Design of Novel Deep Learning Models for Real-time Human Activity Recognition with Mobile Phones

Mark Nutter
ARM Research
Austin, TX
Email: mark.nutter@arm.com

Catherine H. Crawford
IBM Research
Cambridge, MA
Email: catcraw@us.ibm.com

Jorge Ortiz
IBM Research
Yorktown Heights, NY
Email: jjortiz@us.ibm.com

Abstract*—In this paper we present deep learning based techniques for human activity classification that are designed to run in real time on mobile devices. Our methods minimize the size of the model and computational overhead in order to run on the embedded processor and preserve battery life. Prior work shows that the inertial measurement unit (IMU) data from waist-mounted mobile phones can be used to develop accurate classification models for various human activities such as walking, running, stair-climbing, etc. However, these models have largely been based on hand crafted features derived from temporal and spectral statistics. More recently, deep learning has been applied to IMU sensor data, but have not been optimized for resource-constrained devices. We present a detailed study of the traditional hand-crafted features used for shallow/statistical models that consist of a over 561 manually chosen set of dimensions. We show, through principal component analysis (PCA) and application of a published support vector machine (SVM) pipeline, that the number of features can be significantly reduced – less than 100 features that give the same performance. In addition, we show that features derived from frequency-domain transformations do not contribute to the accuracy of these models. Finally, we provide details of our learning technique which creates 2D signal images from windowed samples of IMU data. Our pipeline includes a convolutional neural network (CNN) with several layers (1 convolutional layer and 1 averaging layer and a fully connected layer). We show that by removing the steps in the pipeline and layers in the CNN, we can still achieve 0.98 F1 score but with a much smaller memory footprint and corresponding computational cost. To increase the classification accuracy of our pipeline we added a hybrid bi-class support vector machine (SVM) that was trained using the labeled and flattened convolutional layer after each training image was processed. The learned feature set is almost half the size of the original hand crafted feature set and combining the CNN with the SVM results in 0.99 F1 score. We also investigate a novel application of transfer learning by using the time series 2D signal images to re-train two different publicly available networks, Inception/ImageNet and MobileNet. We find that re-trained ImageNet networks could be created $<$ 5.5MB (suitable for mobile phones) and classification accuracy ranging from 0.83 to 0.93 (F1 score), thus indicating that retraining can be a useful future direction to build new classifiers for continuously evolving activities quickly while also being applicable to mobile device classification. Finally, we show that these deep learning models may be generalizable enough such that classifiers built from a given set of users for a specified set of activities can be used for a new user/subject as well.

Keywords*—machine learning, mobile computing, sensors.

I. Introduction

The ability to monitor and classify human activity via tri-axial inertial measurement units has been of interest to the healthcare community (eldercare), the entertainment and fitness community, and for the purposes of our studies, in urban safety. In the urban safety case, we are specifically interested in understanding normal versus abnormal movement, including both significant changes in signals that may come from falling or being pushed as well as subtle changes in signal like gait change that may come from impairment or involuntary/forced movement. The pervasiveness of mobile devices equipped with such sensing capabilities provides researchers with a cost effective manner in which to study a plethora of accurately labeled human activities with a diverse cohort sample – a critical need for the urban safety case. Furthermore, open data sets of mobile phone activity data are readily available on which to develop and test classification algorithms as well (see [1] [2] for the examples used in our study).

Until recently, the vast majority of Human Activity Recognition (HAR) classifiers have been based on time series windowing and hand crafted feature sets for statistical machine learning models using both time and frequency based data [3]. A good review of the time series processing pipeline for smartphone IMU data in HAR can be found in [4] and [1]. Multiple shallow statistical machine learning techniques (Decision Tree, Naive Bayesian, K Nearest Neighbor, Support Vector Machine) have shown varying classification accuracies with multiple open data sets (for example see [5]). Furthermore, whereas deep learning has been leveraged extensively in computer vision [6] and fused multi-modal, multi-mount appendage sensors IMUs [7], it is only recently that deep learning has been applied to HAR using IMU data (we do a more extensive review of this recent work in section II).

