 | 85.61 | 89.39 | 83.34 | 87.04 | 90.47 |

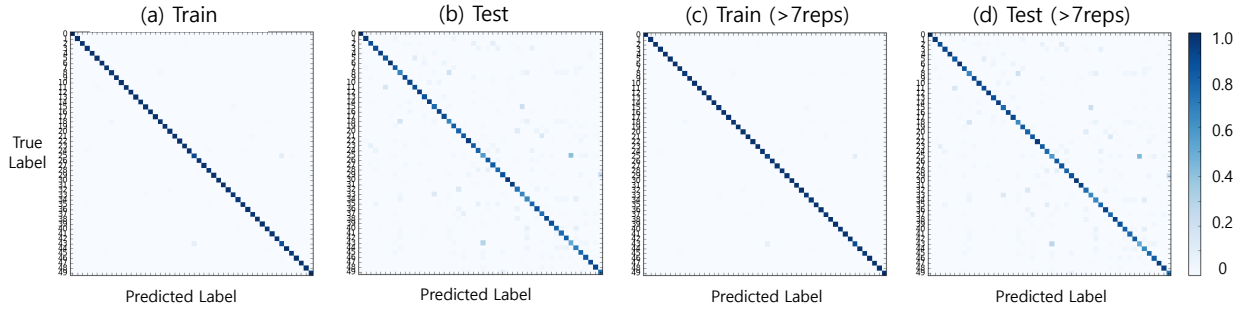


Fig. 6. Normalized confusion matrices for the $3*3*784$ experiment. High classification accuracy is achieved for the majority of exercises, as seen by the dominant diagonal. 43:Declined bench press with barbell exercises are often misclassified as 25:Alternating LEG ROMANIAN deadlift with dumbbell exercises.

standardized over the entire training dataset to have zero-mean and a variance of one in preprocessing. No additional preprocessing such as filtering or frame-aligning is applied to the raw data. In particular, we do not perform any normalization to attempt to correct for differences in sensor placement or alignment between users.

A. Effects of Image Formatting for Convolutions

As explained in III-B, there are three choices for shaping the input data: $84*84$, $9*784$ and $3*784*3$. Note that depending on the image formatting, features which are separated in the input data vector may never have a chance to be convolved together until the last convolution layer. For example, in the $9*784$ full experiment, `Acc_x_local`, located in the top row of the image, will not be convolved with `EulerAngle_z`, located in the bottom row, until the last layer. On the other hand, `Acc_z_world`, located in the 6th row in the image, will have many chances to be convolved with `EulerAngle_x`, located in the 7th row, because they are adjacent to each other.

To remove this bias posed by the location of features in the input data vector, we may separate feature groups so that no convolution happens between the groups in the first layer. This is achieved by a CNN with $3*784*3$ images. In $3*784*3$ images, convolutions will happen only within a feature group in the first layer, and their contributions will be summed up to create new feature maps for the second layer.

Another approach to avoid inter-group convolutions in the first layer is realized in the $9*789$ disjointed model. In this case, the image format is the same as $9*789$ full, however, convolution windows jump from a feature group to another feature group by taking a stride size of three (See Fig. 5). Therefore, the number of features will be reduced from 9 to 3 after the first layer and these 3 features will be convolved in the following layers.

The results after training for 40 epochs with these different image formats are presented in Table I. For the data which have more than 7 reps, the best test result is 90.47%, achieved by the $3*3*784$ model while the worst test result is 88.85%, achieved by the $9*784$ full model. The results show that convolutions on disjointed feature groups ($9*784$ disj and $3*784*3$) provide slightly better results than convolutions over full feature groups ($84*84$ and $9*784$ full). In other words, convolutions between different feature groups (e.g. `Acc_y`, `Acc_z` and `EulerAngle_x`) have little benefit for the classification task. These results are consistent with what we expected: physically meaningful convolutions can create better features than random convolutions.

Fig. 6 presents the confusion matrix for the $3*784*3$ experiments in rep-based predictions. The most easily classifiable exercises are 6:Wide-grip back lat pull-down with pulley machine (Train: 99.86%, Test: 98.08%) and 0:Standing triceps extension with dumbbell (99.90%, 96.21%) while the hard-

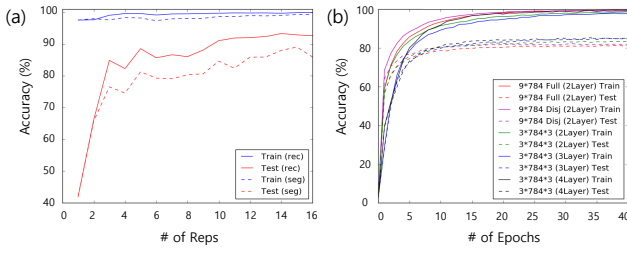


Fig. 7. (a) Performance comparison between small-rep and many-rep exercises. (b) Learning curves with different CNN architectures.

est are 43:Declined bench press with barbell (88.06%, 51.71%) and 25:Alternating Romanian deadlift with dumbbell (90.90%, 56.77%). In particular, the chance that the exercise 25 is misclassified as the exercise 43 is 41.69%. One possible reason for this is that they have relatively weak signals and similar lengths. Indeed, the exercise 43 and 25 are the exercises which have the first and third smallest signal magnitudes among the 50 exercises and similar rep lengths which are 2.00 and 2.11 seconds on average.