The goal of our research is to extend the deep learning approaches currently being developed, specifically targeting classification and potentially training/retraining on constrained computing environments. In this context we not only mean constrained in the sense of limited compute and memory capacity, but also with restrictions on power consumption and potentially limited data available for training. For instance, we consider cases where classifiers should be based on personalized/individual movement to provide the most accurate prediction of subtle changes in behavior for our urban safety

978-1-5090-6014-6/18/$31.00 ©2018 IEEE

case. This would mean less data to train with because it is only one person. In addition, privacy concerns of some individuals may predispose them not to share data. Therefore, in our exploration of deep learning approaches and pipelines for HAR we focus on dimensionality reduction of learned feature sets and reducing the size of classifiers with as little computational cost as possible as this would inherently optimize for storage, compute and therefore battery life on a mobile device.

The main contributions of this paper are as follows:

- We demonstrate a CNN based architecture based on signal images in which the number of features is drastically reduced compared to previous work (via PCA analysis as well as using features that were a result of a flattened convolutional layer) while still delivering accuracy greater than 90% ;
- We apply computer vision based transfer learning techniques to the signal images using known CNN models developed for mobile devices and achieve accuracy greater than 90%
- We demonstrate that the deep learning models developed here may be generalizable across subjects so that models used with other subjects data may be used to classify new users with unlabled similar activities.

The rest of our paper is outlined as follows. In section II we review prior work applying deep learning for human activity recognition. Our own methods and processing pipeline are described in section III. Section IV is a summary of our deep learning and transfer learning pipeline results in terms of both accuracy as well as model computational and memory efficiency. Finally, section V reviews some of the open questions and future work to further improve our methodologies as well as possible further applications and use cases.

II. Related Deep Learning Work

Our objective in this review is to summarize a core set of related work in deep learning for HAR which has influenced our methodology as presented in section III. Therefore, we forego a more detailed summary of the more well established statistical shallow learning techniques which have previously been reviewed in many of the projects we cite here.

The authors in [8] developed a technique to turn the IMU signal data into two dimensional image in which each tri-axial signal was positioned adjacent to every other signal (e.g. "stacked") to capture the influence or relationship across signals. The images were created using non-overlapping windows of the IMU signal samples and the final labeled images used in CNN training were also processed with a two dimensional discrete Fourier transform. Final classification of the activity was done using a SoftMax layer and if the probability of classification for one activity was not above a specified threshhold, the classification went back to a bi-class SVM between the most likely activities. This hybrid approach resulted in greater than 90% accuracy on multiple open HAR data sets with reasonable computational cost.

A deep belief network (DBN) trained on accelerometer spectrograms is presented in [9]. Their pipeline includes both unsupervised and supervised learning, thus giving the advantage of avoiding hand crafted features as well as providing a methodology to take advantage of a potentially substantial amount of unlabeled accelerometer data from mobile phones. With sufficient optimization of pre-training stages in their network, greater than 95% accuracy was achieved in HAR classification.

A comparison of the use of both CNNs and RNNs as well as a study on the effects of hyperparameters for wearables used to classify complex human activity is presented in [3]. These researchers find a large spread in classifier performance (via F1 scores) however, their methodology to train recurrent networks demonstrate new approaches in the deep learning for the HAR space.

The authors in [10] created images from inertial sensor data using recurrence plots (RPs) and visual descriptors. In this work, various shallow and deep learning techniques were used to extract learned visual features and then a linear SVM was used for classification. This differs in our work in that we are actually using the deep learning to classify the activity from an image based approach.

For our work, we borrow the image approach used in [8] and combine it with ideas from [9] to avoid hand crafted features. However, we extend the work cited to include more optimizations that consider computational cost and memory requirements as we are targeting classification in real time on mobile devices. Given the continuous learning required for personalized classifiers, we also investigate new techniques from current computer-vision deep learning to continually learn new images and investigate those approaches as they apply to signal images.