B. Effects of Data Reps and CNN Depth

As presented in Table I, the data which have many reps show better performance than the data which have few reps. This is to be expected in set-based predictions because if there are many reps in a set, then the final decision can be made based on the votes of many rep-based predictions. On the other hand, in extreme cases, single-rep sets cannot take the benefit of voting, thus, they obtain the same performance in rep-based and set-based predictions.

An interesting observation is that many-rep data show better performance than few-rep data even in rep-based predictions (Fig. 7(a)). The reps from 1-rep exercises have 42.32% test accuracy while the reps from 20-rep exercises have 97.23% accuracy. A possible explanation for this result is that people perform more consistent movements when they perform many-rep exercises with a light load, while they tend to demonstrate more explosive and fluctuating movements when they perform few-rep exercises with a heavier load. Indeed, the average loads for 1-rep and 20-rep exercises are 51.1kg and 20.9kg, while the variances of the all reps of 1-rep and 20-rep exercises are 5.59 and 0.94, respectively.

Fig. 7(a) shows that trained knowledge from few-rep exercises fails to be generalized to other unseen trials. This can be due to a large within-class variability in few-rep exercises, as we observed a large variance value from 1-rep exercise data. We may avoid this problem by reporting classification results only when many reps are observed. However, few-rep exercise data should not be ignored because many athletes often train with few-rep exercises for building their muscle strength. In the *PUSH* dataset, 36% of the sets have fewer than 8 reps and 9% of the sets have fewer than 4 reps. Thus, improving the classification performance for few-rep exercises will be the next challenge in our exercise recognition problem.

TABLE II
LEARNING RESULTS WITH THE $3 \times 784 \times 3$ INPUTS WITH DIFFERENT DEPTHS OF CNNs. DEEPER NETWORKS SHOW BETTER GENERALIZATION ABILITY, IF LARGE ENOUGH DATA ARE AVAILABLE.

| Layers | Feature maps | Dataset | Rep | Set | Rec (>7reps) |
|--------|----------------|---------|--------------|--------------|--------------|
| 2 | 32-64 | Train | 98.64 | 99.53 | 99.75 |
| | -(1024)-(1024) | Test | 83.34 | 87.04 | 90.47 |
| 3 | 32-64-128 | Train | 98.49 | 99.39 | 99.78 |
| | -(1024)-(1024) | Test | 85.26 | 88.55 | 92.14 |
| 4 | 32-64-128-256 | Train | 99.12 | 99.65 | 99.86 |
| | -(2048)-(1024) | Test | 84.98 | 88.41 | 92.08 |

To improve the generalization ability of the network, we attempted to increase the depth of the CNN. In this experiment, the 3-layer (92.14%) and 4-layer (92.08%) CNNs show better test results than the 2-layer (90.47%) CNN (Table II). From these results, it appears that additional layers provide some performance improvement. However, CNNs with more than 3 layers fail to outperform the 3-layer CNN. A direction for future work is to develop a customized deep CNN architecture which improves performance over the standard 3-layer CNN, particularly for small-rep exercises. The learning curves with different CNN architectures and image formats are presented in Fig. 7(b).

V. CONCLUSION

In this paper, we propose an approach for classifying large-scale wearable sensor data of exercise movements using CNN and demonstrate 92.14% classification accuracy on a 50-class exercise dataset with the 3-layer CNN. Experimental results empirically indicate that treating different feature groups - local acceleration, world acceleration, Euler angle in this case - as different channels of images ($3 \times 784 \times 3$) gives better results than providing 2D square (84×84) images or rectangle (9×783) images. Also, deeper CNNs give better results than shallow CNNs, although further research is required to fully exploit the benefit of a deeper structure. Lastly, sets with a large number of reps are easier to classify than ones with a few reps because people tend to perform movements more consistently when they perform a large number of reps of an exercise.

The current research used pre-segmented data which have relatively smaller within-class variance than nonsegmented data. The pre-segmented data also ease the variable length problem of time-series data so that CNN can treat the data as fixed-size images using simple zero-padding. In future work, we will investigate classification without segmentation, by introducing deep CNN or combining with neural networks for time-series data, e.g., LSTM [23].