III. Methodology

A. Data Sets, Pre-processing and Signal Images

For our experiments we focused on the publicly available mobile phone IMU datasets in [1] from the University of California Irvine Machine Learning Repository and [2] from the University of Central Florida Center for Research in Computer Vision, hereafter referred to as the UCI and UCF data respectively. The UCI data set is from a waist mounted Android device and the UCF data set is from a waist mounted iPhone. The UCI dataset is composed of data from 30 subjects each capturing 6 different activities multiple times. The UCF dataset is composed of six to nine subjects for nine activities, each activity being repeated multiple times as well.

Before doing any type of classification model building, raw IMU signals must be pre-processed to remove noise as well as to build feature set representation applicable to the deep learning models we are investigating. Following the procedures in [1] we preprocessed the data with with following steps:

1) a median filter of length 3
2) a butterworth filter at 20 Hz to exclude signals outside human activity frequencies
3) calculate the gravity component with a butterworth low pass filter at 0.3Hz
4) subtract the gravity component from the output of the first butterworth signal to get the final set of pre-processed accelerometer and gyroscope data.

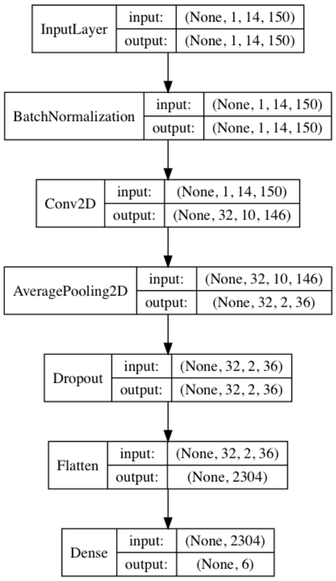
InputLayer
input: (None, 1, 14, 150)
output: (None, 1, 14, 150)
BatchNormalization
input: (None, 1, 14, 150)
output: (None, 1, 14, 150)
Conv2D
input: (None, 1, 14, 150)
output: (None, 32, 10, 146)
AveragePooling2D
input: (None, 32, 10, 146)
output: (None, 32, 2, 36)
Dropout
input: (None, 32, 2, 36)
output: (None, 32, 2, 36)
Flatten
input: (None, 32, 2, 36)
output: (None, 2304)
Dense
input: (None, 2304)
output: (None, 6)

Fig. 1. Deep Neural Network pipeline for classifying 2D gray scale signal images which are created from windowed samples of IMU data as described in III-A. The input shape for the network is a four-tuple of values indicating (batch size, channels, rows, columns). In this figure the value None indicates the batch size is not fixed by the network architecture; during training a batch size of 8 was selected. The number of channels is 1 as the signal images are gray scale. The rows component represents the number permutations for stacked IMU signals, as created in [8]. The columns component represents the window length used in creating the 2D signal images; in this case 150 columns indicates three seconds of non overlapping IMU samples from UCI data set. The output layer for UCI and UCF differ as these data sets have six and nine activities, respectively.

All of our preprocessing steps are coded in Python using the SciPy mathematics, science, and engineering library. Once these filtered signals were calculated we then followed the work in [8] to create 2D signal images with a set permutation algorithm described in that paper, stacking signals on top of each other such that every signal has the chance to be adjacent to every other signal. However, in our work, we note that for many mobile devices linear acceleration is computed simply by subtracting gravity from total acceleration [11]. Linear acceleration is in this sense redundant to the feature set so we instead choose to create signal images using just six core IMU signals (xyz for both gyroscope and acceleration). Further, we did not do any subsequent 2D Discrete Fourier Transform on this signal image based on our observation in feature analysis that the frequency domain did not contribute significantly to the overall accuracy of our results (see section IV-A and Figure 2). Data augmentation is one technique that may be used to overcome the problem of data scarcity [12], such as the case with UCI and UCF data sets. However for this work we do not augment the data by applying simple transformations such as flip, rotate or scale as we considered these would change the HAR signal images in such a way as to be unrecognizable. Recent work on processing environmental audio signals using CNNs [13] however suggests that data augmentation is worth investigating in future work on HAR classifiers.

B. Deep Neural Network (DNN) Pipeline

All of our neural network pipelines are coded in Python using the Keras [14] deep learning library. For our first pipeline, we train a deep neural network from scratch, using the 2D signal images from III-A as training data, and withholding 30% as test data. Our network architecture takes inspiration from [15] but we eliminate the large fully connected layer, as this adds significantly to the network size (Tab. I). For this work we did not use network quantization or other compression techniques [16]; however it is expected these would further reduce the network size which is especially important for mobile-class device.

Network	# Parameters	Size (MB)
DNN (no fc)	12,998	0.2
DNN+fc128	101,758	1.1
DNN+fc256	205,414	2.3
DNN+fc512	545,746	5.0

TABLE I. NUMBER OF PARAMETERS AND MODEL SIZES FOR OUR DEEP NEURAL NETWORK, WITH AND WITHOUT FULLY CONNECTED LAYERS.

To train the deep neural network we used batch normalization [17] to help regularize the input as well as dropout to help reduce the effects of over-fitting to the scarce data [18]. The final layers for UCI and UCF differ slightly, to account for the number of activities in each data set (Fig. 1). As in [8] we also train a bi-class SVM to further improve our classification result. Rather than using the hand crafted feature vector, we instead train the bi-class SVM using flattened output from our network's pooling layer. This saves computational cost of calculating the 561 feature components, as the output of the pooling layer is a natural byproduct of the forward pass through our network.

C. Transfer Learning

Transfer learning is not new either in the general case of image classification or even more novel techniques developed specifically for HAR [19]. However, our goal was to determine if computer vision models could be used with retraining and transfer learning for the signal images generated from IMUs to focus on smaller classifiers which could be continuously trained/updated as well as eventually leverage computer vision deep learning tools for interpretability.

In this work we apply transfer learning by fine tuning two well known ImageNet models, Inception V3 [20] and MobileNet [21] to the HAR task. To the best of our knowledge our work is the first demonstration of CNN trained on natural images being re-trained to classify images comprised of time-domain signals [15], [22]. We take inspiration from previous work demonstrating that CNNs trained on large-scale datasets can be transferred to other visual recognition tasks with limited amounts of training data [23], and that transferring features from distant tasks can be better than using random features [24].

For each model we take the base layers but remove the top fully-connected layers. The models are initialized with their pre-trained ImageNet weights. We then add a small fully-connected top layer sized appropriate to number of HAR activities in the data set (six in the case of UCI, nine for UCF) and train on top of the stored features for ten steps. To further improve our previous result, we ”fine-tune” the last convolutional blocks for a further ten steps. Despite the relative scarcity of training data, we find the retrained networks

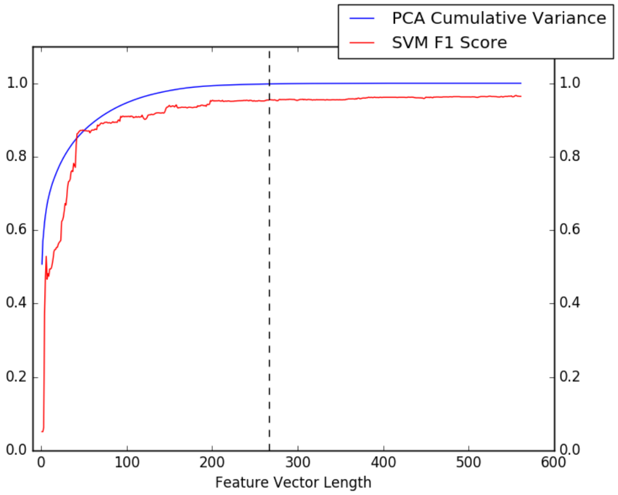

Fig. 2. PCA cumulative variance (blue) and multi-class SVM accuracy (red), by feature vector length. Time domain features are to the left of the line x=266, while frequency domain features are to the right.

accurately classify HAR activities after just twenty epochs. Further improvement may be possible as these networks appear to be under-fitting, as illustrated in Figures 4 and 3.