APPENDIX

- A. List of Exercises for *PUSH* dataset
(Sorted by the number of reps)
0. Standing triceps extension with dumbbell
1. Alternating lunges with dumbbell
2. Hammer-curl with dumbbell

3. Underhand-grip bent-over row with barbell
4. Lying triceps extension with dumbbell
5. Rope triceps push-down with pulley machine
6. Wide-grip back lat pull-down with pulley machine
7. Alternating backward lunges with dumbbell
8. Inclined bench press with dumbbell
9. Preacher curl with EZcurl bar
10. Side-raise with dumbbell
11. Triceps push-down with pulley machine
12. Push-ups
13. Wide-grip front lat pull-down with pulley machine
14. Kettlebell swing
15. Front-raise with dumbbell
16. Right-handed triceps Kick-back with dumbbell
17. Bench fly with dumbbell
18. Alternating traveling lunges with dumbbell
19. Reverse fly with dumbbell
20. Narrow-grip lat pull-down with pulley machine
21. Bench press with dumbbell
22. Alternating lunges with barbell
23. Seated military press with dumbbell
24. Goblet squat with dumbbell
25. Alternating Romanian deadlift with dumbbell
26. Bicep curl with dumbbell
27. Bent-over row with barbell
28. Left-handed split squat with barbell
29. Right-handed one arm row with dumbbell
30. Curl with EZcurl bar
31. Bent-over row with dumbbell
32. Romanian deadlift with barbell
33. Upright row with barbell
34. Hip thrust with barbell
35. Standing military press with dumbbell
36. Inclined fly with dumbbell
37. Inclined bench press with barbell
38. Alternating grip bent-over row with barbell
39. Arnold press with dumbbell
40. Goblet squat with kettlebell
41. Overhead press with barbell
42. Barbell good-morning
43. Declined bench press with barbell
44. Alternating step-ups with dumbbell
45. Bicep curl with pulley machine
46. Narrow-grip bench press with barbell
47. Wide-grip inclined bench press with barbell
48. Right-handed split squat with barbell
49. Wide-grip bench press with barbell

B. List of Exercises Used in RecoFit Research [17]

0. Crunch
1. Row
2. Punch
3. Jumping jack
4. Kettlebell swing
5. Triceps extension
6. Push-up
7. Rowing machine
8. Russian twist
9. Back fly
10. Shoulder press
11. Squat
12. Curl

ACKNOWLEDGMENT

This research was supported by Canada's Natural Sciences and Engineering Research Council. We thank Rami Alhamad and PUSH Inc., who provided the wearable sensor data for the research.

REFERENCES

- [1] F. Ofli, G. Kurillo, . Obdrlek, R. Bajcsy, H. B. Jimison, and M. Pavel, "Design and evaluation of an interactive exercise coaching system for older adults: Lessons learned," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 1, pp. 201–212, Jan 2016.
- [2] J. Fasola and M. Mataric, "A socially assistive robot exercise coach for the elderly," *Journal of Human-Robot Interaction*, vol. 2, no. 2, pp. 3–32, 2013.
- [3] B. Görer, A. A. Salah, and H. L. Akin, "An autonomous robotic exercise tutor for elderly people," *Autonomous Robots*, vol. 41, no. 3, pp. 657–678, 2017.
- [4] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976 – 990, 2010.
- [5] J. Aggarwal and L. Xia, "Human activity recognition from 3d data: A review," *Pattern Recognition Letters*, vol. 48, pp. 70 – 80, 2014.
- [6] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 1097–1105.
- [8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.
- [9] T. N. Sainath, A. r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 8614–8618.
- [10] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, October 2014*, pp. 1746–1751.
- [11] PUSH Design Solution Inc. [Online]. Available: <http://www.trainwithpush.com/>
- [12] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, 1989.
- [13] X. Long, B. Yin, and R. M. Aarts, "Single-accelerometer-based daily physical activity classification," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Sept 2009*, pp. 6107–6110.
- [14] S. Chernbumroong, A. S. Atkins, and H. Yu, "Activity classification using a single wrist-worn accelerometer," in *Software, Knowledge Information, Industrial Management and Applications (SKIMA), 2011 5th International Conference on, Sept 2011*, pp. 1–6.
- [15] S. Chernbumroong, S. Cang, A. Atkins, and H. Yu, "Elderly activities recognition and classification for applications in assisted living," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1662 – 1674, 2013.
- [16] D. Biswas, A. Cranny, N. Gupta, K. Maharatna, J. Achner, J. Klemke, M. Jbges, and S. Ortmann, "Recognizing upper limb movements with wrist worn inertial sensors using k-means clustering classification," *Human Movement Science*, vol. 40, pp. 59 – 76, 2015.
- [17] D. Morris, T. S. Saponas, A. Guillory, and I. Kelner, "Recofit: Using a wearable sensor to find, recognize, and count repetitive exercises," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, NY, USA, 2014*, pp. 3225–3234.
- [18] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.
- [19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning, 2010*, pp. 807–814.
- [20] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *ArXiv e-prints*, Dec. 2014.
- [21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [22] M. A. et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, vol. abs/1603.04467, 2016.
- [23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov 1997.