IV. Results

To test the capabilities of our training and classification pipelines, we designed three different experiments. In the first experiment, for each dataset, we train using the entirety of data collected for all subjects and then randomly select test subjects and activities to determine accuracy and F1 scoring (where accuracy is just the ability to identify an activity correctly). The next experiments we run are to understand the following:

- Is there a general classifier for all subjects for a given activity? – e.g. train on all activities for all subjects except subject N and then use only subject N's data as the test data.
- Can a classifier be used to determine when a new activity or a potential anomaly is detected for a specific subject? – e.g. for a single subject train on all activities except for activity X and determine if the classification step consistently does not recognize (classify) this activity.

A. *Feature Selection*

As we are designing a classification system that can run in real time on mobile devices, we were careful to eliminate as much memory and compute requirements in our model. Therefore, we first reviewed the features which were contributing to the classification in previously developed models. Figure 2 shows the results of running an SVM bi-class classifier based on the total set of data as training data and random sampled test data with different numbers of features. It is evident from this data that O(100) features are required, rather than the original 561, for F1 scores greater than 0.9. More detailed investigation into which features were contributing to the F1 scores showed that the frequency domain did not make significant contributions. Specifically note that the cumulative variance and F1 score do not improve significantly with the addition of frequency domain signals. We used this insight to inform the design of our CNN and CNN+SVM classifiers.

B. *CNN and CNN+SVM Classifiers*

The results for our deep learning pipeline are listed in Table II. We find that by creating 2D signal images comprised of six IMU time series signals and by removing the large fully connected layers, our network can achieve over 0.98 F1 score on UCI and 0.92 on UCF, when training with all data and randomly sampling within the entire data set for subjects and activities. These results can be further improved by training a bi-class SVM as described in section III-B.

Network	UCI	UCF
DNN+fc128	0.98	0.87
DNN+fc256	0.97	0.89
DNN+fc512	0.96	0.88
DNN (no fc)	0.98	0.92
DNN+SVM	0.99	0.97

TABLE II. F1 SCORES FOR VARIATIONS OF OUR CNN ON THE UCI AND UCF DATA SETS.

C. *Transfer Learning*

From our transfer learning experiments we broadly find that:

1) a re-trained ImageNet can achieve near state-of-the art accuracy on HAR (Fig. 5).
2) MobileNet out performs Inception V3 for HAR.
3) accuracy for both models improves with sample window length.
4) accuracy for both models is better with UCI data set, which has only six activities, than with UCF which has nine.

For the UCI data set, Inception V3 achieved an F1 score of 0.92 while MobileNet achieved an F1 score of 0.97. For the UCF data set, Inception V3 achieved an F1 score of 0.83 while MobileNet achieved an F1 score of 0.93 (figure omitted for space).

D. *Experiments for Generalized Classifiers and New Activity Detection*

We also wanted to understand if the deep learning classification methods we employed were generalizable. That is to say, can classifiers built from a given set of users for a specified set of activities be used for a new user/subject as well. Figure 6 shows the distribution of F1 scores for the DNN pipeline described in this paper. From these experiments, we note that the minimum F1 score was 0.66, and 20 out of 25 subject test achieving F1 scores of 0.80 or better. Similar results were found with the MobileNet generated model for these experiments. While this is promising that a generalized model could be generated with enough subjects of different attributes (age, weight, height), there is still work to be done to understand what the limits are of these models and when retraining will be required *a priori*.

The last set of experiments we ran focused on determining if deep learning classifiers could be used to determine for a specific subject if a new activity could be detected. The

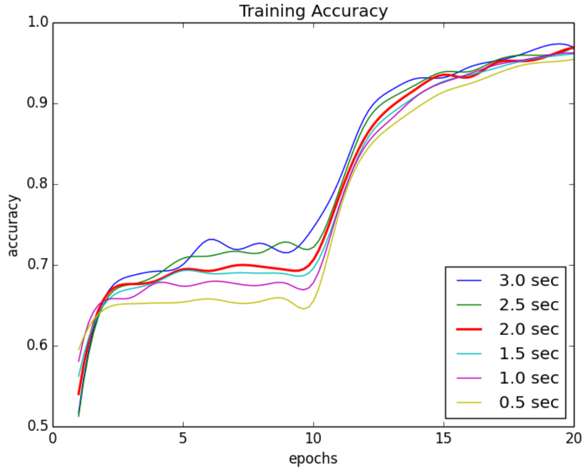

Fig. 3. Training accuracy for the Inception V3 model UCI data set shown with varying training epochs as well as window sizes (seconds, shown in different colors).

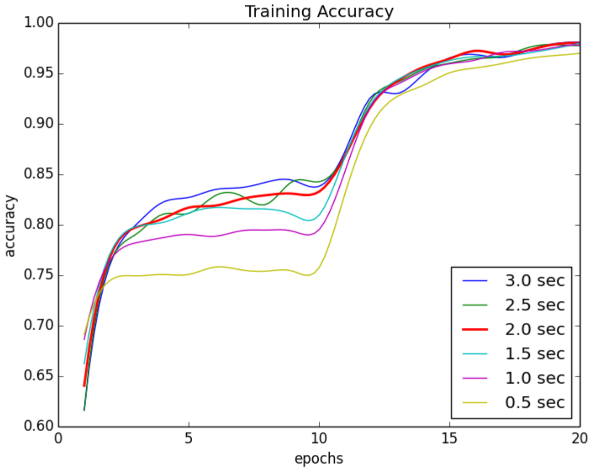

Fig. 4. Training accuracy for the MobileNet model on the UCI data set shown with varying training epochs as well as window sizes (seconds, shown in different colors).

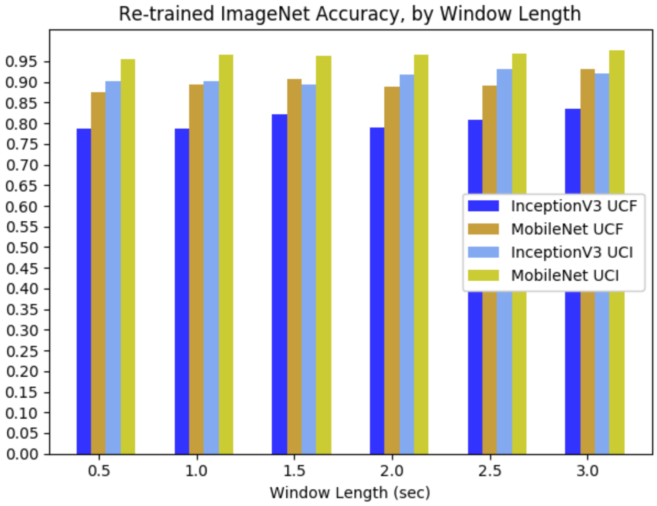

Fig. 5. F1 score for retrained InceptionV3 and MobileNet models, by window length.

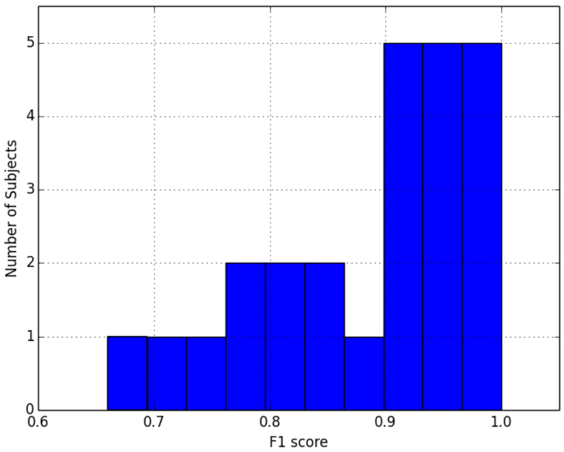

Fig. 6. Histogram of F1 scores for the 25 subjects in the UCI HAR dataset using the DNN pipeline based classifer trained on all other subject data except for the subject in the test set.

results of those experiments, again using the DNN pipeline and MobileNet, showed that for classifying on data that was not part of the subjects training set, the classification was inconclusive with F1 scores generally between 0.4 and 0.5. At this point, without further types of probalistic reasoning in the pipeline, it is difficult to imagine that this alone could be used to determine a new activity had been found, versus, for instance a slight anomaly in an existing activity. More investigation is required.

V. Discussion

The goal of our work was to develop a novel deep learning based classifier that was suitable to deploy, both in terms of size and computational cost on mobile devices for real time *in situ* classification. To that end, we investigated feature set reduction, optimized CNN architectures and even applying transfer learning techniques from computer vision to 2D images constructed from IMU signals. Our results demonstrated that we can get greater than 0.9 (F1) accuracy using both our CNN pipeline as well the MobileNet model with transfer learning on over 9 activities. MobileNet proved to provide higher accuracy than the much larger and more complex Inception V3 model (see figures 7 and 8). From our review of the papers that used these datasets as well as the review of state of the art HAR work presented here (see for instance [5] for a summary of methods showing F1 scores between 0.6 and 1.0 for various activities and various datasets), our methods are consistent, if not better than those results. This is promising as MobileNet transfer learning models have already been deployed on Android devices for object detection in real time [25]. We also showed that for both our DNN pipeline and the MobileNet based model has the potential for generalization across multiple users with a sufficient sample size across subjects with multiple attributes in the training set. However, much more work is needed to determine how these models can be used for accurate detection of new activities versus anomalies in existing activities as well.

Whilst the MobileNet based classifier has shown greater than 90% accuracy for the UCI and UCF datasets we need a greater understanding of both *why* this method works so well, perhaps by using the interpretability methods and tools from computer vision [26]. Our hope is that these tools can relate sections of the image responsible for classification to the actual components of the physical activity via IMU signals that provide for distinctive classification. Furthermore, given that both Inception and MobileNet accuracy decreased going from 6 labeled activities to 9 labeled activities, we also need to investigate if this trend continues experimenting with datasets that include even more labeled activities or more complex

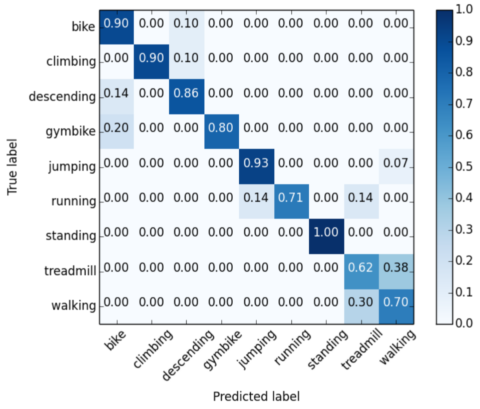

Fig. 7. Confusion matrix for retrained Inception V3 with the UCF data set and window length of 3 seconds.

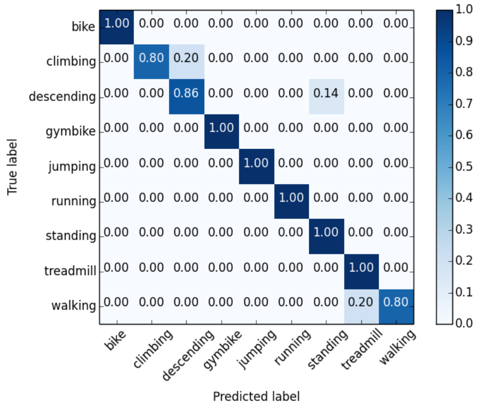

Fig. 8. Confusion matrix for retrained MobileNet with the UCF data set and window length of 3 seconds.

activities.

In the future, we will implement our own Android and wearable application to easily collect and label data for an individual as well as a prototype system to upload data into our training pipeline as well as download classifiers and then demonstrate the classifiers on Android devices in real time. Finally, our current set of experiments and results have focused on stationary behavior not postural transition behavior, e.g. walking to running, sitting to standing. In order to have a fully accurate activity recognition pipeline, we need to address this by applying time series and temporal/state dependence deep learning techniques such as LSTM or kShape, or even state machines or Hidden Markov Models (see for instance [9]), optimizing those for mobile devices and integrating with our current CNN or transfer learning based pipeline.

REFERENCES

[1] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones." in *ESANN*, 2013.

[2] C. McCall, K. K. Reddy, and M. Shah, "Macro-class selection for hierarchical k-nn classification of inertial sensor data." in *PECCS*, 2012, pp. 106–114.

[3] N. Y. Hammerla, S. Halloran, and T. Ploetz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *arXiv preprint arXiv:1604.08880*, 2016.

[4] O. S. Schneider, K. E. MacLean, K. Altun, I. Karuei, and M. M. Wu, "Real-time gait classification for persuasive smartphone apps: Structuring the literature and pushing the limits," in *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, ser. IUI '13. New York, NY, USA: ACM, 2013, pp. 161–172. [Online]. Available: http://doi.acm.org/10.1145/2449396.2449418

[5] U. Fareed, "Smartphone sensor fusion based activity recognition system for elderly healthcare," in *Proceedings of the 2015 Workshop on Pervasive Wireless Healthcare*, ser. MobileHealth '15. New York, NY, USA: ACM, 2015, pp. 29–34. [Online]. Available: http://doi.acm.org/10.1145/2757290.2757297

[6] M. Hasan and A. K. Roy-Chowdhury, "Continuous learning of human activity models using deep nets," in *European Conference on Computer Vision*. Springer, 2014, pp. 705–720.

[7] S. Münzner, P. Schmidt, A. Reiss, M. Hanselmann, R. Stiefelhagen, and R. Dürichen, "Cnn-based sensor fusion techniques for multimodal human activity recognition," in *Proceedings of the 2017 ACM International Symposium on Wearable Computers*, ser. ISWC '17. New York, NY, USA: ACM, 2017, pp. 158–165. [Online]. Available: http://doi.acm.org/10.1145/3123021.3123046

[8] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *Proceedings of the 23rd ACM International Conference on Multimedia*, ser. MM '15. New York, NY, USA: ACM, 2015, pp. 1307–1310. [Online]. Available: http://doi.acm.org/10.1145/2733373.2806333

[9] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep activity recognition models with triaxial accelerometers." in *AAAI Workshop: Artificial Intelligence Applied to Assistive Technologies and Smart Environments*, 2016.

[10] O. A. Penatti and M. F. Santos, "Human activity recognition

from mobile inertial sensors using recurrence plots,” *arXiv preprint arXiv:1712.01429*, 2017.

[11] developer.android.com, “Motion Sensors,” https://developer.android.com/guide/topics/sensors/sensors$_{m}otion.html$, 2017.

[12] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” *CoRR*, vol. abs/1405.3531, 2014. [Online]. Available: http://arxiv.org/abs/1405.3531

[13] J. Salamon and J. P. Bello, “Deep convolutional neural networks and data augmentation for environmental sound classification,” *CoRR*, vol. abs/1608.04363, 2016. [Online]. Available: http://arxiv.org/abs/1608.04363

[14] F. Chollet *et al.*, “Keras,” https://github.com/fchollet/keras, 2015.

[15] H. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. J. Mollura, and R. M. Summers, “Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning,” *CoRR*, vol. abs/1602.03409, 2016. [Online]. Available: http://arxiv.org/abs/1602.03409

[16] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding,” *CoRR*, vol. abs/1510.00149, 2015. [Online]. Available: http://arxiv.org/abs/1510.00149

[17] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627435.2670313

[19] J. Wang, Y. Chen, L. Hu, X. Peng, and P. S. Yu, “Stratified transfer learning for cross-domain activity recognition,” *arXiv preprint arXiv:1801.00820*, 2017.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[22] A. Kumar and B. Raj, “Deep CNN framework for audio event recognition using weakly labeled web data,” *CoRR*, vol. abs/1707.02530, 2017. [Online]. Available: http://arxiv.org/abs/1707.02530

[23] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and transferring mid-level image representations using convolutional neural networks,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1717–1724.

[24] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS’14. Cambridge, MA, USA: MIT Press, 2014, pp. 3320–3328. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969197

[25] M. Harvey, “Creating insanely fast image classifiers with MobileNet in TensorFlow,” https://hackernoon.com/creating-insanely-fast-image-classifiers-with-mobilenet-in-tensorflow-f030ce0a2991, 2017.

[26] M. T. Ribeiro, S. Singh, and C. Guestrin, “”why should i trust you?”: Explaining the predictions of any classifier,” in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. New York, NY, USA: ACM, 2016, pp. 1135–1144. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939778